International Journal of Computers for Mathematical Learning (2006) 11:1–55
DOI 10.1007/s10758-006-0001-z

KAMRAN SEDIG and MARK SUMNER

CHARACTERIZING INTERACTION WITH VISUAL MATHEMATICAL REPRESENTATIONS

ABSTRACT. This paper presents a characterization of computer-based interactions by which learners can explore and investigate visual mathematical representations (VMRs). VMRs (e.g., geometric structures, graphs, and diagrams) refer to graphical representations that visually encode properties and relationships of mathematical structures and concepts. Currently, most mathematical tools provide methods by which a learner can interact with these representations. Interaction, in such cases, mediates between the VMR and the thinking, reasoning, and intentions of the learner, and is often intended to support the cognitive tasks that the learner may want to perform on or with the representation. This paper brings together a diverse set of interaction techniques and categorizes and describes them according to their common characteristics, goals, intended benefits, and features. In this way, this paper aims to provide a preliminary framework to help designers of mathematical cognitive tools in their selection and analysis of different interaction techniques as well as to foster the design of more innovative interactive mathematical tools. An effort is made to demonstrate how the different interaction techniques developed in the context of other disciplines (e.g., information visualization) can support a diverse set of mathematical tasks and activities involving VMRs.

KEY WORDS: computer-supported mathematical reasoning, human–computer interfaces, interaction techniques and design, interactive visual representations, mathematical cognitive tools, visual mathematical representations

1. INTRODUCTION AND RATIONALE

This paper presents a categorization and characterization of computer-based interactions by which learners can explore and investigate visual mathematical representations (VMRs). In order to do this, this paper draws together research in human–computer interaction design, information visualization, visual reasoning, and cognitive technologies to bear on exploring VMRs. This paper has a threefold purpose. First, it aims to demonstrate how the different interaction techniques developed in the context of other disciplines can support a diverse set of mathematical tasks and activities involving VMRs. Second, it intends to provide a framework to help designers of mathematical tools select and analyze these techniques and to foster the design of more innovative interactive mathematical tools. Third, it hopes to engender further interest among mathematics education researchers and researchers in the above-mentioned disciplines to collaborate with each other and to conduct more systematic research in this area. This paper is a preliminary step in the development of a more systematic framework to provide guidelines as to which interaction techniques are most appropriate for given mathematical tasks. As such, this is *not* intended to be a prescriptive paper. Additionally, the paper does not address pedagogical issues related to conditions and environments in which computer-based mathematical tools can or should be used. These are long-term research problems and are beyond the scope of this article to address.

Visual representations may be defined as a collection of graphical symbols that visually encode causal, functional, structural, and semantic properties and relationships of a represented world – either abstract or concrete (Glasgow, Narayanan and Chandrasekran, 1995; Peterson, 1996; Card, MacKinlay and Shneiderman, 1999; Cheng, 2002). VMRs are those representations that encode these properties and relationships for a represented world consisting of mathematical structures or concepts (Cuoco and Curcio, 2001; Hitt, 2002). VMRs are used extensively in mathematics education. Some examples of VMRs used in different areas of mathematics include graphs (topological, functional, and statistical), visual patterns, geometric structures, diagrams, and representations of abstract concepts such as fractions, operations, and counting. Visual representations are used in every discipline and at different levels of complexity (Barwise and Etchemendy, 1998). Although their benefits to reasoning, problem solving, and learning are not fully understood yet (Scaife and Rogers, 1996; Otero, Rogers and du Boulay, 2001), the fact that visual representations bring value to these processes is shared by many researchers (Paivio, 1983; Larkin and Simon, 1987; Pettersson, 1989, 1996; Jonassen, Beissner and Yacci, 1993; Glasgow et al., 1995; Peterson, 1996; Zhang, 1997; Hansen, 1999; NCTM, 2000; Cuoco and Curcio, 2001; Cheng, 2002). Various research suggests that using static graphical representations can support cognitive activities – such as organizing, comparing, modeling, visualizing, and interpreting – involved in investigation and exploration of mathematical objects, concepts, and problems (Botsmanova, 1972; Keller and Keller, 1993; Pimm, 1995; Holzl, 1996; English, 1997; Zhang, 1997; Barwise and Etchemendy, 1998; Matsuda and Toshio, 1998; Arcavi and Hadas, 2000; NCTM, 2000; Bridger and Bridger, 2001; Diezmann and English, 2001; Healy and Hoyles, 2001; Yerushalmy and Shternberg, 2001).

During thinking and reasoning about or with information, people (users, learners, or subjects) often need to analyze the information, evaluate it, and elaborate it (Fisher, 1990; Glasgow et al., 1995; Ormrod, 1995; Peterson, 1996; Davidson and Sternberg, 2003). This will make it easier for them to deconstruct, decode, and understand its layers of meaning, make sense of it, and/or apply it. These broad, general cognitive tasks are dynamic and take place by selectively and constructively interacting with and expanding on the information, based on previous knowledge. Examples of more specific and low-level cognitive tasks people perform on information are: rearrange and reorganize it, add to it, look at it from different perspectives, focus on and look for specific elements within it, modify it, magnify it, hide or eliminate parts of it, decompose it, identify and locate parts of it, create links within it, and probe its parts. Learners perform these tasks with VMRs as well and may need mental support. When thinking and reasoning about or with a static VMR, if a learner[1] has never encountered the VMR before or has no prior knowledge of it, how can the learner interpret and investigate the VMR? For instance, someone who has no prior knowledge of various 2D representations of a cube may interpret a wire-frame, 2D representation of a cube as 8 independent lines or 3 parallelograms. In this case, being able to view the VMR from different perspectives (e.g., by rotating it) may, for instance, assist the learner in interpreting the representation.

With the advance of computing, we now have powerful computer-based cognitive tools[2] that encode and display information visually and allow for varied forms of interaction with the displayed information in order to perform different perceptual and cognitive tasks (Norman, 1993; Card et al., 1999; Lajoie, 2000; Beynon et al., 2001). Mathematical cognitive tools can be used to encode and display VMRs and allow learners to interact with and investigate the VMRs. In this way, interaction can be viewed as a mediator between a VMR and its user, intended to support the cognitive tasks that can be performed on or with the static, displayed representation (Tall and West, 1992; Kaput, 1993; Card et al., 1999; Stojanov and Stojanoskl, 2001; Sedig, Rowhani, Morey and Liang, 2003). Interaction can extend and enhance the expressiveness of VMRs by allowing their inherent syntax and semantics to be explored in a dynamic fashion. Many researchers from different disciplines have suggested that interactive computational tools can enhance the communicative power of visual representations by allowing different features, elements, and layers of the static information to be made explicit and available when needed (Tall and West, 1992; Hanson, Munzner and Francis, 1994; Pimm, 1995; West, 1995; Holzl, 1996; Tweedie et al., 1996; Dix and Ellis, 1998; Matsuda and Toshio, 1998; Strothotte, 1998; Arcavi and Hadas, 2000; Olive, 2000; Gonzalez-Lopez, 2001; Morey and Sedig, 2004; Moyer et al., 2001; Otero et al., 2001; Sedig et al., 2001, 2003; Spence, 2001; Straesser, 2001; Kordaki and Potari, 2002; Stylianou, 2002). A few of the general benefits these researchers have suggested include: interaction can mediate dialectic, formative, and emergent reasoning and understanding by providing opportunities for experimentation (e.g., by redoing, reformatting, rearranging, and adapting the information); interaction can support exploratory, discovery-based investigation of ideas by facilitating the qualitative and intuitive understanding of ideas before grasping formal concepts; interaction can guide and transform the path of reasoning and understanding; interaction can help an evolutionary, iterative process of sense making by allowing construction and reconstruction of the visual information; interaction, by distributing the information processing load of the visual information, can allow users to think in partnership with the information; and, finally, interaction can serve as the coordinator between the internal mental model of learners and the external visual representation.

So far we have discussed interaction in a general sense. But, what do we mean by interaction in this paper? Computer-based interaction and interactivity have many connotations and meanings in different contexts (Norman and Draper, 1986; Laurillard, 1993; Shedroff, 1999; Sims, 1999, 2000; Yacci, 2000; Otero et al., 2001; Preece, Rogers and Sharp, 2002). Interaction can operate at two levels: macro-level and micro-level. Macro-level interaction is concerned with the high-level, pedagogical design choices, such as constructivism, situated learning, instructivism, cognitive apprenticeship, and so on (e.g., see Boyle, 1997). Micro-level interaction is concerned with the lower-level, cognitive-task design possibilities, such as zooming, searching, rearranging, correlating, and so on (e.g., see Card et al., 1999). Different micro-level interactions can be combined and embedded in macro-level interaction designs. As such, because micro-level interactions are intended to support lower-level cognitive tasks, they can be used in a wide range of environments, such as simulations, microworlds, and educational games. This paper categorizes and characterizes micro-level interactions. Therefore, as important as macro-level interactions are, it is outside the scope and length of this article to discuss pedagogic-based interactions.

In the context of this paper, interaction refers to a user, learner, and/or student communicating with one or more VMRs via a human–computer interface. Interaction with a VMR has at least two implications: the learner acting upon a VMR, and the VMR responding or reacting in some form for the learner to interpret. An example of this type of interaction is when a learner clicks on a diagram (action) and the diagram is animated (reaction).

There are many micro-level interaction techniques. Since interaction design is a young discipline, most of the early research has been devoted to building and evaluating domain-specific tools, ranging from productivity and edutainment tools to cognitive and educational tools. This process of construction of tools has lead to the creation of dozens of interaction design techniques and methods. Many of these techniques can be used to interact with VMRs. Unfortunately, research in this area is scattered across and reported in several disciplines, such as human–computer interaction, visualization technologies, and cognitive technologies. Moreover, often, the characterizations and descriptions of these techniques either are bound to the domain for which they were developed or are too general and inapprehensible, making it difficult to apply them to the design of interaction for VMR-based mathematical software (e.g., see Parker, Franck and Ware, 1997; Card et al., 1999). Finally, no attempt has been made to organize and categorize these techniques according to their common features so as to reduce the space of possibilities available to designers of cognitive tools. In order to design interactive VMRs, designers of mathematical cognitive tools must know the various types of interaction that are available, have a clear understanding of the characteristics of each type of interaction, be aware of the various ways a particular type of interaction may be used, and be familiar with some examples of how these techniques can be used to interact with VMRs.

