SERIE B — INFORMATIK

A Classification Scheme of 3D Interaction Techniques

Lutz Kettner*

B 95–05
April 1995

Abstract

An enhanced classification scheme is described, capable to provide accurate analysis of 3D interaction techniques. It is based on examples found in the literature and summarized here in a review. The feature space of the scheme has proven to be useful in analysing particular subjects of their behavior and in deriving new interaction techniques. Several interaction techniques for the rotation task are analyzed using Fitts' law to predict their performance. The treatment specializes on 2D input devices, but several results are easily generalizable to higher dimensional input devices.

*Freie Universität Berlin, Graduiertenkolleg "Algorithmische Diskrete Mathematik", Takustr. 9, 14195 Berlin, E-mail: kettner@inf.fu-berlin.de. Supported by the Deutsche Forschungsgemeinschaft under grant We 1265/2-1.

1 Introduction

We need two definitions to clarify terminology for the rest of the text:

Definition 1 (interaction task): *An* interaction task *is what a user wants to do (e.g. translating, rotating or scaling an object, moving the viewpoint, inserting, deleting, modifying, etc.).*

Definition 2 (interaction technique): *An* interaction technique *is a means for achieving a certain desired interaction task (e.g. virtual sphere rotation, object handles for constrained translation, etc.).*

Here, we are interested in 3D interaction techniques that are performed with 2D input devices such as mice or graphic tablets. The key problem is the information mapping from the lower degree of freedom of the input device to the higher degree of freedom of many 3D interaction tasks (*e.g.* rotation with three degrees of freedom). To support the user in learning and remembering a 3D interaction technique, *metaphors* are used, like in other user interfaces (*e.g.* a typewriter as a metaphor for word processing on a computer). Thus, we are talking about *3D metaphors*, defined as 3D interaction techniques associated with metaphorical concepts.

This will be explained in more detail soon: including an example, mentioning the relevant context, and stating a definition for 3D metaphors. Afterwards, we give a short review of existing descriptions of 3D metaphors in literature and approaches for systematization. Then, we present our proposition for a classification scheme and apply it to selected examples. The discussion focusses on what we can deduce from features in the classification scheme, how we can predict performance, and what type of new 3D metaphors we can expect.

The goal is to systematize existing research results in a unified framework. Developments of 3D metaphors are often based on ad hoc ideas and decisions. There is a gap between theory and practical implementations in many applications. Collecting them together in a taxonomy will focus attention on unsolved problems, and we can investigate gaps in the feature space which promise new and interesting 3D metaphors.

Altogether, insight in these problems can help us in developing good solutions bridging the gap between two-dimensional input devices and 3D interaction tasks. This will enhance the usability and the propagation of 3D applications.

1.1 2D input devices

Current developments emphasize input devices with more than two degrees of freedom, ranging from space mice to data suits. Nonetheless, two-dimensional devices, like mice or digitizers in pen computers, still possess several advantages:

- Two-dimensional input devices are widely spread and accepted with the use of traditional graphical user interfaces. Thus, together with their need for less sensors than 3D input devices, they are more economical.

- The forearm can usually rest on the table, while three-dimensional devices fatigue the operator[1]. This counts in ergonomics.
- Normally, three-dimensional interaction is embedded in a dialog context of two-dimensional window operations, menu selections and keyboard typing. The disadvantage of frequent changes between different devices might outweigh advantages of specialized devices. Thus, a data-glove is not appropriate for large amounts of typing.

  Current developments like 3D widgets [Conner et al., 1992] do all tasks with 3D interaction techniques. This seems reasonable for specialized applications, but applications with other interaction task demands or even mixed application use will also prefer other input devices.
- Depending on the application, 3D interaction will gain advantages from restrictions to one or two degrees of freedom. Difficulties with handling a great number of degrees of freedom can introduce errors. Thus, interaction time is limited due to correcting these errors when preciseness is demanded. Even in two dimensions, ruler and compass provide the reduction to one degree of freedom.
- Also physical size is a limiting factor for the application of a device, see trackballs in notebooks or digitizers in pen-tops.

Another reason for our restriction to two dimensional devices is that the device-to-operation mappings are not as interesting for 6D devices as they are for 2D devices. In [Ware and Osborne, 1990] the main principles for 6D devices are presented: one can be called the WORLD-IN-HAND-metaphor, the other one can be called the EYEBALL- or CAMERA-IN-HAND-metaphor. The associated mappings are inverses to each other. The FLY-THROUGH-metaphor can be applied independently. Both can also be found for 3D metaphors with 2D devices. A couple but not all of the phenomena around these 3D metaphors can also be found with the natural mappings of 6D devices.

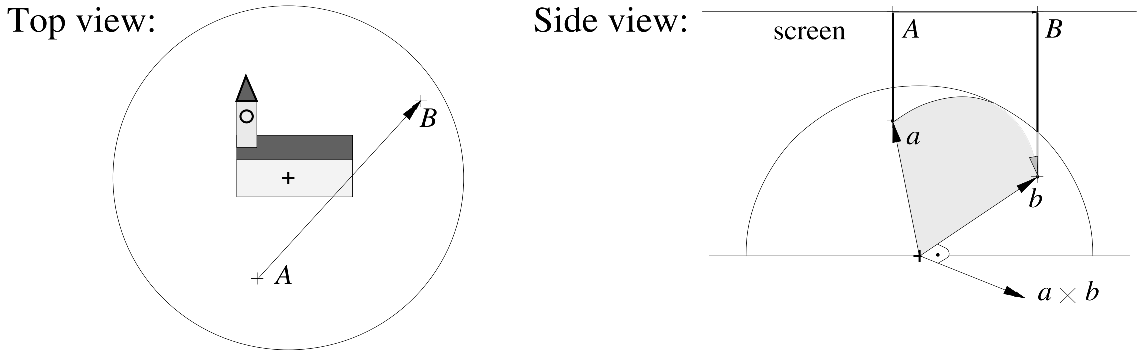
1.2 An example: the VIRTUAL-SPHERE-metaphor

The VIRTUAL-SPHERE-metaphor was first presented in [Chen et al., 1988] and briefly summarized in [Hultquist, 1990]. The metaphorical concept is as follows: Imagine the manipulated object embedded in a glass sphere. At the user interface, the circumference of the sphere is only drawn as a thin circle around the object, see figure 1 on the left side[2]. Now, the sphere can be rotated around its center by simply touching it at its imaginary surface and moving it around with the mouse pointer.

[1] A notable exception is the spaceball.

[2] This is done in the original implementation, but in fact most other implementations omit this drawn circle as a prompt.

Figure 1: The VIRTUAL-SPHERE-metaphor.



We can see a side view of this imaginary configuration on the right side of figure 1. The screen is on the top, with the mouse pointer moving from A to B. We look at a single small move event, although we have enlarged it in the figure for a clearer presentation. Both positions are projected on the virtual sphere to the endpoints of the vectors a and b. Assume that the virtual sphere is a unit sphere with its center in the origin. Then the rotation goes around the axis $a \times b$ with the angle $\arccos(a \cdot b) \in [0, \pi)$, all orientations with the right-hand rule. The mouse movement describes a geodesic on the sphere.

Bigger rotations are accumulated from many small mouse move events. Note that the metaphor is not transitive. A rotation from a movement from A to B to a third point C is in general not equal to a rotation from a single mouse move event from A to C.