This paper characterizes a large number of micro-level interaction techniques that can be used to support VMR-based mathematical tasks; it organizes and categorizes these techniques according to their common characteristics and features; and, it provides a diverse set of examples to suggest how these interactions can be used to accomplish different tasks. Because of shortage of empirical research and lack of a prescriptive framework in this area, the paper does not provide fast rules or guidelines as to which interactions are the most appropriate for given tasks and situations.

The rest of this paper is organized as follows. In Section 2, we present a few preliminary concepts that assist with later discussions. Section 3 forms the bulk of this article, where we categorize and characterize the different interaction techniques. Numerous and varied mathematical examples and tools are used to clarify the concepts which are presented and to demonstrate the applicability of the different methods and ideas discussed. Section 4 presents a summary of the article and makes suggestions regarding directions for future research in this area.

2. PRELIMINARY CONCEPTS

Before presenting a categorization and characterization of different interaction techniques, this section discusses three concepts that may assist in understanding what follows. The three concepts are affordance, flow, and focus.

2.1. *Affordance*

In human–computer interaction, affordance refers to the way an on-screen object (in this case a VMR) advertises its usage cues – i.e., what sorts of operations can be performed on it (Preece et al., 2002). There are two types of affordances: real and perceived. Real affordance refers to what interactions are possible with a VMR. Perceived affordance refers to what interactions the learner perceives to be possible. The goal of interaction design is for interactive VMRs to communicate their affordances so clearly that the learner can easily perceive their real affordances. For instance, if a control point on a VMR is interactive, it can advertise its affordance by flashing or by a change of color or cursor as a result of a mouse-over action (see Figure 1 where the cursor indicates that the vertex is interactive).

2.2. *Flow*

Flow of interaction refers to the effect of the interaction on how the learner perceives the relationship between cause (action) and effect (reaction or response) in the time–space continuum (Sedig and

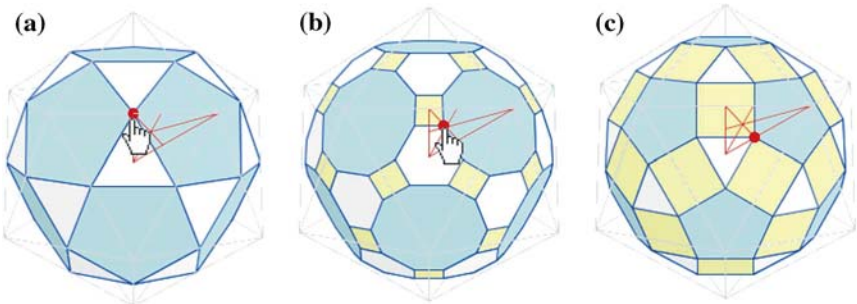

Figure 1. Continuous metamorphic flow from icosidodecahedron (a) to rhombicosidodecahedron (c).

Morey, 2005). There are two types of interaction flow: continuous and discrete. Continuous flow is when cause and effect are observed simultaneously. In continuous flow, a VMR fluidly responds to interaction with it. For instance, Figure 1 is a screen capture of a mathematical cognitive tool, Archimedean Kaleidoscope (Morey and Sedig, 2003), that allows users to investigate how Platonic and Archimedean solids are related by interacting with their visual representations. The figure shows an icosidodecahedron (a) being continuously morphed into a rhombicosidodecahedron (d). The learner can click on the interactive vertex control (black dot) on the solid to change its shape to other Platonic and Archimedean solids. Interaction with this VMR is continuous because the movement of the mouse cursor is fluidly and dynamically translated into a change in the shape of the solid, visualizing the intermediary snapshots or stages of change. Continuous interaction can allow learners to control the flow and communication of information, as it is possible to freeze intermediary images in time by holding the mouse click down while thinking and reasoning about the image. Discrete interaction takes place when user's action and VMR's reaction are separated in time. For instance, in Figure 1, rather than continuously morphing the icosidodecahedron, it can be instantly transformed into the rhombicosidodecahedron by clicking a button, in which case no intermediary snapshots are shown. This latter interaction flow is discrete because transitions take place in temporally distorted snapshots. To compensate for this temporal distortion, learners may need to resort to undoing and reversal of actions.

2.3. *Focus*

Focus refers to the center of attention of the learner while interacting with a VMR (Sedig and Morey, 2005). Since controls that allow interaction with aspects of interest of a VMR can either be inside or outside it, a learner can interact with a VMR directly or indirectly. In direct interaction, the learner is directly focused on a VMR and interacts with it without any other intermediary representations. In indirect interaction, the learner interacts with a VMR through another representation. For instance, a mathematical cognitive tool, K-Lattice Machine (Sedig et al., 2005), allows both direct and indirect interaction with K lattices, where K lattices are 2-dimensional infinite structures that have the property that each vertex has a K representation. Figure 2 shows a screen capture of the K-Lattice Machine. Two VMRs are seen in the figure: a state-transition diagram[3] (a), and a geometric lattice structure (b). These two VMRs are informationally equivalent, but provide for different ways of investigating the lattice (ibid.). If the focus of the learner is on the transition diagram and exploring its Hamiltonian paths, then the learner can click on different transition paths to observe how the diagrammatic representation can generate the geometric lattice. This means that the learner interacts with the transition diagram directly and with the lattice indirectly. This indirect interaction with geometric K-lattices can allow users to investigate the relationship between the two different forms of representation of lattices (ibid.). If the focus of attention is on the geometric

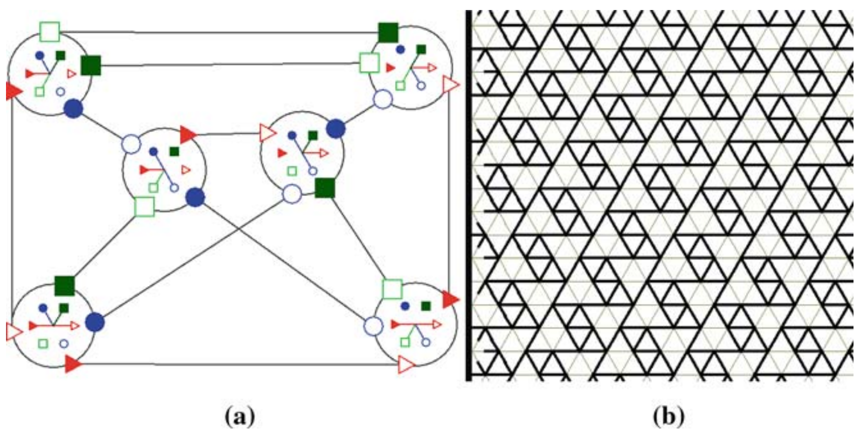

Figure 2. Direct interaction with the transition diagram (a), resulting in indirect interaction with the lattice (b).

representation of the lattice, the learner can interact with the lattice itself to construct it.

3. CATEGORIZATION AND CHARACTERIZATION OF INTERACTION TECHNIQUES

We have organized micro-level interactions with VMRs into two main categories: basic interactions and task-based interactions. Basic interactions are based on three fundamental, root metaphors derived from the way in which humans use their bodies to interact with the external world. These three root metaphors are conversing, manipulating, and navigating. Conversational interaction is based on the use of the mouth and talking; manipulational interaction is based on the use of the hands and handling; and, navigational interaction is based on the use of the feet and walking (Sedig and Morey, 2005; Sherman and Craig, 2003). Just as people use these means to explore and make sense of the objects in the physical world, the same techniques can be used to make sense of VMRs and reason about them in mathematical cognitive tools. These three interactions are foundational and elemental in that other forms of interaction are based on them. Task-based interactions are based on low-level cognitive tasks learners perform to explore VMRs, as discussed in Section 1. These are 12 in number, and they include: animating, annotating, chunking, composing, cutting, filtering, fragmenting, probing, rearranging, repicturing, scoping, and searching. The 12 task-based interactions presented in this section are based on and describe the available techniques found in the research literature.

In the case of both categories, often an interaction represents a number of techniques that have common features, goals, intended benefits, and/or characteristics. We characterize these techniques as variations of one another and categorize them under one interaction. For instance, in the literature one may find techniques such as distorting, scaling, and stretching. This paper categorizes and characterizes these techniques under one task-based interaction, called repicturing. This is because the main task or goal is to allow an alternative viewing of a VMR, but using different techniques. This approach, we believe, reduces the space of possible interactions and makes it easier to design and analyze interactive VMRs. Additionally, this approach can facilitate the categorization of newly developed techniques under the existing categories and interactions.

Finally, the task-based interactions are built on top of the basic interactions. One or more basic interactions can be used to create a task-based interaction. For instance, one such task-based interaction is composing a VMR. As will be seen later, this can be achieved by using conversation, manipulation, or both. Combining the task-based interaction techniques themselves can allow learners to perform more integrated, complex tasks.

3.1. *Basic Interactions*

3.1.1. *Conversing*

The archetypal human method of interaction is conversation. Conversing with a VMR refers to expressing actions and intentions through a conversational language – i.e., symbolic or lexical commands issued as input to the VMR. When conversing with a VMR, learners can issue an input action to a VMR to give it instructions, to query it, to transform it, and so on. Input techniques such as form fill-ins, text-based menus, natural language dialogs, pen-based gestures, macros, procedure-based programming languages, and command-based text can all be regarded as conversational interaction. Generally, conversational interaction requires the learner to have some prior understanding of the VMR and knowledge of the symbolic commands understood by the system. That is, the learner must know and be able to properly use the symbolic language that the tool recognizes. The need for prior understanding of the conversational language can sometimes make it difficult for learners to express their intentions (Hutchins, Hollan and Norman, 1986). Additionally, conversational interaction is not very effective in advertising the interaction affordances of representations.

The flow of conversational interaction is discrete, in the sense that a command has to be formulated and submitted to the system before the learner receives any feedback from the system. As such, some researchers have suggested that command-based interaction demands more cognitive, conscious effort on the part of the user (Holst, 1996). The focus of this type of interaction is indirect since communication with VMRs takes place via the intermediary of a symbolic language (Sedig et al., 2001). Conversational interaction, in the form of macros and procedures, is often more effective when learners need to express repetitive and recursive activities, which may not be as easy to express using the other basic interactions, that is, manipulation or navigation.