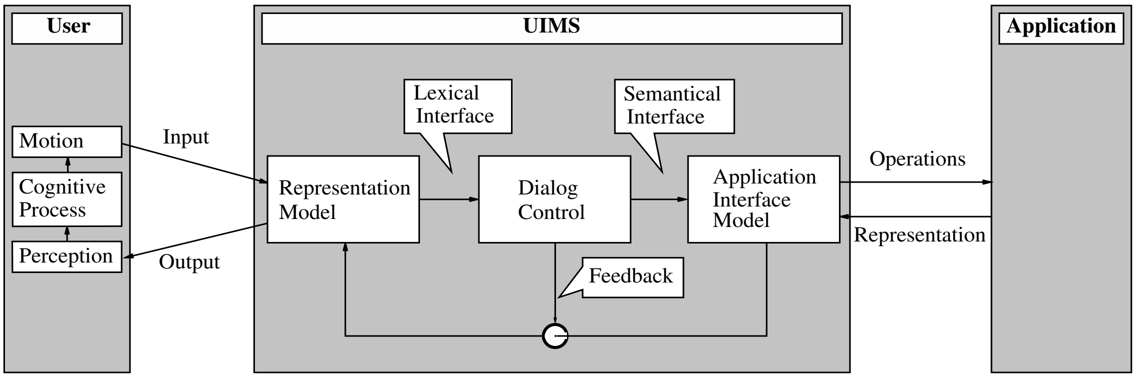
1.3 Seeheim-model for UIMS

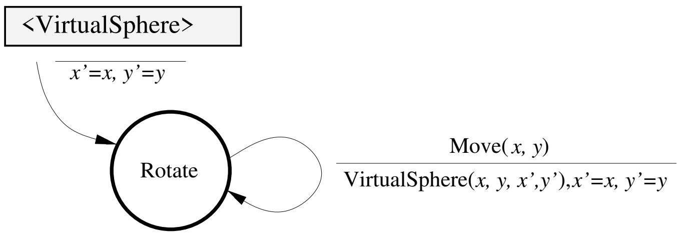
To formalize descriptions, the implementations of 3D metaphors are located in the Seeheim-model for user interface management systems (UIMS) [Olsen, 1992]. It provides us with clear interface definitions as depicted in figure 2. From the left side, 2D user interface events, like mouse moves or button presses, are received as *lexical tokens*. On the right side, *semantic operations* of a 3D graphics application are executed. 3D metaphors belong to the *syntactical analysis* in the dialog control of the UIMS. So, two different characteristics describe the behavior of a 3D metaphor: first, the dialog control which can easily be modeled with state transition diagrams, and second, the mapping from 2D parameters of the user interface events to the 3D parameters of the semantic operations.

The information flow from the application to the user represents the visualisation part. The feedback from the dialog control can influence the representation and adds its own information to this channel. The representation and the application interface model serve as interfaces to abstract from concrete devices and applications.

For our example above, the behavior can be described as follows: The points A and B are represented in x, y-screen coordinates as (x', y') and (x, y) respectively.

Figure 2: Seeheim-model for user interface management systems (UIMS)





$$\begin{aligned}&\mathrm{VirtualSphere}(x, y, x', y') := \\ &\qquad \boldsymbol{M} = \boldsymbol{M}'\,\boldsymbol{R}_{axis}(a \times b, \arccos(a \cdot b)), \\ &\qquad a = \Pi(x', y'),\ \ b = \Pi(x, y).\end{aligned}$$

The state diagram handles the mouse movements and remembers one previous mouse position. Therefore, the primed variables are initialized at the transition entering the **Rotate**-state and are updated on every mouse movement at the right transition. The events which enter or leave the **Rotate**-state are not specified. They are usually button-press events or other imperative events as defined in general for input devices in [Sherr, 1988].

The parameter mapping is expressed with the function **VirtualSphere**. $\boldsymbol{M}'$ denotes the old, $\boldsymbol{M}$ the new transformation matrix that is manipulated with the metaphor. $\boldsymbol{R}_{axis}(n, \theta)$ denotes the rotation matrix for the angle θ around the axis n. Π denotes the projection of a point (x, y) in screen coordinates onto the imaginary unit sphere, resulting in a unit vector. These $\boldsymbol{M}$ and $\boldsymbol{R}_{axis}$ are elements of the application interface model.

1.4 Definition of a 3D metaphor

We have seen that a 3D metaphor can be described in an informal fashion to stress the intention behind the metaphor. This is what someone should explain to a novice to establish relations to his real world experience. Moreover, we have seen that a 3D metaphor can be defined formally for implementation purposes, divided into a transition diagram and a mapping function. We can tie them together for a complete definition.

Definition 3 (3D metaphor): *A* 3D metaphor *explains the handling of a 3D operation with a two dimensional input device. It consists of two parts: the* metaphorical concept *that describes the structural mapping from the user's mental model into the human-computer interface, and the* implementation *that maps interface events into the three-dimensional mathematical model of the application. To facilitate establishing a mental model, the metaphorical concept relates to the user's previous experience with three dimensions.*

Let us focus for a moment on metaphors in their usual meaning. [Lakoff and Johnson, 1980] talk about *metaphorical concepts*, because a metaphor deals with a whole domain, mapping concepts from this well known domain from which the metaphor is chosen into a new context to which the metaphor is applied. We can analyse *specificity, clarity, richness* and other topics mentioned in [Carroll and Mack, 1985]. On the other hand, we should avoid treating them as *analogies* that map all concepts of a domain. Metaphors do only map selected concepts and allow exceptions and extensions [Halasz and Moran, 1982].

A 3D metaphor performs the mapping from the user's mental model about 3D interaction into the 2D user interface actions. The implementation of a 3D metaphor performs the mapping from the 2D user interface events into the 3D operations of an application. The question how the mental model of 3D interaction for a user looks like is addressed for example in [Shepard and Cooper, 1986, Carlton and Shepard, 1990]. The results concerning *mental rotations* and also translation and scaling tasks indicate that the user has an analogous 3D model in contradistinction to a digital symbolic model. If the mental model of the user is isomorphic to the 3D mathematical model of the application in some sense, we can identify the metaphor as the inverse mapping of its implementation.

In our work, we are especially interested in the first part of our definition. What qualities of a 3D metaphor are observable from the user's point of view, and how do they influence the usability of the 3D metaphor? This will mostly determine the implementation. We have therefore analyzed 3D metaphors found in literature within this framework. An overview is given in the next section. Then we present a classification scheme with a feature space based on metaphorical concepts we have found.

1.5 Known 3D interaction techniques

For many 2D interaction techniques there exists a canonical extension to 3D. Translation extends easily, if we project the 2D position on a plane, *e.g.* the xy-plane centered in an object, to achieve a 3D position. We can use direct manipulation in the 3D graphics display or a 2D indirect manipulator in the usual 2D graphical user interface around. They are well documented with several other 3D metaphors in [Foley et al., 1990].

[Johnson, 1963] explains the system *Sketchpad III*, the first computer application known with 3D interaction. It uses a lightpen and rotary potentiometers to manipulate in three orthogonal and one perspective view of the 3D scene. The system already includes a snapping technique with a sphere or cylinder shape for the 3D cursor.

[Thornton, 1979] describes NUMBER-WHEELS, a kind of incremental slider implementation with a graphics tablet that can model infinitely large rotation angles. He associates a mass to the wheel, so that rotations continue with the speed they have had at the end of an interaction. A consequence of this metaphorical concept is that friction will slow down the wheel. We call it SPIN-EFFECT, as it is known for rotational metaphors.

[Evans et al., 1981] concentrate on 3D metaphors for rotations. The TURNTABLE denotes rotations around the axis perpendicular to the screen. A reference point is needed to rotate around. The STIRRER analyzes the mouse path to detect circular movements and maps the angle to a rotation similar to the TURNTABLE. In difference, no absolute reference point is needed and two degrees of freedom, *i.e.* linear movements, are free to be mapped to rotations around the two remaining axes. This leads to gesture recognition which is a concept that provides many degrees of freedom, see [Rubine, 1991, Bröckl and Hartenstein, 1993] for details.

[Chen et al., 1988] is the source of our example above. The paper compares four metaphors for rotation in a user experiment. Three of them are slider arrangements drawn in the 3D graphics area. The fourth and winning metaphor is the VIRTUAL-SPHERE. The second experiment compares it with the combined metaphor from [Evans et al., 1981] above. No significant difference in performance time and angular accuracy is reported.

An even simpler metaphor is the ROLLING-BALL. It maps the x and y coordinates of the mouse movements to incremental rotations around the y- and x-axis respectively. One might expect that this restricts the metaphor to a two-dimensional subset of rotations, but this is false. [Hanson, 1992] explains in detail the surprising fact: If a sphere is rotated in clockwise order using a flat object at its top, which equals the incremental style for rotations, the sphere itself will rotate counterclockwise. Therefore, all three degrees of freedom can be manipulated.

Two rotation metaphors are reported in [Bröckl et al., 1992]. The first is a derivation of the VIRTUAL-SPHERE. The relative mouse movement is separated into an angular and a radial component as for polar coordinates. The radial component is used for the VIRTUAL-SPHERE, the angular component is used for the TURNTABLE. Thus, any movement around the center rotates around the normal axis of the screen. The second one is called GLOBE-metaphor. A point is selected on an imaginary globe. Here, the new viewpoint is located, and the new viewing direction points to the center of earth.

[Nielson and Olsen, 1986] introduce a new translation metaphor with a segmen-

Fig. 3a: Segments around a centroid for a translation metaphor.

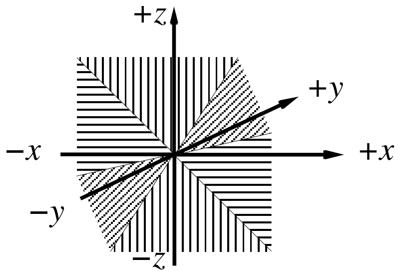
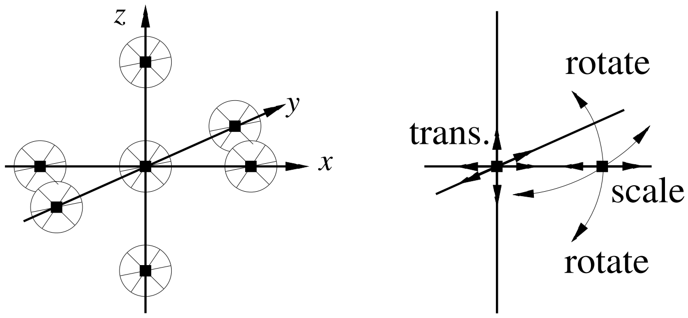


Fig. 3b: The combination of handles and segmentation results in a fairly complex metaphor



tation around the mouse pointer. The orientations of the segements are aligned to the projection of the local coordinate frame of the object to be manipulated, as depicted in figure 3a. The metaphor behaves like a pie menu; the direction of the mouse movement selects the direction in which to translate. [van Emmerik, 1990] extends the concept to seven handles, as shown in figure 3b, providing translation, rotation and nonuniform scaling operations in one metaphor. [Conner et al., 1992, Snibbe et al., 1992, Zeleznik et al., 1993] implement these concepts of handles and pie menu segments in their 3D widget set.

[Houde, 1992] goes a step further and makes the operation behind a handle visible to the user by drawing the handle as a hand symbol that looks like performing this operation. The paper presents a stepwise refinement of a prototypical furniture placing application. Results from several informal user tests are reported that influence the design.

[Nielson and Olsen, 1986] also present interaction techniques for translation, rotation, and scaling that are based on selecting points on surfaces or edges of objects. The techniques are defined through multiple points, *e.g.* two points define a translation from one to the other. This approach differs from the previous ones in the fact that it allows precise drawings in distinction to fuzzy sketching techniques. Further work on precise interaction techniques is documented in [Bier, 1986, Bier, 1990]. A complete drawing system is presented.

An important, specialized task is the viewpoint navigation. [Mackinlay et al., 1990b] describe a LOGARITHMIC-FLIGHT-metaphor that provides effective viewpoint navigation in three space with only a 2D pointing device. The complex WALKING-metaphor needs three modal states to achieve a five-degrees-of-freedom control [Mackinlay et al., 1990a, Robertson et al., 1993]. Two states are represented by small transparent icons in the visual field. The third state will be entered during interaction outside these icons.

1.6 Formal Approaches

[Chen et al., 1988, Houde, 1992] have done empirical examinations to compare different *ad hoc* design ideas, sometimes justified through usability tests, see previous section. A slightly different approach is reported in [Grissom and Perlman, 1993]. A whole interface with several interaction tasks is evaluated, not only to compare the performance of different techniques, but also to test the integration in the interface; how well the techniques fit together. Nothing specific is said in advance about interaction techniques. Only a set of basic tasks is defined, over which the performance has to be evaluated.

[Mackinlay et al., 1990a, Card et al., 1991] are motivated from taxonomies of input devices, but they reach a level with their classification scheme that is capable to describe 3D metaphors. Their key example is the WALKING-metaphor. A great advantage of this approach is the unified framework dealing with physical device properties and logical parameter mappings. The complete interaction process from physical movements up to operations in the application can be modeled. But this strength is also its weak point. The scheme is not expressive enough for our purposes: in particular, there are no interface definitions, what is physical and what is logical. It is difficult to model the dialog control in this scheme. The dialog flow is determined through a mixture of physical device properties and application behavior. A minor point is that the layout transformation is not precise enough, *i.e.* the transformation model is not clear and rotations around arbitrary axes are missing.

Earlier taxonomies for logical input devices can be found in [Foley et al., 1990]. They provide a crude classification in locator, pick, valuator, keyboard and choice devices, also used in graphical system standards like GKS or Phigs. They are not able to describe 3D metaphors in sufficient depth.

2 Classification Scheme

We adopt the idea of morphological analysis from [Mackinlay et al., 1990a] and decompose 3D metaphors into elementary metaphors that form more complex ones with a small set of composition operators. If we look at the dimension of the mapping, we can decompose all 3D metaphors to one dimensional mappings. These are *elementary metaphors*. Examples are a simple 1D slider or the TURNTABLE.

To collect 3D metaphors, we reviewed papers from the computer graphics, the human computer interaction, the ergonomics and cognitive science field. We have found 42 nicely distinguishable 3D metaphors to build up and justify a feature space. An overview is presented in the previous section.

The analysis by decomposing metaphors sometimes results in elementary ones that are useless and difficult to understand as a stand-alone interaction technique. Thus, we start presenting the composition operators, continue with the classification scheme for composed and elementary metaphors, and finish with the description of all features. But first of all, we want to clarify our method and criteria to build this classification scheme.

2.1 Feature space criteria

As explained in the introduction, we want to distinguish 3D metaphors by their qualities. The classification scheme consists of features, any of them describing a quality, all together forming the feature space. Any 3D metaphor will be classified as a point in this feature space, but not all points might represent reasonable metaphors. So, how to select the features?

- We have chosen a user centered approach. Features will only be allowed, if they can be observed using the interface without looking at the implementation of the 3D metaphor.
- For any feature, there should be at least one example of two different metaphors, so that they are distinguishable with this feature.
- For any two different metaphors should exist at least one feature that distinguishes between them.
- No feature should be reducable from a set of other features.

The decision what metaphors are different depends on subjective judgements and will influence how fine or coarse the classification scheme will be.

A weak point in the theory is the lack of foundation from cognitive science for the metaphorical concepts we are going to use in the classification. So they are chosen as a result of an orthogonal decomposition of the metaphors as it is common in computer science. It is not sure if it is even user independent, but this decomposition is in principle supported by the empirical results reported from [Shepard and Cooper, 1986] and the theoretical considerations in [Carlton and Shepard, 1990]. One important point is that a 3D rotation around the view axis is not significantly faster than rotations around arbitrary axes in 3D. One might expect a performance advantage, because the task projected to 2D seems easier, but this is not the case for rotations with three-dimensional objects. So the decomposition of a rotation into an arbitrary rotation axis and a rotation angle is fairly reasonable, especially when compared to Eulerian angles with three fixed axes in arbitrarily chosen order. The analog nature of parameters like angles and distances is also stated in [Shepard and Cooper, 1986].

2.2 Composed and elementary metaphors

A *composed metaphor* is recursively defined by a composition technique that fits several 3D metaphors together. Two composition types can be distinguished: temporal ordering and spatial grouping. Parallel (interleave), alternative, and sequencial grouping of events are well known in the description of the temporal behavior of a dialog system.

Spatial grouping gives access to different dialogs at different places of the graphical user interface. It can be subdivided according to the reference system involved.

Handles are an example for marked areas in object space, segmentations are examples in world space, fixed to screen coordinates, and pie-menu-like selections are connected to the relative reference system of the mouse pointer. Examples are given in the review above.

Spatial grouping occurs in conjunction with the alternative or parallel dialog control—the first variant with and the second without state change in the state transition diagram. E.g., a pie menu with alternative dialog control will choose the selected menu item once at the beginning of interaction, while the parallel dialog control chooses again at every mouse movement.

Classification (composed metaphor): *A* composed metaphor *is characterized by a five-tuple consisting of the composition operator over a set of 3D metaphors as operands, a set of metaphorical concepts as mentioned in the definition af a 3D metaphor, a seven-tuple of subitems to distinguish a whole bunch of variants, and the prompt and feedback styles that describe their perception quality. The complete classification with all possible choices for the features is as follows:*

<NAME> = (
    (*parallel* | *alternate*)(*pie menu* | *segments* | *handles*) |
        parallel | *alternate* | *sequence:* {<NAME1>, <NAME2> ...},
    concepts $\subseteq$ {line, plane, multiple views, number-wheel-angle,
            number-wheel-axis, turntable-angle, turntable-axis,
            rotation-axis, orientation, parameter},
    variant = <EXTENDED VARIANTS>,
    prompt $\subseteq$ {syntactic symbolic, syntactic 2D, syntactic 3D,
            semantic, none},
    feedback $\subseteq$ {syntactic symbolic, syntactic 2D, syntactic 3D,
            semantic, none}
)

<EXTENDED VARIANTS> = (
    orientation $\subseteq$ {parallel, arbitrary},
    surrogate $\subseteq$ {reference frame, wheel, sphere,
            bounding box, handles, none},
    synchronisation $\subseteq$ {movement, direction, none},
    transformation $\subseteq$ {camera in hand, world in hand},
    integration $\subseteq$ {position, speed, acceleration},
    spin effect $\in$ {yes, no},
    commutativity $\in$ {yes, no}
)

The metaphorical concepts and the other features will mostly depend on the 3D metaphors used to built the composed one. Features from the type subset are unions, the *spin effect* and the *commutativity* feature are logically anded from the underlying metaphors' features. The *concepts* and the *surrogate* features are not strictly unions. Two line concepts orthogonal to each other and belonging to the same operation will be substituted through the plane concept. The multiple view concept is unique and will occur at a certain level of complexity. The sphere as a selected object is constructed from two orthogonal wheels. All features are explained in the next section. Examples are given thereafter.