3.1.2. *Manipulating*

Manipulating a VMR refers to pointing to the VMR using a screen cursor to touch, handle or grasp one of its elements or the whole representation to perform some sort of action upon it (Hutchins et al., 1986; Shneiderman, 1988; Sedig et al., 2001; also see Figure 1). Manipulation usually supports actions such as selecting, dragging, moving, modifying, or morphing a VMR. Manipulation is intended to be a tangible form of interaction. With manipulation, learners can "reach" their "hand" into VMRs to handle them. As such, interactive VMRs may be viewed as virtual manipulatives (Moyer, Bolyard and Spikell, 2001).

There are many different ways one can manipulate a VMR, and with many varying goals. Some dynamic geometry tools, for instance, incorporate dragging and dropping as a form of manipulation to help users explore and understand various concepts in geometry (Holzl, 1996). Effectiveness and ease of manipulating VMRs depends on how well their affordances are advertised. Generally speaking, feedback from the environment, either in the form of a change in the shape of the cursor or a highlighting of an element of the VMR, advertises the interaction affordances of the VMR, allowing the learner to know what is manipulable and what action the manipulation will cause.

In its original conception as an interaction technique, in contrast to conversation-based command interaction, manipulation was intended to reduce users' cognitive load by allowing them to engage with the on-screen representations directly (Hutchins et al., 1986). This type of interaction is called Direct Manipulation and aims to have the following features: continuous representation of the objects of interest; rapid, reversible, incremental actions with immediate feedback; and physical actions and button pressing instead of complex command language syntax (Shneiderman, 1988; Preece et al., 2002). This interaction method is widely used in all kinds of software including mathematical cognitive tools.

As direct manipulation became a widespread interaction style, some researchers questioned its effectiveness in the context of learning activities, as learners, due to ease of interaction, would not engage with the learning material consciously; furthermore, this interaction style involved little reflective thought compared to command-based linguistic interactions (Svendsen, 1991; Golightly, 1996; Holst, 1996). Problems attributed to direct manipulation include directness of interaction and immediateness of feedback. For instance, in solving the 8-puzzle,[4] changing the focus of interaction from direct to indirect (i.e., moving puzzle pieces using adjacent buttons rather than pointing at the pieces and moving them directly) resulted in learners paying more attention to each move and using a "look-ahead" strategy. The group that used the direct manipulation version of the activity used strategies that involved less planning and were based mainly on "trial-and-error", since recovering from errors was easy (Golightly, 1996).

Using broad characterizations of interactions, such as manipulation, and assessing their effectiveness without considering other interactivity issues, however, may not provide accurate ways of thinking about these interactions. To investigate that manipulation is not an educationally ineffective interaction method, using interactive 2D transformation geometry VMRs, Sedig et al. (2001) have argued and empirically demonstrated that the problem with direct manipulation lies in "what is manipulated" rather than with manipulation itself or the directness of interaction. They have suggested that, when considering manipulation, an important factor in determining its efficacy is the focus of interaction (Sedig et al., 2001). Such considerations result in a less general, and more refined, characterization of manipulation as a technique for interacting with VMRs.

3.1.3. *Navigating*

Navigating a VMR refers to moving on, over, or through the representation. In discussions of virtual environments, navigation involves two activities: traveling and wayfinding (Chen, 1999; Ware, 2000; Sherman and Craig, 2003). Traveling refers to how one is to move on or through a space. Wayfinding refers to methods of determining where one is located and finding a path to get to other locations. Navigation is an exploratory process that rarely modifies the VMR itself. Spence (1999) describes navigation as being composed of browsing, mental modeling, and interpreting. Browsing takes advantage of the learner's visual perception to analyze a VMR; mental modeling involves developing an internal mental organization of the VMR; and, interpreting is making sense of one's current mental model. Navigation can be used to explore both structural and conceptual VMRs. For instance, learners can step through 3D VMRs to explore their interior elements and connections. They can also walk upon flat, 2D VMRs, exploring surface features and relationships (Sedig et al., 2003).

Some techniques used in navigating VMRs related to traveling and wayfinding include walking on or through a VMR and flying over, around, or through a VMR. Walking can be continuous or discrete. For instance, Hanson and Ma (1995) have designed a system that allows continuous walks on manifolds along a local geodisc path. Element-to-element walks are the more prevalent form of navigation. This can usually be performed with VMRs in which the VMR is partitioned according to distinct elements. One form of element-to-element movement is an attribute walk (Card et al., 1999), in which a learner can select an element within the VMR and search for the next element with the same attributes. An example of this type of walk is presented in the section on searching. Flying can allow the learner to get a "bird's eye view" of the VMR. It is especially useful for 3D structures. Raskin (2000) describes flying as a fly-and-dive process, where one increases altitude to gain an overview over the environment and dives down to view specific details. This process allows learners to familiarize themselves with both the context and the specifics of the VMR.

Finally, navigation can be used to help learners develop an internal spatial cognitive map of a VMR (Golledge, 1999; Spence, 1999, 2001; NCTM, 2000; Sedig et al., 2003). Cognitive mapping can be divided into three knowledge development stages: (1) landmark knowledge, (2) route knowledge, and (3) survey knowledge. Landmark or location knowledge is the knowledge a learner has about visual details of specific locations on or in a VMR. Route or path knowledge is the knowledge a learner has about the sequence of navigational actions required to follow a route. Survey knowledge is the knowledge a learner has about the spatial layout, structure and relations among locations and routes in the environment. A learner obtains survey knowledge directly from maps or indirectly by integrating the knowledge of several routes into a network of routes – a cognitive map. Echoing the same concept, the National Council of the Teachers of Mathematics (NCTM, 2000) calls for students to be engaged in four types of mathematical investigation of spatial representations: direction (which way?), distance (how far?), location (where?), and representation (what objects?). In investigating these questions, learners notice landmarks, then build knowledge of a route (a connected series of landmarks), and finally put many routes and locations into a kind of overall cognitive map (ibid.).

3.2. *Task-based Interactions*

3.2.1. *Animating*

Animating refers to interacting with a VMR to generate movement within it (Jones and Scaife, 2000). Animation can occur unidimensionally or multidimensionally, depending on the complexity of the VMR. For instance, in multidimensional representations, several movements can occur simultaneously or on multiple planes. Benefits of animating VMRs include: attracting and directing attention to embedded detail, visualizing dynamic and transitional processes, supporting external cognition, increasing visual explicitness of encoded information, and facilitating perception of semantic and temporal transformations inherent in the VMR (Thompson and Riding, 1990; Park, 1998; Jones and Scaife, 2000; Morrison et al., 2000; Sedig et al., 2003).

An example of animating VMRs is in a system called PARSE, Platonic Archimedean Solids Explorer (Sedig et al., 2003). PARSE provides learners with solid transition maps depicting transitional relationships among a set of solids (Figure 3). Learners view animated metamorphoses of the solids along the transitional paths by clicking on interactive arrows to take element-to-element walks on the map. This type of animation can bridge the visual gap between the two solids and enhance sense making of the static representation of the solids by communicating their semantically inherent temporal transformations (Sedig et al., 2003; Thomas and Demczuk, 2002). Some

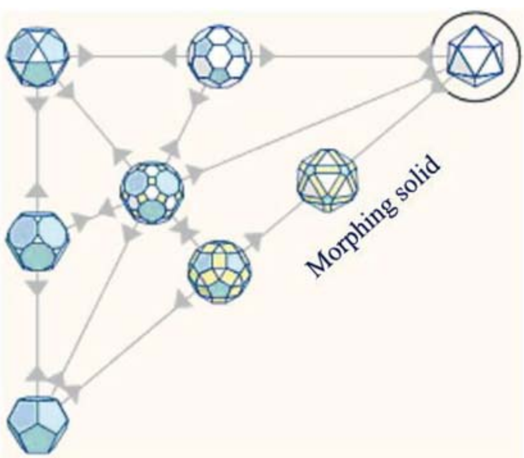

Figure 3. Animated walk along paths of a solid transition map.

other examples of animating VMRs include demonstrating the slicing of 4-dimensional structures (Hanson et al., 1994), automatic rotation and metamorphosis of 3-dimensional shapes (Morey and Sedig, 2003), and rocking and tumbling VMRs (Morey and Sedig, 2004).

One of the problems with animating VMRs is that animation may increase their visual explicitness, resulting in the reduction of mental effort and reasoning on the part of the learner due to overconfidence in the amount of knowledge obtained from the animation, and hence shallowness of learning (Jones and Scaife, 2000; see also Salomon, 1979). Another potential pitfall of animation is temporal transience. Because of temporal movement, the animated information has transience, and learners may be unable to parse it or reaccess pieces of it since it is not always perceptually available (Jones and Scaife, 2000). To deal with temporal transience, while animating transition from one geometric solid to another, PARSE breaks down animation steps into small multiples (e.g., Figure 4). Small multiples are a series of images with the same design structure placed parallel to one another to help comparative reasoning (Tufte, 1997). This automatic breaking down of the animation steps makes the steps available for further investigation and reflection. A study of PARSE showed that small multiples enhanced this interaction technique and helped learners with their reasoning and exploration of how geometric solids can be obtained from one another (Sedig et al., 2003).

3.2.2. *Annotating*

Annotating refers to interacting with a VMR to augment it by placing notes or marks on or around the VMR. These notes can be added to the VMR using various forms and methods such as highlighting, crossing off, labeling, coloring, breadcrumbs, recording, and sticky-notes. Annotation can allow learners to add personal metadata to a

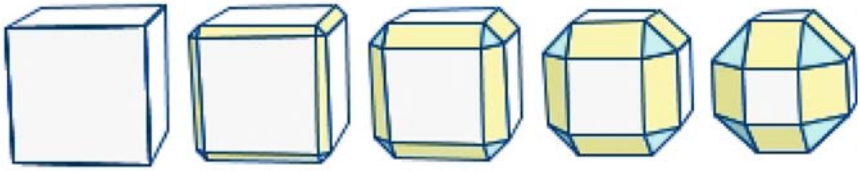

Figure 4. Small multiples showing transition from a cube to a rhombicuboctohedron.

VMR. Annotations can always stay open and visible or can be shrunk and iconified and be opened on demand.