An elementary metaphor is reduced to only one-dimensional mappings. We include the semantic operation that the metaphor performs in the application to distinguish cases in which the metaphorical concept is the same; *e.g.* scaling and translating can often use the same concepts.

Classification (elementary metaphor): *An* elementary metaphor *is characterized by a five-tuple consisting of the operations type that is performed from the semantic operation of the 3D application, the metaphorical concept, a six-tuple of subitems to distinguish the variants, and the prompt and feedback styles that describe their perception quality. The complete classification with all possible choices for the features is as follows:*

<NAME> = (
    operation $\in$ {translation, rotation, uniform scaling, scaling,
        selection, parameter},
    concept $\in$ {line, number-wheel-angle, number-wheel-axis,
        turntable-angle, turntable-axis, rotation-axis, parameter},
    variant = <VARIANTS>,
    prompt $\in$ {syntactic symbolic, syntactic 2D, syntactic 3D,
        semantic, none},
    feedback $\in$ {syntactic symbolic, syntactic 2D, syntactic 3D,
        semantic, none}
)

<VARIANTS> = (
    orientation $\in$ {parallel, arbitrary},
    surrogate $\in$ {reference frame, wheel, sphere,
        bounding box, handles, none},
    synchronisation $\in$ {movement, direction, none},
    transformation $\in$ {camera in hand, world in hand},
    integration $\in$ {position, speed, acceleration},
    spin effect $\in$ {yes, no}
)

2.3 Description of all features

2.3.1 Operation

The operation is determined from the semantic operation of the 3D graphics application. It helps in distinguishing among metaphors, if the metaphorical concept is applicable to different operations. It serves as the main feature which tells what the metaphor is good for.

2.3.2 Concept

The metaphorical concept is mentioned in the introduction. It roughly tells how to use this interaction technique. This is done with references to the user's experience with the real world. The concept is responsible for building the mental model in the user's mind.

Line: denotes interaction with the reduced degree of freedom to one on a straight line. It can be applied to operations like translation, selection or scaling.

Plane: denotes interaction with the reduced degree of freedom to two dimensions on a plane. It can be applied to the same operations as for the line concept, but only in composed metaphors as can be seen from the dimension two.

Multiple views: Different viewpoints enable the selection of a line or a plane in one view that serves for the *line* or *plane* metaphorical concept used for interaction in another view. If multiple views are not supported by the user interface of the application, dialog techniques can switch between different views.

Number-wheel-angle: The number-wheel-concept describes the illusion of manipulating the cylindrical surface of a wheel with its axis placed in the plane of the 2D interaction. So, a linear movement is mapped to the angular parameter of a rotation. Only the projection of the pointer movement onto the direction of the wheel (perpendicular to its axis) is mapped.

Number-wheel-axis: In addition to the angle-concept, the axis-concept states that the axis of rotation is perpendicular to the pointer movement. The axis-concept determines the rotation axis and the angle-concept determines the angle of rotation.

This is an example of a more or less useless concept to build a stand-alone metaphor. It works only in conjunction with the number-wheel-angle concept and follows the decomposition of a rotation in axis and angle.

Turntable-angle: The turntable-concept describes the illusion of a generalized wheel with its axis arbitrarily placed outside the plane of 2D interaction. The name is derived from the view from the top onto a turntable where the rotation axis is perpendicular to the plane of 2D interaction. There, the angle is derived from the polar angle of the circular component of movements around the known center of the wheel. In general, the rotation describes elliptic curves around the known center. In distinction to the number-wheel, the disc at the side of the wheel is manipulated.

Turntable-axis: The extension of the turntable-angle-concept to the axis-concept is more difficult than for the number-wheel. But it follows the same rule, namely that the angle-concept determines the angle of rotation whereas the axis-concept determines the rotation axis. In conjunction, both concepts result in the VIRTUAL-SPHERE-metaphor, our example above. Thus, the axis-concept derives the rotation axis from the fact that the observed pointer movement must describe an elliptic curve around the known center of rotation. Again, the axis is perpendicular to the pointer movement if pictured in 3D.

Rotation-axis: This concept uses the pointer movement direction as the orientation of the rotation axis. Nothing is stated about the rotation angle.

Parameter This concept is derived from the mathematical background of the semantic operations that use one dimensional parameters as the most general concept. This concept can be used for all metaphors, but it is only recommended for situations where no better real world concept can be found than the underlying mathematical description of the operation.

2.3.3 Variant: orientation

The 3D metaphor often implies a reference frame associated with the manipulated object. The *orientation* of this reference frame can be aligned to the plane of the 2D interaction, in other words to the screen space reference frame. The precise definition states that the z-axis of the reference frame must be parallel with the surface normal of the 2D interaction plane to call the object reference frame to be aligned with the screen reference frame.

Extensions of 2D interaction techniques are typical candidates for aligned reference frames. The techniques from [Nielson and Olsen, 1986] take their benefits from non-alignment.

2.3.4 Variant: surrogate

The manipulated object can be substituted by a simplified representation, the *surrogate*, for direct manipulation. This makes interaction easier and conforms to user's expectations that different looking objects should nevertheless behave similar [Houde, 1992].

2.3.5 Variant: synchronisation

The mapping from the hand movement, in this case represented through the pointer movement to achieve a more abstract treatment, to the 3D movement can conform to the request of *kinesthetic correspondence* [Pique, 1986], also called *stimulus response* *compatibility* [Foley et al., 1990]. We distinguish between synchronous movement, where the pointer and the selected 3D point move identically in the 3D graphics area, synchronous direction, where the direction of movement is similar, but the amount of movement differs, and asynchronous movement otherwise. The exact amount of the scale factor from pointer to 3D movement is not used for classification. Also, the control-to-display ratio from input device movement to pointer movement is ignored, because it belongs to the lexical input handling in the UIMS and not to the 3D metaphor. Nonetheless, it will influence the overall behavior of kinesthetic correspondence.

A nice, non-trivial example is the VIRTUAL-SPHERE-metaphor. [Chen et al., 1988] simplify the projection on the sphere to a linear mapping. The metaphor achieves therefore synchronous direction, while the description from [Hultquist, 1990], as presented in the introduction, results in a synchronous movement. The ROLLING-BALL-metaphor achieves also synchronous direction, while metaphors with absolute control of Eulerian angles are asynchronous.

2.3.6 Variant: transformation

Two different transformation types are known. The metaphor manipulates the selected object according to the CAMERA-IN-HAND- or the WORLD-IN-HAND-metaphor [Ware and Osborne, 1990].

2.3.7 Variant: integration

Usually, the position of the pointer is mapped to the parameters of the semantic operations. Additionally, integration of the position over the time enables interaction techniques with speed or acceleration controls. An example is the FLY-THROUGH-metaphor [Ware and Osborne, 1990]. The usage of derivations for 3D metaphors was not observed, though it was usually used in the mapping from mouse to pointer movements.

2.3.8 Variant: spin effect

Modern graphics performance allows animation in interaction techniques. The SPIN-EFFECT continues the semantic operation with the speed known at the end of interaction. So, the interaction technique proceeds beyond one input cycle.

2.3.9 Variant: commutativity

Composed metaphors add one new variant: Does the implementation of the metaphor *commute* or not? This determines, whether the result of an interaction only depends on the endpoint or on the whole path of the movement. The question can be reformulated, do all closed path movements lead to the identity transformation? Thus, a metaphor commutes if the transformations used in the semantic interface do commute. It is reasonable that 3D metaphors without continuous feedback have to be commutative.

2.3.10 Prompt and feedback

Prompt and feedback can be given at different levels of interaction. UIMS distinguishes between the lexical, the syntactical and the semantical level [Olsen, 1992]. Therefore, 3D metaphors cannot influence prompt or feedback for user interface elements at the lexical level. At the syntactical level, prompt and feedback can be distinguished through their representation in symbolic, 2D or 3D graphical form [Hübner, 1990]. The semantic level can also be used for prompt or feedback, *e.g.* the metaphor provides continuous feedback by feeding every input event to the semantic operation.

2.4 Selected examples

Many descriptions of 3D metaphors in the literature lack information to classify them completely, though any feature has significant influence on the behavior of a metaphor. Thus, we choose a VIRTUAL-SPHERE implementation that is used in the demo programs from Silicon Graphics as an example to classify:

VIRTUAL-SPHERE = (
  parallel: {TURNTABLE-ANGLE, TURNTABLE-AXIS},
  concepts: {turntable-angle, turntable-axis},
  variant = (
    orientation: {arbitrary},
    surrogate: {sphere},
    synchronisation: {direction},
    transformation: {world in hand},
    integration: {position},
    spin effect: yes,
    commutativity: no
  ),
  prompt: {none},
  feedback: {semantic}
)

Two elementary metaphors are used, one for the angle and one for the axis. They are quite similar. Replacing axis for angle in the following classification of the TURNTABLE-ANGLE gives us the TURNTABLE-AXIS-metaphor.

TURNTABLE-ANGLE = (
  operation: rotation,
  concept: turntable-angle,
  variant = (

orientation: arbitrary,
surrogate: wheel,
synchronisation: direction,
transformation: world in hand,
integration: position,
spin effect: yes
),
prompt: none,
feedback: semantic
)

A more complex example is the WALKING-metaphor, as described in [Mackinlay et al., 1990a]. The metaphor controls five degrees of freedom. Thus, we can expect a minimum of five elementary metaphors. First, the metaphor uses a segmentation of the screen space into two icon regions and the exterior. One icon stands for a 2D orientation task for the view direction. The other icon symbolizes a panning operation perpendicular to the view direction. The exterior region allows walking in the view direction and turning to the left and right.

WALKING = (
alternate segment: {EYE-CONTROL, PAN-CONTROL, WALK-CONTROL},
concepts: {line, plane, number-wheel-angle},
variant = (
orientation: {parallel},
surrogate: {wheel, sphere},
synchronisation: {direction, none},
transformation: {camera-in-hand},
integration: {speed},
spin effect: no,
commutativity: no
),
prompt: {syntactic symbolic},
feedback: {syntactic 2D}
)

Several features have a set of more than one entry, due to the different behaviors of the metaphors from which the composed metaphor is built from. The EYE-CONTROL uses a sphere as a surrogate and synchronous direction. The PAN-CONTROL uses no surrogate instead of the selected object, the viewpoint in this case. The WALK-CONTROL uses the wheel, the selected object and asynchronous control for the walking. All three are parallel compositions. We use an operator symbol '$\times$' for a short hand notation:

EYE-CONTROL = X-EYE-CONTROL $\times$ Y-EYE-CONTROL
PAN-CONTROL = X-PAN-CONTROL $\times$ Y-PAN-CONTROL

WALK-CONTROL = Z-WALK-CONTROL × Y-ROTATE-CONTROL

The z-axis points towards the viewer, the y-axis upwards. The elementary metaphors for eye control and rotation are identically, dito for both pan control metaphors. But the Z-WALK-CONTROL stands alone with its asynchronous movement.

X-EYE-CONTROL = Y-EYE-CONTROL = Y-ROTATE-CONTROL = (
operation: rotation,
concept: number-wheel-angle,
variant = (
orientation: parallel,
surrogate: wheel,
synchronisation: direction,
transformation: camera-in-hand,
integration: speed,
spin effect: no
),
prompt: syntactic symbolic,
feedback: syntactic 2D
)

X-PAN-CONTROL = Y-PAN-CONTROL = (
operation: translation,
concept: line,
variant = (
orientation: parallel,
surrogate: none,
synchronisation: direction,
transformation: camera-in-hand,
integration: speed,
spin effect: no
),
prompt: syntactic symbolic,
feedback: syntactic 2D
)

Z-WALK-CONTROL = (
operation: translation,
concept: line,
variant = (
orientation: parallel,
surrogate: none,
synchronisation: none,
transformation: camera-in-hand,

integration: speed,
spin effect: no
),
prompt: none,
feedback: syntactic 2D
)

3 Discussion

The classification scheme is put to the test when a previously unknown metaphor emerges. [Shoemake, 1992] presents the ARCBALL-metaphor. Similar to the VIRTUAL-SPHERE-metaphor, the axis and the angle are determined with $a \times b$ and $a \cdot b$ respectively, with the slight difference that not the formula for rotation around an axis is used, but a quaternion $q = [a \times b, a \cdot b]$. A quaternion represents also a rotation but twice as fast as in the VIRTUAL-SPHERE-metaphor (see [Shoemake, 1991] for details about the usage of quaternions to represent three-dimensional rotations). The second difference is that the transformation is not applied continuously. Instead, the orientation of the object at the beginning of the interaction is preserved and absolutely transformed with the quaternion. Therefore, the metaphor is commutative. Here is the classification, based on the demo implementation; the elementary metaphors are omitted:

ARCBALL = (
parallel: {ARCBALL-ANGLE, ARCBALL-AXIS},
concepts: {turntable-angle, turntable-axis},
variant = (
orientation: {arbitrary},
surrogate: {sphere},
synchronisation: {direction},
transformation: {world-in-hand},
integration: {position},
spin effect: no,
commutativity: yes
),
prompt: {syntactic 2D},
feedback: {syntactic 3D}
)

The general idea how to achieve commutative metaphors can also be applied to the synchronous variant of the VIRTUAL-SPHERE. The result is a new metaphor. Other examples are presented in the next subsection, where we discuss how the classification scheme can be used to find new metaphors. Afterwards, we analyse the synchronisation feature in greater detail and discuss a theoretical approach for a performance prediction.

3.1 Examples of new 3D metaphors

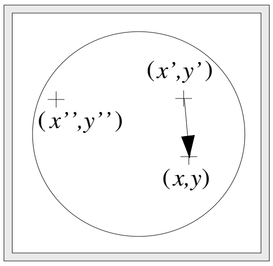
Several methods can be used to investigate the classification scheme for new metaphors: First, all feasible features for existing metaphors can be analyzed. Second, the composition operators can be applied to all possible combinations of 3D metaphors. Third, the domain of a feature can be extended. The most promising feature is surely the metaphorical concept. The last possibility is the discovery of other useful features. Examples follow for the first three cases.

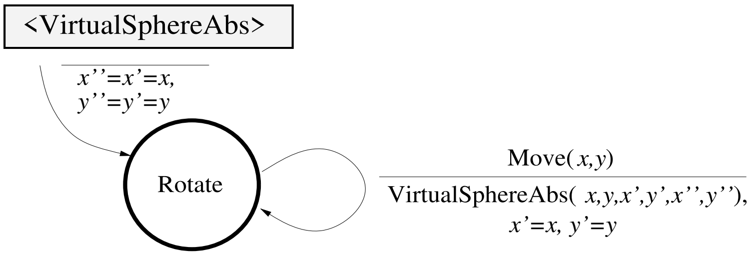
Most rotation metaphors are not commutative due to their incremental implementation style. We are interested in a conversion to commutative ones. A benefit is obvious for interaction techniques that are not able to give appropriate feedback. We present a general construction method. Assume a transformation model where the transformation matrix $\boldsymbol{M}$ is consecutively adapted with each input event through a matrix multiplication from the right $\boldsymbol{M} = \boldsymbol{M}'\,\boldsymbol{R}$. Previous values are primed. $\boldsymbol{R}$ is typically determined by the last two input events.

Theorem 4: *Any non-commutative 3D metaphor can be transformed to a commutative one.*

Proof: We initialize $\boldsymbol{R}'$ with the identity transformation I at the beginning of interaction, calculate the transformation $\boldsymbol{R}$ to be an absolute transformation, *i.e.* with respect to the starting point of interaction with transformation $\boldsymbol{M} = \boldsymbol{M}_0$, and rewrite the transformation rule above to $\boldsymbol{M} = \boldsymbol{M}'\,\boldsymbol{R}'^{-1}\,\boldsymbol{R}$, which still fits in our transformation model. Now we proceed: Initially, $\boldsymbol{M} = \boldsymbol{M}_0\,I^{-1}\,\boldsymbol{R} = \boldsymbol{M}_0\,\boldsymbol{R}$. Now assume that $\boldsymbol{M}' = \boldsymbol{M}_0\,\boldsymbol{R}'$. The next transformation step is then $\boldsymbol{M} = \boldsymbol{M}'\,\boldsymbol{R}'^{-1}\,\boldsymbol{R} = \boldsymbol{M}_0\,\boldsymbol{R}'\,\boldsymbol{R}'^{-1}\,\boldsymbol{R} = \boldsymbol{M}_0\,\boldsymbol{R}$. Thus, $\boldsymbol{M}$ depends only on the current transformation $\boldsymbol{R}$ and the starting transformation $\boldsymbol{M}_0$.$\square$

The VIRTUAL-SPHERE from section 1.2 is used as an example. We do not store $\boldsymbol{R}'$. It is calculated from previous events. The left picture shows three events: (x'', y'') is the 2D projection of the 3D point a where interaction starts, (x', y') is the point b of the last event, and (x, y) is the current point c.