Annotation can be accomplished in two ways: manually and automatically. Manual annotation gives the learner control over when and where to add annotations to a VMR. Figure 5 shows the graph of a rectangular hyperbolic function ($xy = 1$) that has been annotated to highlight important properties of the function. Adding these notes to the VMR can facilitate identification of its characteristics. Annotations can make apparent features that would not be immediately apparent otherwise. Learners may add formulas, calculations, techniques, or other comments to which they can refer when encountering similar functions.

Another example of manual annotation of VMRs is shown in Figure 6, a screen capture of a prototype system, LatticeSpace, designed in our Cognitive Engineering research laboratory. In this example, while investigating the structure of a 3D lattice, users can annotate the lattice by adding black markings to it to trace their path. These markings are called breadcrumbs. Breadcrumbs are like electronic footprints to mark one's path in a space being navigated. They can support memory tracing and reasoning, and hence wayfinding in mathematical structures – i.e., what parts of the VMR have been visited and where the current location is.

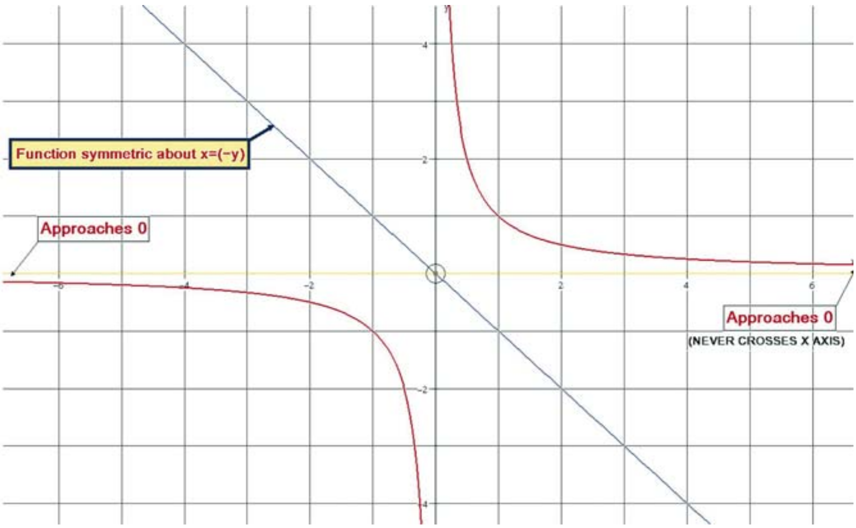

Figure 5. A rectangular hyperbolic function, manually annotated to highlight important features.

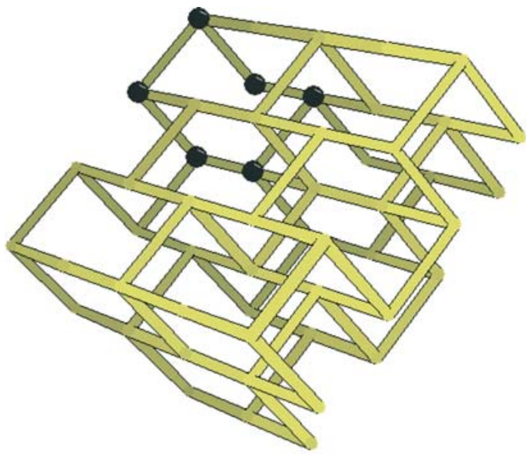

Figure 6. Annotating a 3D lattice using breadcrumbs.

In automatic annotation, the system leaves traces on a VMR in response to particular learner actions. Figure 7 demonstrates a prototype idea of how automatic annotation can facilitate traveling a graph. The figure shows a simple directed graph with nodes and edges annotated as an automatic recording of a learner's interaction with the graph. As the learner clicks each node or edge, walking the graph, the tool can leave breadcrumbs (dots on nodes and edges) highlighting which and how many times elements of the graph have been visited. Numbers are automatically added along edges to show the

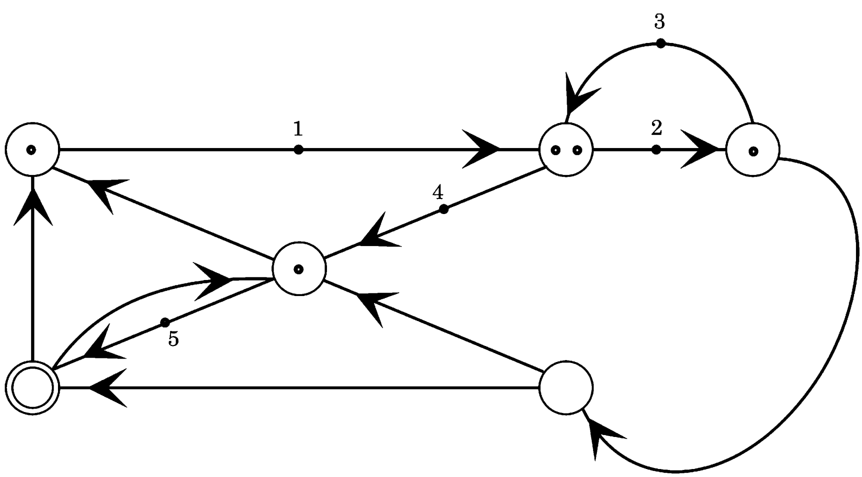

Figure 7. A directed graph, automatically annotated to trace interaction.

order of the traversal of the edges. Additional annotation is provided to convey which node is currently being visited by outlining the current node. Thus, in Figure 7, the learner may wish to determine whether or not the graph possesses an Eulerian or Hamiltonian path (Sedig et al., 2005).

Annotations can be permanent or temporary. In permanent annotation, added marks or text become permanently embedded into the VMR. Temporary annotation permits learners to remove the added annotations. Temporary annotation is needed when the learner may no longer want to have the VMR annotated, as for instance when the VMR has become visually noisy as a result of too many annotative marks. Figure 8 shows an example of temporary annotation facilitated by the Geometer's Sketchpad (Jackiw, 1995). In this example, as a learner drags the endpoint *a* around a circle, the motion path of the midpoint of line *ab* is automatically annotated. Learners select points they wish to leave traces. This is a temporary annotation since the trace is removed when another action is performed. This type of annotation allows learners to see possible locations of the midpoint of *ab* for positions of *a* around the circumference of the circle.

Some applications and benefits that may result from annotating VMRs include: annotation can help learners explain a VMR, ask questions about its content, and review some of its aspects (Sherman and Craig, 2003); annotation can help promote understanding and reflection by creating external marks on a VMR with which one can reason (Peper and Mayer, 1986); it can reduce mental activities by supporting a learner's goals, plans, perception, memory and reasoning (Preece et al., 2002); it can facilitate recall and reflection on past action, thereby supporting search for solutions to problems (Pimm,

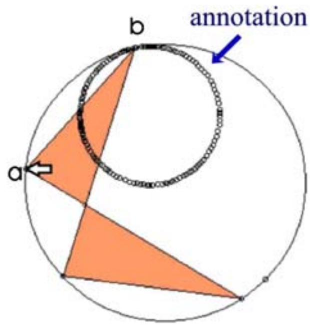

Figure 8. Automatic annotation tracing motion path of midpoint of *ab*.

1995; Preece et al., 2002; Sedig et al., 2002); and, it can allow learners to organize and make links between various objects (Shneiderman and Kang, 2000).

Finally, in virtual environments, annotation is used in ways that may be applicable to VMRs (Sherman and Craig, 2003). Some ways of annotating VMRs include (1) elemental, where a note is attached to an element within a VMR; (2) temporal, where a comment is attached to a VMR during a process of change; and (3) causal, where an explanation is attached to an interactive element of a VMR or a VMR that has a causal relationship to other visual elements.

3.2.3. *Chunking*

Chunking refers to interacting with a VMR to group, unite, or cluster a number of similar or related, but disjointed, visual elements into one visual structure. One may then think of and interact with this new structure as a single element. In cognitive psychology, chunking is referred to as the mental process of collecting and grouping elements that have strong association with one another (Gobet, Lane, Croker, Cheng, Jones, Oliver and Pine, 2001). Chunking supports and facilitates cognitive processes involved in encoding, extracting, remembering, and understanding information (Winn, 1993; Gobet et al., 2001). Dealing with chunks rather than atomic elements can alleviate the learner's short and long-term memory by reducing the amount of information that must be stored since chunks are conceptually treated as single elements (Fleming, 1993). Chunking can particularly be useful in cases when learners need to think of and mentally manipulate a group of elements as one entity. This not only makes the mental and interactive task more efficient, it can also allow learners to build more complex knowledge from the basic elements of the VMR.

Chunking can be performed either automatically or manually. Automatic chunking is often performed and predefined by the system. For example, Figure 9 shows a hypothetical case where a number of blocks are arranged randomly (a). A child may select all the blocks by dragging a window around them (a) and clicking on a button to group them. The system can then automatically chunk the blocks into a single, perceptually-easier block with the value 10 (b). This newly formed chunk can then be used to perform more complex operations.

An example of the value of chunking is in building K lattices. Each K-lattice structure can have several K-lattice patterns, the smallest

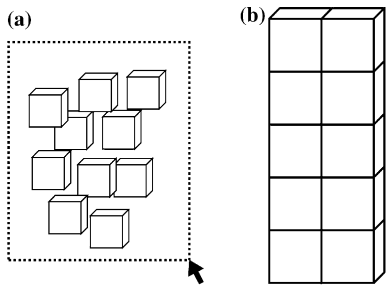

Figure 9. Grouping blocks to form meaningful chunks.

number of chunked Ks that uniquely describes the lattice. Using a K-lattice pattern as a chunked unit, by recursively joining it to itself, the K lattice can be created (Sedig et al., 2005). As a way of hypothesizing and investigating the structure of such lattices, manual chunking can be used to group together Ks into K-lattice pattern chunks. For example, Figure 10c shows a relatively simple K-lattice structure. A learner may click on a few Ks having different orientations (a) to chunk them together into the configurations shown in 10b. The learner can then manipulate these chunks to link them together and build a more complex structure. Any one of the chunks displayed in 10b can then be used as a building block for creating the lattice structure seen in 10c.