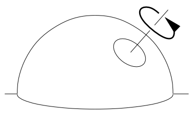


$$\begin{aligned}
&\mathrm{VirtualSphereAbs}(x, y, x', y', x'', y'') := \\
&\qquad \boldsymbol{M} = \boldsymbol{M}'\,\boldsymbol{R}_{axis}(a \times b, -\arccos(a \cdot b))\,\boldsymbol{R}_{axis}(a \times c, \arccos(a \cdot c)), \\
&\qquad a = \Pi(x'', y''),\ b = \Pi(x', y'),\ c = \Pi(x, y).
\end{aligned}$$

The second technique to find new metaphors looks for new combinations with composition operators. All combinations of operators and metaphors so far known give us several exotic combinations like rotation in x-direction with scaling in y-direction, but also a few interesting examples. Some are based on sequential combinations with the ROTATION-AXIS- and ORIENTATION-metaphor. We use '$\rightarrow$' as a shorthand notation.

NEW1 = ROTATION-AXIS $\rightarrow$ NUMBER-WHEEL-ANGLE
NEW2 = ORIENTATION $\rightarrow$ TURNTABLE-ANGLE
NEW3 = ORIENTATION $\rightarrow$ STIRRER

It is a nice exercise to figure out a reasonable interpretation of these metaphors that are new, as far as it is known to me. Let's try: NEW1 uses the pointer position at the time of the state change from the left to the right metaphor together with a known reference point at a center to determine a rotation axis in the plane. So far, it is the rotation-angle concept. The angle is then measured with the NUMBER-WHEEL-metaphor. The associated concept states orthogonality. Thus, the direction from which to measure the angle is also determined. The implementation is fixed.



NEW2 and NEW3 start with the ORIENTATION-metaphor that uses the projection onto the unit sphere to get a orientation from an input point. The orientation is represented as a unit vector in the sphere center. It can be used as a rotation axis that is no longer restricted to the plane. Therefore, the TURNTABLE-ANGLE-metaphor is the natural choice, but the STIRRER-metaphor is also useful, because it has no singularity in the center, which is the starting point in this case. The figure on the right demonstrates the metaphor: A point is selected that induces an axis around which the rotation can be performed in a natural way.

An interesting point is that constrained rotations can easily be represented by points on the sphere denoting the constrained rotation axis. Snapping techniques can be used to select such a constrained rotation. Even such an unusual idea as a grid for rotations is easily accessible here. Other approaches like [Shoemake, 1992] draw great circles to represent the restricted rotations.

All three examples focus on the direction of the rotation axis, while the previously mentioned rotation metaphors emphasize the orthogonality principle to choose the axis. Different tasks may result in different preferences between these techniques. Real world examples are screw drivers, valves, and cranks.

3.2 Analysis of the synchronisation feature

An interpretation problem occurs for the VIRTUAL-SPHERE-metaphor in the case of the synchronous direction. E.g., if a linear mapping is used, as in [Chen et al., 1988], the selected point will follow the pointer movement for a while, until it reaches a rotation of $\pi/2$ where the direction of the point reverses. The same problem occurs for the ROLLING-BALL metaphor. [Hanson, 1992] suggests therefore to turn off the pointer and emphasize the point on top of the sphere as the manipulated point throughout the whole interaction to avoid this distraction.

The manipulated geometric shape is a sphere for synchronous movement. To get a shape for the synchronous direction with the linear mapping, we drop the perpendicular from the pointer position and intersect it with a ray starting from the center with the angle calculated from the pointer position. Accordingly scaled, we get the unit sphere in the known case of synchronous movement. The synchronous direction results in the shape $z = r/\tan(r\pi/2)$, where r denotes the distance of the pointer position from the point above the center. The shape is rotationally symmetric. It has an extreme point at $r = 2$; the rays intersect never for angles $\pi \leq \phi \leq 2\pi$.

An interesting compromise between both techniques arises with the selection of a geometric shape and the calculation of the mapping from the pointer position to the angle from this shape. We choose a parabola $z = 1 - r^2$. Intersecting the perpendicular with this shape results in the mapping $\phi = \arctan(r/(1 - r^2))$ for the angle ϕ. We achieve synchronous movement for small rotations near the point above the center, because the parabola touches the sphere in this point. The shape for synchronous direction has its maximum with $z = 2/\pi$ as the limit for $r \to 0$, which is far away from 1 for the two other shapes. On the other hand, all three shapes intersect for $\phi = \pi/2$. This means that a movement from the center to the diameter will result in a rotation angle of $\pi/2$ for all three metaphors. The scaling factors are similar. In contradistinction, the ARCBALL-metaphor gives here a rotation angle of π. Another property of our geometry is that there always exists an intersection—there is no singularity for $\phi = \pi$. Indeed, the mapping never reaches π. A selected point will not reappear on the wrong side after it has disappeared on one side (assuming hidden surface removal). The control ratio for this new metaphor behaves fine within the diameter, and will decrease gracefully for pointer moves going to infinity.

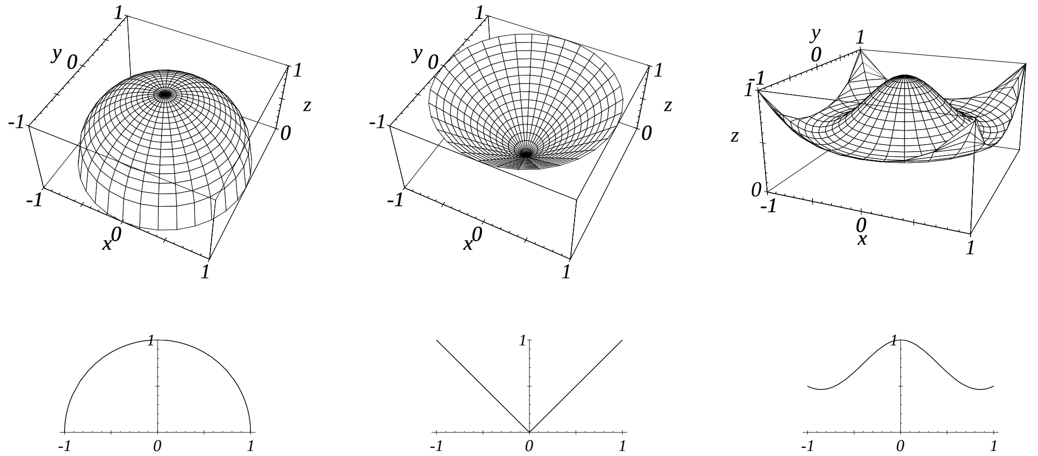
3.3 Control precision

[Nielson and Olsen, 1986] introduce the *control precision* to describe the projective shortening of a direction. They use it as an indicator for difficulty of interaction tasks along this direction. We are interested to apply this idea to rotation metaphors and therefore give a slightly different definition:

Definition 5 (control precision c): *The control precision c is defined for a pointer position (x, y) and the movement of a selected point in three space according to a function $f(x, y)$ with respect to the pointer as the reciprocal of the absolute value of the partial derivations:*

$$c = \frac{1}{\left|\frac{df}{dx}, \frac{df}{dy}\right|} \quad \in \mathbb{R}^+ \tag{1}$$

Figure 4: The control precision is drawn (from left to right) for the normal VIRTUAL-SPHERE-, the TURNTABLE-, and the VIRTUAL-SPHERE-metaphor with the new parabolic shape. A 3D plot in the upper half unit cube and a 2D intersection is shown.



For a simple translation metaphor like that from [Nielson and Olsen, 1986] with synchronous movement, we get the control precision $c_t = v \cdot w$, where v is a unit vector in the direction of the pointer movement, and w is a unit vector in the direction of the selected point movement. Control precision $c = 1$ denotes a one to one control. Control precision $c = 0$ denotes singularities in the mapping; arbitrarily small pointer moves result in large object transformations. In the example of the translation metaphor, the move direction of the selected point vanishes to a single point in the projection.