A chunking technique used in information visualization applications is elision, where groups or clusters of related visual elements are hidden or collapsed together by being represented as one visual

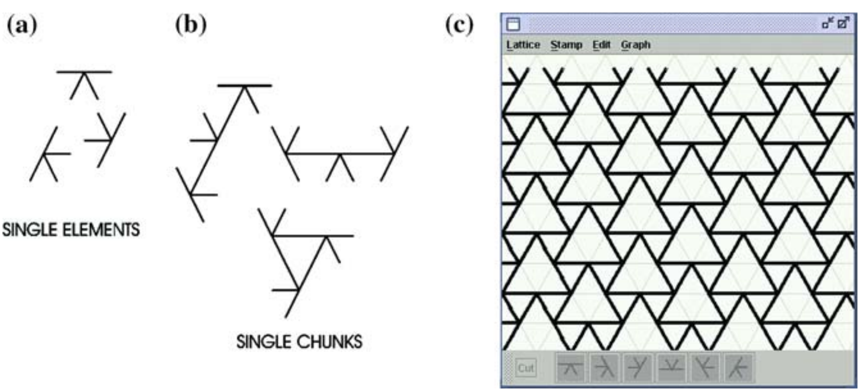

Figure 10. Chunking Ks to facilitate the construction of a repetitive K-lattice structure.

representation to give prominence to other elements. Interacting with the collapsed visual representation causes it to reveal what it represents. For instance, this technique has been applied to large graph visualizations to keep the number of graph elements perceptible (Parker et al., 1997). In the case of other VMRs, elision can also be used to chunk mathematical objects, allowing learners to treat them as belonging to the same class of objects.

3.2.4. *Composing*

Composing – whose variants are assembling, building, joining, linking, connecting, and constructing – refers to an interaction method by which learners put together separate visual elements to create a VMR (e.g., Figure 10c, above). Composing is similar to chunking, but the goal is often to build the whole VMR, rather than its subcomponents. Additionally, the elements need not be strongly associated for them to be connected together. Composing a VMR often allows learners to participate in the creation of the VMR in a generative manner, rather than an observatory manner, as may be the case with animation for instance.

An example of composing is in Geometer's Sketchpad (Jackiw, 1995), where learners can use menu-based conversation to construct geometric diagrams. Figure 11 shows a simple geometric construction containing a triangle inscribed within a circle and a perpendicular bisector for each edge of the triangle. A learner can use context-

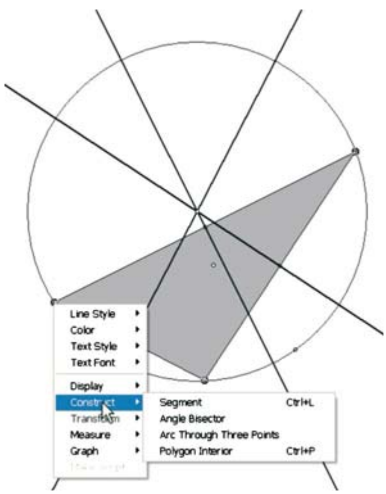

Figure 11. Using conversation in Geometer's Sketchpad to compose a VMR and develop visual geometric proofs (adapted from Hanna, 2000).

sensitive, text-based menus to create the geometric elements of the VMR and assemble them together.

Another well-known example is in the original Logo Turtle Geometry microworld (Papert, 1980), where conversation (procedural code) is used to compose visual designs. Cabri-géomètre also allows for the composition of geometric structures using macro-based conversational interaction, a simpler and easier version of procedures (Holzl, 1996).

Composing can offer learners an opportunity to confirm, explore, and experiment with mathematical concepts (Hazzan and Goldenberg, 1997; Jones, 2000; Marrades and Gutierrez, 2000; Healy and Hoyles 2001). Composing has been used as an interaction to promote deductive reasoning and provide means to work out geometric proofs (Hanna, 2000; Scher and Goldenberg, 2001; Leung and Lopez-Real, 2002). Learners can use conversation or manipulation, provided in dynamic geometry environments (DGEs), to compose and interact with VMRs to aid in performing such proofs. For instance, Figure 11 shows how the perpendicular edge bisectors of a triangle intersect at a single point (Hanna, 2000). Learners begin by constructing a circle. Within the circle, they inscribe a triangle with vertices fixed along the circumference. A perpendicular bisector is created for each side of the triangle. As any vertex of the triangle is manipulated and dragged around the circumference, the bisectors continuously intersect at a single point. The interactive nature of these environments can lead learners to discover novel proofs, outside the realm of traditional Euclidean geometry.

3.2.5. *Cutting*

Cutting – whose variants are cropping, slicing, truncating, pruning, and clipping – refers to interacting with a VMR to remove unwanted or unnecessary portions of the VMR and thus control the exposed regions of it (Banks, 1992; Roth, Chuah, Kerpedjiev, Kolojejchick and Lucas, 1997). Learners may select particular elements or regions to be removed. Conversely, learners may select elements or regions to be retained and clip all remaining portions. Cutting allows learners to focus their attention on relevant parts of a VMR for further analysis. It can, for instance, be beneficial for VMRs that are composed of repetitive patterns, since only a portion of the VMR is needed to understand the entire structure. Figure 12a displays part of an infinite K lattice that a learner may wish to cut. The learner may drag a window over a region to be retained and click on a “crop” button.

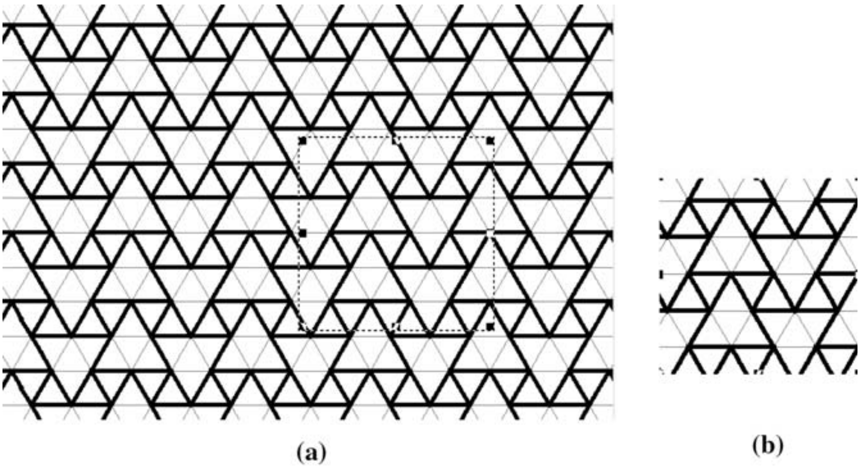

Figure 12. An infinite lattice structure cropped to show only the cropped portion.

The region that remains after the learner has cropped the lattice (b) still preserves enough visual information about the repetitive pattern for the learner to mentally extrapolate the overall structure. Here, the learner can focus on a particular portion of the lattice without getting lost within the larger configuration.

Cutting can also be useful when areas of interest are occluded by other elements of the VMR. Learners may wish to cut particular elements from the visual display in order to see underlying or internal features. For instance, cutting has been used to reveal the internal geometry of 3D doughnut surfaces by slicing them horizontally to allow learners to see that the intersection is made of two concentric circles (Banks, 1992). Slicing 3D solids can produce unique and interesting shapes. Slicing a solid, such as a cone, can produce new faces that form various types of curves, such as ellipses and parabolas. Figure 13 demonstrates a prototype idea with two VMRs: a cone and a fixed plane slicing through it. A learner can manipulate (rotate or translate) the cone to change the slicing position while the system provides continuous visual feedback. When the learner is satisfied with the position of the VMRs with respect to each other, the cut command can be issued and the upper portion of the cone can be removed, leaving the remaining portion of the solid.

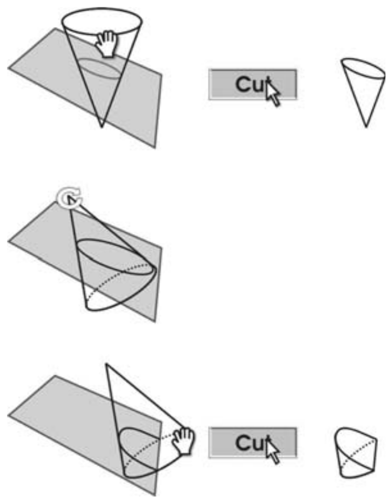

Figure 13. Fixed plane cutting through a cone.

3.2.6. *Filtering*

Filtering refers to interacting with a VMR to show, hide, or transform a selected subset of the visual elements of the VMR according to whether or not these elements possess certain characteristics and/or satisfy certain criteria. VMRs are often composed of several different types of components or contain several layers of detail and abstraction. Filtering allows learners to have control over what objects they want to view and the degree of detail and abstraction of the VMR (Strothotte, 1998; Card et al., 1999; Spence, 2001). Filtering may be used to hide elements that make a VMR appear noisy or may help learners investigate elements within a VMR that meet certain criteria. Thus, filtering can be used as an analysis, sense making, and investigation technique.

Filtering a VMR can be achieved using discrete, range-based, or magic lens filters. Discrete filters allow the learner to use toggles to specify which components of a VMR are to be displayed. Range-based filters allow the learner to specify a range of values for the visual components. If the visual components fall within the range values, then they are displayed. Discrete and range-based filters can be combined using logical operators such as "AND" and "OR". Magic lens filters allow the learner to transform portions of a VMR.

Discrete filters can support the exploration of VMRs. For example, discrete filters can be used when a learner may wish to see the nodes of a graph that have in-degrees equal to five, or to see all the faces of a geometric solid that are pentagons. Another example of a VMR that benefits from this type of interaction is 4D polytopes. Four-dimensional polytopes are often very complex, containing several sets of components such as vertices, edges, faces, and cells. As such, learners may have difficulty viewing and analyzing all these components when they are all visible. In an interactive polytope visualization tool, called Polyvise (Morey and Sedig, 2004), discrete filters are used to remove a subset of these components. For instance, Figure 14 shows a cantitruncated polytope containing 20 three-dimensional cells: 5 truncated octahedrons, 10 triangular prisms, and 5 truncated tetrahedrons. As can be seen, the VMR looks noisy (a), and it is difficult to differentiate its constituent components. After applying a discrete cell filter to the VMR, only one type of cell is shown: the 10 triangular prisms (b). Other filters can be applied to the polytope to obtain different images, hence helping learners to explore and make sense of these complex structures.