To deal with the angle of rotation metaphors, we use a selected point from the unit sphere. The definition from above simplifies to the paramter r in polar coordinates due to rotational symmetry. The control precision for the VIRTUAL-SPHERE-metaphor with synchronous movement is $c_1 = \sqrt{1 - r^2}$, see figure 4. The half sphere illustrates the singularities at the diameter and the good control ratio within a great range. The same metaphor with the simplified linear mapping, hence only synchronous direction, has a constant control precision of $c_2 = 2/\pi$.

At this point, the observation is that synchronous movement, as demanded from the direct manipulation paradigm, and the avoidance of singularities exclude each other.

Two more control precisions are depicted in figure 4: first, the TURNTABLE-metaphor, and second, the VIRTUAL-SPHERE-metaphor with the modified, parabolic shape. No singularities and a fine control behavior can be observed for the second one. The control precision is $c = (1 - r^2 + r^4)/(1 + r^2)$.

3.4 Performance prediction

Fitts' law is a promising tool for performance prediction. It was succesfully applied to 2D mouse pointing tasks [MacKenzie and Buxton, 1992]. A detailed description is given in [MacKenzie, 1992]. Different versions of this law are in use. The principle observation is that the time T needed for a pointing task is linear to an *index of difficulty* ID: $T = a + b\, ID$. The constants a and b are empirically determined and available for different muscle groups or the hand/forearm in combination with an input device. The index of difficulty ID is dependent from the distance A to move (amplitude) and the target width W, which is the tolerated error measured as an interval around the target point: $ID = \log_2(A/W + 1)$. This version of Fitts' law is the best justified through experiments and a theory [MacKenzie, 1992].

When we apply Fitts' law to translation metaphors, we observe that the projective shortenings nullify each other in the fraction. For control precisions tending to zero this cannot be true. Another well known critique is that Fitts' law does not model the influence of the pixel resolution on the screen for small amplitudes A and widths W. So, for translation metaphors this tool is usable for many but not all cases. This extends to all metaphors with linear mappings.

To analyse rotation metaphors, we look at pointer movements starting in the center of the VIRTUAL-SPHERE-metaphor. The amplitude is $A = \sin(\alpha)$ with α as the induced angle assuming the unit sphere. The width W is $W = \sin(\alpha + \gamma/2) - \sin(\alpha - \gamma/2)$. γ is the tolerated error interval around the angle α. A trigonometric equality gives us $W = 2\cos(\alpha)\sin(\gamma/2)$. Identifying A with r, the control precision c from the previous subsection computes to $c = \sqrt{1 - A^2} = \sqrt{1 - \sin^2 \alpha} = \cos\alpha$. The index of difficulty ID for the VIRTUAL-SPHERE-metaphor can now be expressed in terms of the amplitude A, the angular tolerance γ and the control precision c, which is dependent from the position of distance A from the center as depicted in figure 4:

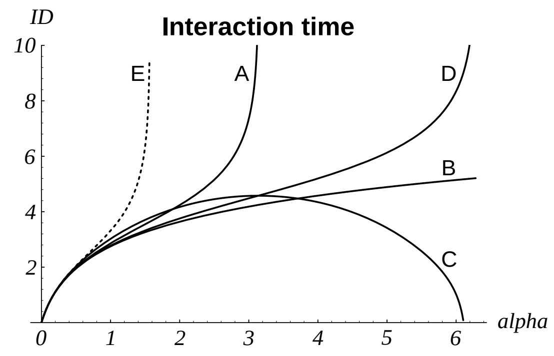
$$ID = \log_2\left(\frac{A}{W} + 1\right) = \log_2\left(\frac{A}{2c\sin\frac{\gamma}{2}} + 1\right).$$

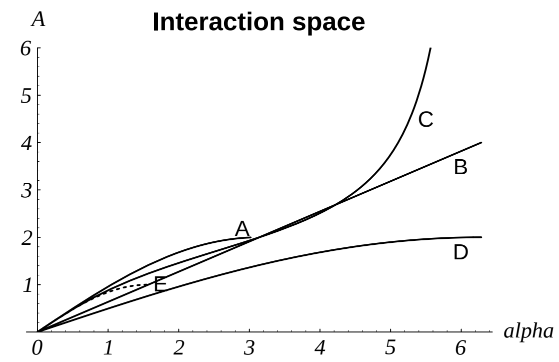
A slightly different movement model is to choose the starting point at the position $-\alpha$. The amplitude A equals to $2\sin\alpha$ and the total rotation angle is now 2α. Calculations result in the identical formula from above. This means that the performance time increase only by a small constant while the angle has been doubled and the accuracy stays the same. Thus, theory states that this movement model is superior than the previous one. It would be nice to see in an experiment, if users behave according to this model.

If the linear mapping is used with the VIRTUAL-SPHERE-metaphor, the control precision $c = 2/\pi$ is introduced as a scale factor in the mapping: $A = c\alpha$ and $W = c\gamma$. Again, we achieve an index of difficulty ID in terms of the control precision c and the amplitude A: $ID = \log_2\left(\frac{A}{c\gamma} + 1\right)$.

For the modified parabolic shape version of this metaphor, no similar result is easy achievable. But in general, if we assume small angular tolerances γ, we can

Figure 5: Interaction time is linear to the index of difficulty ID on the left—given for a fixed angular resolution of ten degree. The space needed for an interaction task of an angle α is shown on the right. VIRTUAL-SPHERE-metaphor with synchronous movement (A), linear mapping (B), parabolic geometry (C), ARCBALL-metaphor (D), and VIRTUAL-SPHERE-metaphor with the movement model starting in the midpoint (E, dashed line).





approximate the mapping from the pointer position to the angle locally through its tangent and achieve $\gamma = \frac{\delta\alpha}{\delta A} W$. Remembering the definition of the control precision c, the approximated index of difficulty ID for rotation metaphors in general is:

$$ID \approx \log_2 \left(\frac{A}{c\gamma} + 1 \right)$$

Thus, the control precision is a good quality criteria for the different areas of a metaphor. To compare different rotation metaphors, we choose the second movement model with the VIRTUAL-SPHERE-metaphor in its synchronous movement variant (A), synchronous direction with the linear mapping (B), the parabolic shape variant developed in the previous subsection (C), and the ARCBALL-metaphor (D). For comparisons with (A), the first movement model with the VIRTUAL-SPHERE-metaphor is drawn as a dashed line (E). The figure 5 shows on the left the exact theoretical interaction times given as the index of difficulty ID over the angle α for a fixed angular tolerance γ of ten degree, and on the right the space needed for the pointer on the screen to achieve such an angle. Other values for γ change only quantity aspects of the figure.

The left figure demonstrates that interaction time increases dramatically near the singularities of (A), (D), and (E). (B), (C), and (D) allows rotation angles greater than π with one stroke. Ignoring (E), one can state that for angles smaller than $\frac{2}{3}\pi$ no significant performance difference should be expected. The decrease in interaction time for (C) for angles greater than π seems curious, but it can be explained that the tolerated width W increases faster than the distance A for large angles. If we compare it with the needed place to do this rotation task in the right figure, we can see that we will not reach the interesting part say of angles greater than 5.5 radians, because limited capabilities in speed and length of our arm will impose side effects to this theory at this point. Fitts' law is invariant under scaling of the whole working area, so if we wish to have the capability to rotate up to five radians with a single stroke, we can use the right figure to determine the unit size in the working area and the interaction times from the left figure are still valid. Under this aspect, it is interesting, if the variants (B), (C), and (D) are really merely undistinguishable in practice.

Some features like singularities or kinesthetic correspondence are not covered here. It is also possible that Fitts' law does not correctly model the complete feedback loop for rotation metaphors. The mapping process from linear movement to rotations may imply some extra burden to the user. Also known is that users are able to do mental translations either in the 3D space or in the 2D projected space, dependent on what the advisor asks for [Shepard and Cooper, 1986]. Effects like this can also influence this analysis. Thus, empirical results are necessary and also interesting for rotations consisting of several strokes.

4 Conclusion

We have presented a formal framework to achieve a uniform description of 3D interaction techniques. We suggest a classification scheme for 3D metaphors and have motivated the feature space through a couple of examples from literature. This scheme was proven to be useful in deriving new 3D interaction techniques and to investigate interesting features. We have discussed in detail the commutativity of composed metaphors and the synchronous movement vs. direction. The new parabolic shape is derived from this. The analysis gives us a quality parameter for metaphors, the control precision. It is nicely linked to the performance prediction of rotation metaphors with Fitts' law. The performance time for selected rotation metaphors is predicted and related to the workspace needed for that.

A couple of questions arise: Is the necessary choice between true direct manipulation and the avoidance of singularities meaningful for the user performance? Is the parabolic shape an interesting compromise? If there is a performance difference between these metaphors, will it possibly be task independent and we can forget the other ones? Can the performance prediction be justified with empirical tests?