An example of the use of range-based filters is in dynamic queries (Card et al., 1999; Spence, 2001). Dynamic queries are used in information visualization applications to allow for continuous flow of interaction with visual information spaces, facilitating opportunistic, rather than pre-planned, browsing and reasoning. As a user continuously adjusts the values of query sliders, the visualization is dynamically updated to satisfy the query criteria. This interaction technique allows for dynamic exploration and immediate visual

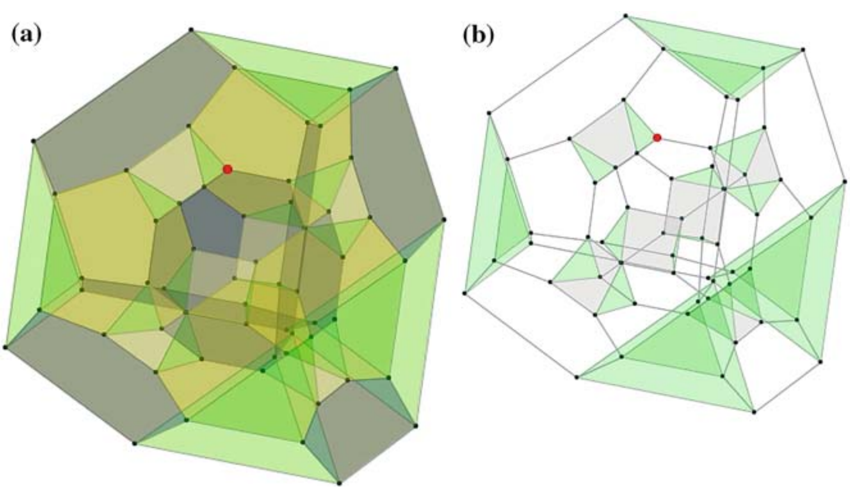

Figure 14. Discrete filtering applied to a 4D polytope in Polyvise.

feedback to the user. Figure 15 shows portions of a screen capture of PARSE (Sedig et al., 2003). The figure displays a composite VMR consisting of a set of solid transition maps, discussed before. PARSE uses dynamic queries to allow learners to explore the attributes of these solids (a): number of vertices, edges, and faces. In Figure 15b,

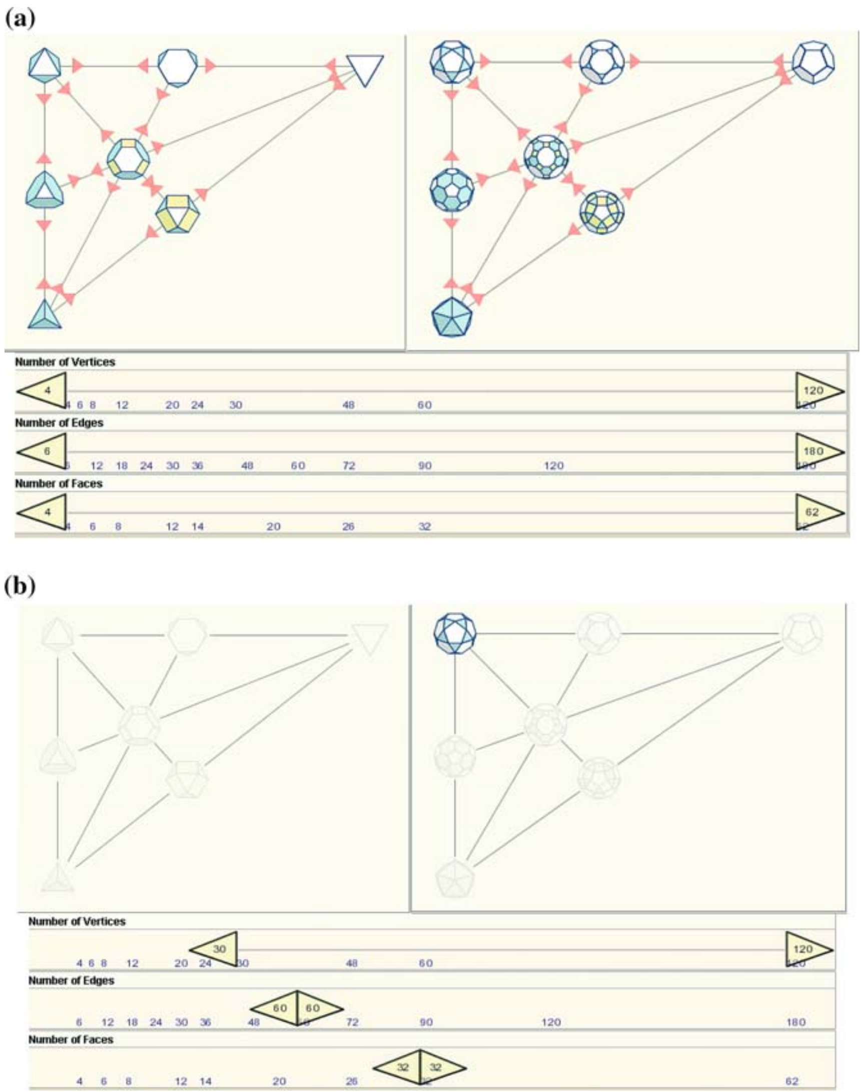

Figure 15. Dynamic filtering used to hide solids with characteristics outside a selected range.

only solids satisfying criteria specified by the sliders at the bottom are displayed – that is, solids having greater than or equal to 30 and less than or equal to 120 vertices, exactly 60 edges, and exactly 32 faces. As the learner manipulates and adjusts the values of the sliders, the learner filters the VMR, which is dynamically updated to reveal only those solids that satisfy the new criteria.

Magic lens filters are useful for interacting with layered abstractions (Card et al., 1999). These filters are usually in the form of a movable, transparent window that has a transformation operator encoded in it. Viewing any area of a VMR using a magic lens applies the transformation operator to it and thus changes the look of that area of the VMR. These operators can be any mathematical transformation, such as addition, multiplication, subtraction, or convolution. By overlapping a number of lenses on top of one another, their effects can be combined. Figure 16 shows a prototype example where a magic lens is placed over the graph of a simple function, $f(x)$. This particular filter is multiplicative; that is, the function $g(x)$

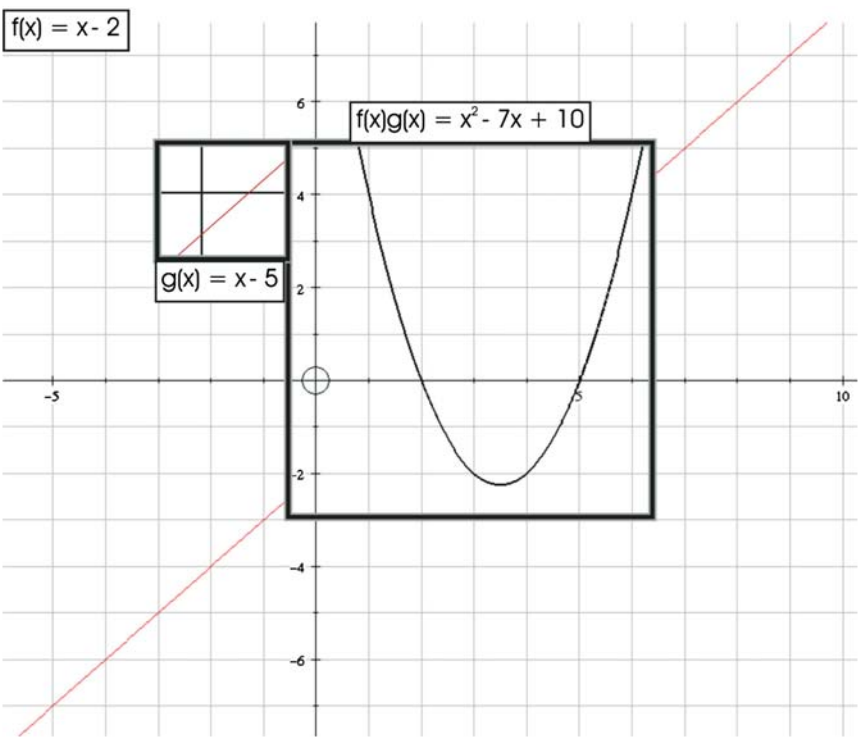

Figure 16. Magic lens filter applied to graph of a simple function.

associated with the lens is multiplied by $f(x)$, and the product of the two is seen through the lens. A thumbnail version of $g(x)$ is linked to the lens. Outside the border of the lens, the function $f(x)$ is shown without any changes. A learner can explore different areas of the graph by moving the lens to a different position over the graph. Along the top of the lens, the equation for the resulting curve is displayed.

3.2.7. *Fragmenting*

Fragmenting – whose variants are dissecting, decomposing, partitioning, segmenting, splitting, and unitizing – refers to interacting with a VMR to break it into its component or elemental parts. Fragmenting is the reverse of composing and chunking. Fragmenting can allow the learner to see component parts of a VMR, and to further interact and reason with those parts (Frederickson, 2003). Fragmenting can be used to help learners in understanding concepts such as division and fractions (Olive, 2000). A child may apply fragmenting to a set of equal line segments to explore the idea of equivalent fractions. Figure 17 shows a set of line segments of equal length each of which has been fragmented into a number of equal parts. By selecting and removing a portion of one of the segments, a learner may use the chosen fraction to measure other partitioned segments so as to verify or discover equivalence. In the figure, four line segments have been fragmented into two, three, four, and six parts, respectively. By "manually" selecting 2/3 first, the learner can

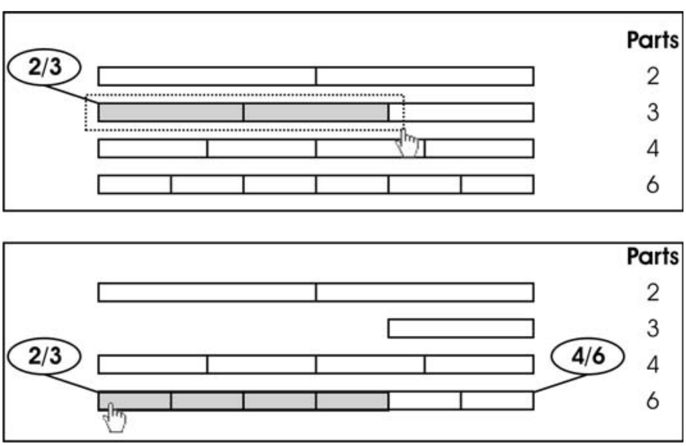

Figure 17. Fragmenting line segments to discover equivalent fractions.

then drag the fragment over the other segments to discover that it aligns equally with 4/6.

Three-dimensional objects can be fragmented so as to reveal their component faces. This fragmenting process could be thought of as breaking open the object. Learners can fragment a 3D figure in a connected or disconnected manner. Figure 18a shows how connected fragmenting is applied to an icosahedron to break it open and reveal each of its 20 faces. Although the object is fragmented, the resulting layout maintains the connections between faces and their relative positions within the larger structure as a folding (Webb, 2003). In Figure 18b, however, the object has been broken open leaving the faces disconnected. Additionally, fragmentation can be performed discretely, by clicking a button to show the open and closed state, or continuously, by dragging a slider to dynamically show the stages involved in the process of breaking the structure open.

Fragmenting may also help learners visualize the similar components or patterns that make up a VMR. Examples of such VMRs include: iterative or repetitive lattice structures, such as the K lattices; rep-*n*-tiles, polygons that can be dissected into *n* smaller copies of themselves (Weisstein, 2003); and tessellations. Figure 10a and b (above) shows the process of chunking used to create the building blocks for the lattice displayed in Figure 10c. Fragmenting, alternatively, would take the structure in Figure 10c and divide it into its elemental chunks as seen in Figure 10b. These chunks could be further fragmented to reveal their atomic structures as shown in Figure 10a.

3.2.8. *Probing*

Probing refers to interacting with a VMR to focus on, select, or drill into some aspect, property, or component of the VMR for further analysis or information (Strothotte, 1998; Card et al., 1999; Ware, 2000; Spence, 2001). Some VMRs encode information in layers of

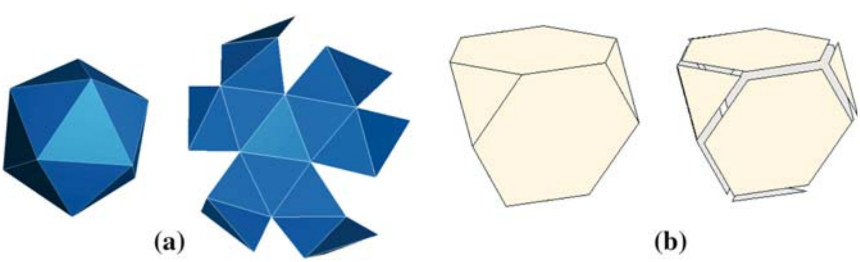

Figure 18. Connected and disconnected fragmentation.

detail and abstraction. As such, this information is not always visually explicit or accessible to the learner. Probing is a details-on-demand interaction mechanism, allowing learners to focus on some aspect of a VMR to investigate and explore it at their own pace. As a broad concept, probing encompasses other interaction techniques such as pop-up menus, panning, zooming, and magic lenses. In most situations, probing is for navigation purposes and does not alter the VMR itself.

Pop-up menus can be used to probe individual elements of a VMR by conversationally requesting additional information. The detailed information may be displayed within the VMR itself or in some other designated area of the screen. Figure 19 shows a screen capture of PARSE (Sedig et al., 2003) where the learner has requested further details about a geometric solid by right clicking the object and choosing to view the attributes of the solid (a). The details are displayed in another area of the screen (b). In this way, VMRs can be minimally displayed, reducing visual noise and allowing more space for other information while providing details-on-demand as the need arises.

Geometer's Sketchpad (Jackiw, 1995) provides pop-up-menu probing of VMRs as well. Figure 20 shows a Geometer's Sketchpad screen capture where the learner is requesting information regarding the area of a circle. The area is then displayed and dynamically maintained in another area of the screen. The same interaction technique can be used to request other information from the VMR (i.e., the circle).

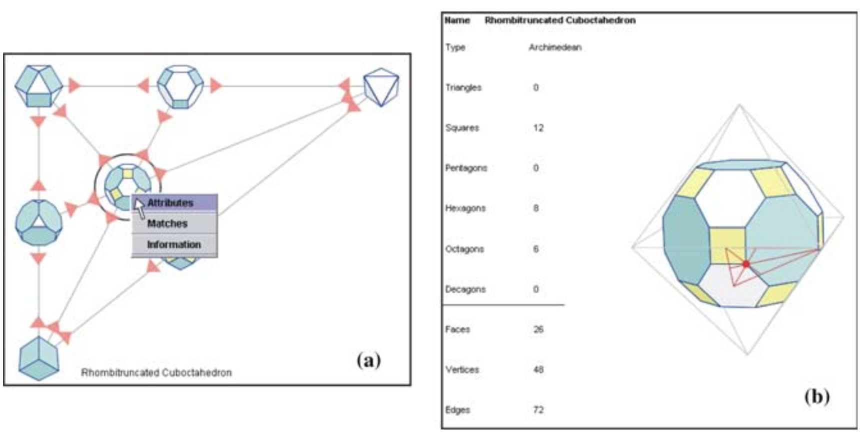

Figure 19. Probing a geometric solid to discover more details using pop-up menus.

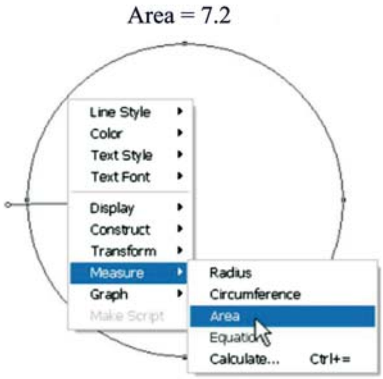

Figure 20. Probing using a pop-up menu.

Panning and zooming generally refer to the smooth movement of a limited viewport over a larger visual space (Card et al., 1999; Spence, 2001). Neither one alters the VMR or any of its portions being probed. Panning does not change the size of what is being probed, whereas zooming does. Panning a VMR can be performed using miniaturized context maps or using scrollbars. Figure 21 shows the context map (a) and detail window (b) that may be used to explore a fractal set VMR. A movable panning window is located within the context map, shown as the black rectangle. By dragging this panning window across the context map (a), the learner can see the area within the panning window dynamically displayed on another part of the screen to reveal the details within the detail window (b). This same idea could be accomplished without a context map by

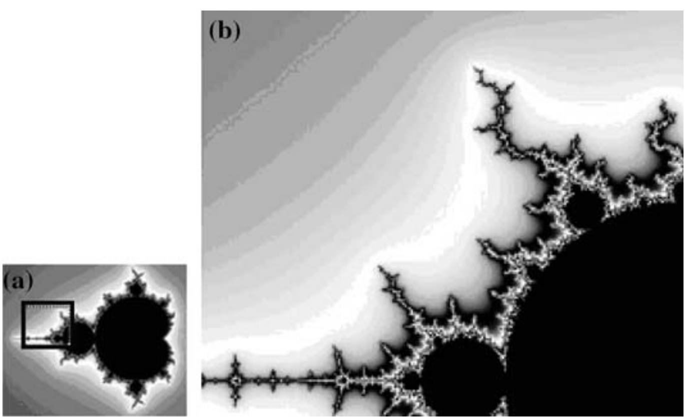

Figure 21. A context map (a) and details window (b) for panning a fractal set.

using scrollbars. In this case, the viewport has horizontal and vertical scrollbars, and only the part of the VMR passing under the viewport can be seen. It is also conceivable to have 3D scrollbars to pan through, rather than over, 3D VMRs.

Zooming changes the level of detail of a VMR being displayed and is a way of focusing on a part of a VMR. In effect, zooming brings learners closer to or further from the mathematical structure or concept they wish to analyze. Zooming can be geometric, conceptual, or semantic.

Geometric zooming alters the amount of structural detail the learner is able to see. Geometric zooming is particularly useful for geometrically deep structures such as the fractals above. A selected portion of the window can be expanded to show the details of that region. This can be accomplished through a nested action by repeating this process within a region that has already been expanded. This interaction can be useful when visual context is not important since increasing the level of detail often leads to cropping the portions of the VMR that no longer fit within the viewing window (Spence, 2001).

Conceptual zooming increases or decreases the level of detail shown regarding an abstract concept the learner is studying. While discovering the concepts of rational numbers and infinity, for example, children may benefit from exploring the number line in Figure 22 which represents a conceptually deep structure. As children select a region into which to zoom, concepts such as the division of numbers into equal parts or the infinite number of times two points can be divided are demonstrated. This can be most effectively shown

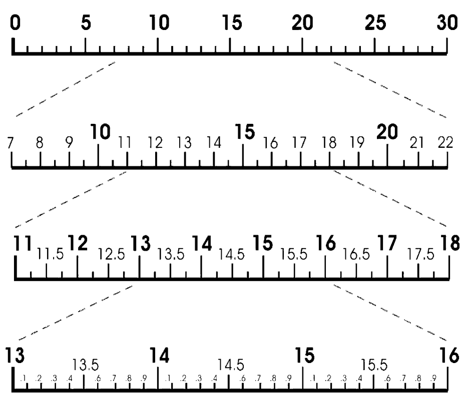

Figure 22. Several levels of conceptual zooming of a real number line.

when the flow of zooming is continuous, allowing learners to see the transition into the number space.

Semantic zooming refers to a form of zooming where probing an item changes its appearance from one notational form to another (Spence, 2001). For instance, while exploring combinatorics, learners can be given a VMR containing 720 dots, where these dots could represent apples in 2 houses on a street: each house containing 3 rooms, each room containing 4 boxes, each box containing 5 bowls, and each bowl containing 6 apples. Zooming on these dots can then change their notation from dots to apples and gradually showing their underlying divisions. Semantic zooming is a particularly powerful technique for investigating layers of abstraction encoded in VMRs, especially VMRs that, due to lack of space, pack in a great deal of information into a small visual area.

Magic lenses not only can act as filters (as discussed previously) but also as probes (Card et al., 1999; Spence, 2001). The learner can move a magic lens over a component or region of a VMR, and additional properties of the underlying component or region are revealed. Rather than a pop-up menu, the mathematical tool can provide a number of different magic lenses that probe a VMR for different types of information. One of the positive features of such lenses is that the flow of interaction can be continuous. Therefore, as the learner moves a magic lens over the VMR, the probed information is dynamically updated.