Another question concerns composition rules. What conditions make a composition a good composite metaphor? An intuitive suggestion is: keep the feature sets small. Do not combine too much different features. As an example, the WORLD-IN-HAND- and the CAMERA-IN-HAND-metaphor might be confusing when represented simultaneously in a single metaphor.

Acknowledgements

I would like to thank my advisor Ulrich Bröckl-Fox for support and for guidance throughout the development and execution of these studies. I also wish to thank Bernd Gärtner and Emo Welzl for proof reading.

References

[Bier, 1986] Bier, E. A. (1986). Skitters and jacks: Interactive 3D positioning tools. In *Proceedings of the 1986 Workshop on Interactive 3D Graphics, Chapel Hill*, pages 183–196.

[Bier, 1990] Bier, E. A. (1990). Snap-dragging in three dimensions. In *Proceedings of the 1990 Symposium on Interactive 3D Graphics, Special Issue of Computer Graphics, Vol. 24*, pages 193–205.

[Bröckl and Hartenstein, 1993] Bröckl, U. and Hartenstein, K. (1993). Gaphors: Combining gestures and metaphors in order to obtain a more efficient 3D-interface. In *Proceedings of the IJCAI'93, A one-day workshop, Looking at People: Recognition and Interpretation of Human Action.*

[Bröckl et al., 1992] Bröckl, U., Klingert, A. J., and Schmitt, A. (1992). Towards standardized user interfaces for three-dimensional interaction: The development of three-dimensional widgets. In *Proceedings of the Second International Conference on Computational Graphics and Visualization Techniques.*

[Card et al., 1991] Card, S. K., Mackinlay, J. D., and Robertson, G. G. (1991). A morphological analysis of the design space of input devices. *ACM Transaction on Information Systems*, 9(2):99–122.

[Carlton and Shepard, 1990] Carlton, E. H. and Shepard, R. N. (1990). Psychologically simple motions as geodesic paths. *Journal of Mathematical Psychology*, 34(2):127–228.

[Carroll and Mack, 1985] Carroll, J. M. and Mack, R. L. (1985). Metaphor, computing systems and active learning. *International Journal of Man-Machine Studies*, 22:39–57.

[Chen et al., 1988] Chen, M., Mountford, S. J., and Sellen, A. (1988). A study in interactive 3-D rotation using 2-D control devices. *ACM Transaction on Graphics*, pages 121–129.

[Conner et al., 1992] Conner, D. B., Snibbe, S. S., Herndon, K. P., Robbins, D. C., Zeleznik, R. C., and van Dam, A. (1992). Three-dimensional widgets. In *Proceedings of the 1992 Symposium on Interactive 3D Graphics, Special Issue of Computer Graphics, Vol. 26*, pages 183–188.

[Evans et al., 1981] Evans, K. B., Tanner, P. P., and Wein, M. (1981). Tablet-based valuators that provide one, two or three degrees of freedom. *Computer Graphics*, 15(3):91–97.

[Foley et al., 1990] Foley, J. D., van Dam, A., Feiner, S. K., and Hughes, J. F. (1990). *Computer Graphics — Principles and Practice (2nd Ed.).* Addison-Wesley.

[Grissom and Perlman, 1993] Grissom, S. B. and Perlman, G. (1993). StEP(3D): A portable discount usability evaluation plan for 3D interaction. technical report OSU-CISRC-2/93-TR7, Department of Computer and Information Science, Ohio State University, Columbus, OH.

[Halasz and Moran, 1982] Halasz, F. and Moran, T. P. (1982). Analogy considered harmful. In *Proceedings Human Factors in Computer Systems Conference*, pages 383–386.

[Hanson, 1992] Hanson, A. J. (1992). The rolling ball. In Kirk, D., editor, *Graphics Gems III*, pages 51–60. Academic Press.

[Houde, 1992] Houde, S. (1992). Iterative design of an interface for easy 3-d direct manipulation. In Bauersfeld, P., Bennett, J., and Lynch, G., editors, *Human Factors in Computing Systems, CHI'92 Conference Proceedings: Striking A Balance*, pages 135–142. ACM/SIGCHI, ACM Press.

[Hübner, 1990] Hübner, W. (1990). *Entwurf Graphischer Benutzerschnittstellen. Ein objektorientiertes Interaktionsmodell zur Spezifikation graphischer Dialoge.* Springer Verlag.

[Hultquist, 1990] Hultquist, J. (1990). A virtual trackball. In Glassner, A. S., editor, *Graphics Gems*, pages 462–463. Academic Press.

[Johnson, 1963] Johnson, T. E. (1963). Sketchpad III: A computer program for drawing in three dimension. In *Proceedings AFIPS Spring Joint Computer Conference*, volume Vol. 23, pages 347–353, Detroit, Michigan.

[Lakoff and Johnson, 1980] Lakoff, G. and Johnson, M. (1980). *Metaphors We Live By.* The University of Chicago Press.

[MacKenzie, 1992] MacKenzie, I. S. (1992). Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 7(1):91–139.

[MacKenzie and Buxton, 1992] MacKenzie, I. S. and Buxton, W. (1992). Extending Fitts' law to two-dimensional tasks. In Bauersfeld, P., Bennett, J., and Lynch, G., editors, *Human Factors in Computing Systems, CHI'92 Conference Proceedings: Striking A Balance*, pages 219–226. ACM/SIGCHI, ACM Press.

[Mackinlay et al., 1990a] Mackinlay, J., Card, S. K., and Robertson, G. G. (1990a). A semantic analysis of the design space of input devices. *Human Computer Interaction (Hillsdale), Lawrence Erlbaum*, 5(2-3):145–190.

[Mackinlay et al., 1990b] Mackinlay, J. D., Card, S. K., and Robertson, G. G. (1990b). Rapid controlled movement through a virtual 3D workspace. *Computer Graphics*, 24(4):171–176.

[Nielson and Olsen, 1986] Nielson, G. M. and Olsen, D. R. (1986). Direct manipulation techniques for 3D objects using 2D locator devices. In *Proceedings of the 1986 Workshop on Interactive 3D Graphics, Chapel Hill*, pages 175–182.

[Olsen, 1992] Olsen, Jr., D. R. (1992). *User Interface Management Systems: Models and Algorithms*. Morgan Kaufman Publishers Inc.

[Pique, 1986] Pique, M. E. (1986). Semantics of interactive rotations. In *Proceedings of the 1986 Workshop on Interactive 3D Graphics, Chapel Hill*, pages 259–269.

[Robertson et al., 1993] Robertson, G. G., Card, S. K., and Mackinlay, J. D. (1993). Information visualization using 3D interactive animation. *Communications of the ACM*, 36(4):57–71.

[Rubine, 1991] Rubine, D. (1991). Specifying gestures by example. *Computer Graphics*, 25(4):329–337.

[Shepard and Cooper, 1986] Shepard, R. N. and Cooper, L. A. (1986). *Mental Images and Their Transformations*. MIT Press, Cambridge, Massachusetts.

[Sherr, 1988] Sherr, S., editor (1988). *Input Devices*. Computer Graphics — Technology and Applications. Academic Press.

[Shoemake, 1991] Shoemake, K. (1991). Quaternions and 4×4 matrices. In Arvo, J., editor, *Graphics Gems II*, pages 351–354. Academic Press.

[Shoemake, 1992] Shoemake, K. (1992). Arcball: A user interface for specifying three-dimensional orientation using a mouse. In *Proc. of Graphics Interface '92*, pages 151–156.

[Snibbe et al., 1992] Snibbe, S. S., Herndon, K. P., Robbins, D. C., Conner, D. B., and van Dam, A. (1992). Using deformations to explore 3D widget design. *Computer Graphics*, 26(2):351–352.

[Thornton, 1979] Thornton, R. W. (1979). The number wheel: A tablet based valuator for interactive three-dimensional positioning. *Computer Graphics*, 13(2):102–107.

[van Emmerik, 1990] van Emmerik, M. J. G. M. (1990). A direct manipulation technique for specifying 3D object transformations with a 2D input device. *Computer Graphics Forum*, 9:355–361.

[Ware and Osborne, 1990] Ware, C. and Osborne, S. (1990). Exploration and virtual camera control in virtual three dimensional environments. In *Proceedings of the 1990 Symposium on Interactive 3D Graphics, Special Issue of Computer Graphics, Vol. 24*, pages 175–183.

[Zeleznik et al., 1993] Zeleznik, R. C., Herndon, K. P., Robbins, D. C., Huang, N., Meyer, T., Parker, N., and Hughes, J. F. (1993). An interactive 3D toolkit for constructing 3D widgets. *Computer Graphics*, 27(4):81–84.