3.2.9. *Rearranging*

Rearranging – some of whose variants are sorting, ordering, sequencing, and stacking – refers to interacting with a VMR to change the spatial position and/or direction of the elements within it. Rearranging the elements of a VMR can provide learners with a better understanding of the internal relations of the VMR (Spence, 2001). Rearranging is especially useful in helping learners achieve a solution when a visual representation is in a state of flux (Preece et al., 2002). Rearranging the elements of a VMR can often help learners find a solution to a problem, allowing them to see new information or highlighting information that is otherwise difficult to see (Spence, 2001).

Rearrangement can be used to demonstrate concepts (Frederickson, 2002). Kordaki and Potari (2002) describe rearrangement as a method of helping children calculate the area of complex shapes, while indirectly demonstrating the concept of the conservation of area. An example of the use of rearrangement to demonstrate conservation of area is seen in Figure 23. This figure shows a VMR consisting of four simple triangles. In this example, the learner clicks on a vertex to select it. The selected vertex then acts as a hinge allowing the learner to rotate one of the connected triangles by direct manipulation. This process can be repeated resulting in the formation of a combination of shapes, while the area of all these shapes remains the same. Thus, rearrangement in this example can support combinatorial reasoning, an important scientific reasoning skill (Halpern, 2003). Similar to the example in Figure 23, visual rearrangement can be used to demonstrate the Pythagorean theorem.

Rearrangement of visual elements can also be a useful method for solving problems (Greeno, 1978). Elements can be organized by moving them around, sorting them, or stacking them to reveal patterns or to group elements by characteristics. Problems that require rearrangement as an essential interaction include the Chinese Tangrams (Figure 24a), the Soma Cube (Weisstein, 2003), and the 8-puzzle (Golightly, 1996). In Tangrams, for instance, given a number of 2D geometric shapes, the shapes need to be rearranged to fit into an outline without any overlaps. Generally, direct manipulation is the basic interaction used for rearranging VMRs. However, conversation can also be used to specify the spatial position and direction of visual elements, which, as was stated before, may require familiarity with the syntax of conversation. Sedig et al. (2001) have designed a tool, Super Tangrams, that allows indirect manipulation of the Tangrams

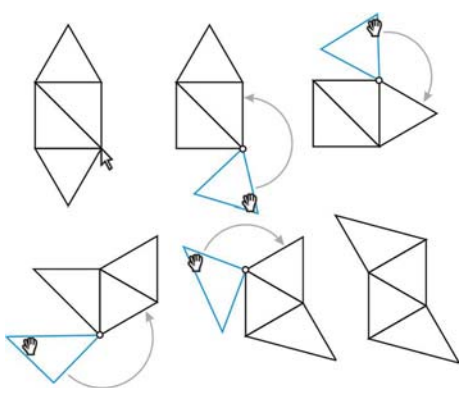

Figure 23. (*L to R and T to B*): Rearranging a VMR to demonstrate conservation of area.

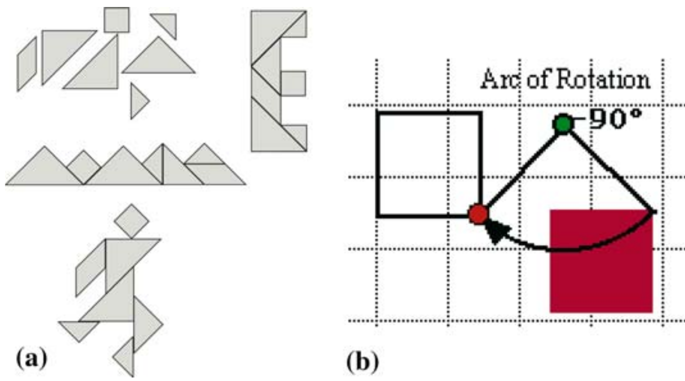

Figure 24. Chinese Tangrams (a) and rearranging a square piece using indirect manipulation (b).

pieces to rearrange them. Figure 24b shows an example of two VMRs in Super Tangrams, a square and an arc of rotation. Rather than manipulating the square directly to move it, in this example, the learner manipulates the two control points on the arc of rotation VMR, one for adjusting the center of rotation and the other for adjusting the angle of rotation. Directly manipulating these two control points on the arc representation results in the square being indirectly manipulated. Immediate feedback, in the form of a ghost image of the square, allows learners to explore what the result of their continuous adjustment of these two control points is. This style of interaction has been referred to as direct concept manipulation, as opposed to direct object manipulation. This is because if students are to focus on the concept of rotation, rather than focusing on the shape being rotated, they can directly interact with a visual representation of rotation (Sedig et al., 2001; de Souza and Sedig, 2001).

3.3. *Repicturing*

Repicturing refers to interacting with a VMR to display it in an alternative manner. Repicturing allows learners to view a VMR from different perspectives. There are three main approaches in which learners can repicture a VMR: spatially, semantically, or aesthetically. The first one is concerned with the geometry of the VMR, the second with its meaning and conceptual equivalence, and the third with the look and feel of the VMR. In spatial repicturing, the form of the representation stays the same. However, either the VMR goes through some form of geometric transformation or the space in which it exists is geometrically transformed. These transformations include rotating, scaling, magnifying, bending, folding, distorting, dilating, stretching, resizing, shrinking, morphing, and twisting (Leung and Apperley, 1994; Card et al., 1999; Morey et al., 2001; Frederickson, 2002; Weisstein, 2003). In semantic repicturing, the VMR is displayed using an alternative representational form, different form the one being seen. Allowing learners to repicture a VMR semantically can provide reasoning-congruent visual forms, thereby promoting a more flexible, diverse mathematical thinking (Ainsworth et al., 1997; Bridger and Bridger, 2001; Gadanidis et al., 2004; Labeke, 2001; Scher and Goldenberg, 2001; Sedig et al., 2005). In aesthetic repicturing, the form of the VMR stays the same, but learners have control over such things as the color of its components, the degree of lighting of the VMR, and/or the angle of lighting, according to the learners' perceptual or cognitive preferences.

Figure 25 shows a simple example of spatial repicturing in the form of twisting. In this figure, the learner applies a spatial transformation to a cylinder (a) by manipulating a slider control to twist the cylinder, resulting in continuous, dynamic generation and visualization of a series of double-conic shapes, with the final one seen in Figure 25b. Spatial repicturing is also employed as an interactive technique in Polyvise (Morey and Sedig, 2004). Figure 26 shows a screen capture of Polyvise, whereby learners can spatially repicture a

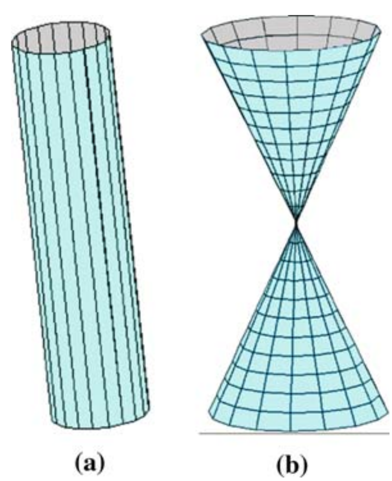

Figure 25. Repicturing a cylinder by twisting it.

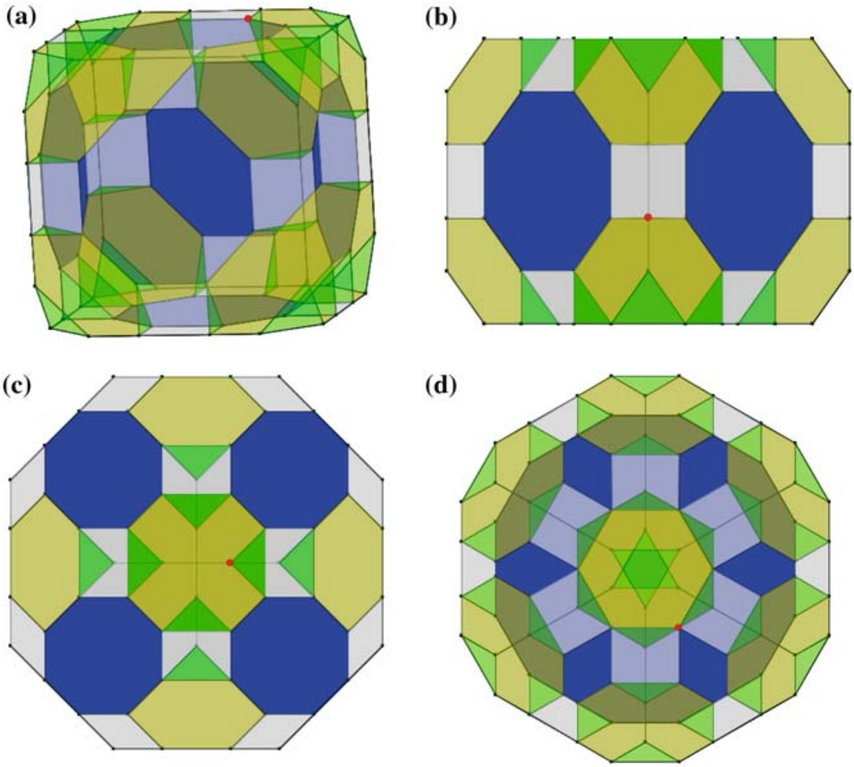

Figure 26. Repicturing a 4D polytope by reorienting and visual stacking.

complex cantitruncated hypercube polytope (a) to view it from different orientations (b–d). By stacking many of the elements of the VMR – such as vertices, lines, faces, and cells – Polyvise highlights symmetries of the polytope. These repictured visualizations can act as starting landmark positions from which learners can begin to explore a polytope, reducing the possibility of getting lost in the 4D hyperspace (ibid.). It is possible to imagine numerous scenarios in which learners interactively transform VMRs, such as lines, surfaces, solids, and knots, to explore and investigate the effect of different spatial repicturings on these VMRs.

VMRs can also be repictured indirectly by applying transformations to the coordinate system or space in which they reside. This is usually done by distorting or deforming the visual space (Card et al., 1999; Spence, 2001). Within the field of information visualization, various types of distortions have been investigated and effectively applied to representations in ways that facilitate exploration of large information spaces (Leung and Apperley, 1994; Spence, 2001). In situations where details are context-sensitive and screen space is limited, distortion is a useful technique allowing learners to quickly explore details of a VMR without losing the larger context. Distortion maintains visual and psychological continuity since learners are able to see connections between the local detail and overall context (Card et al., 1999). There are various techniques used to distort the visual space such as fisheye, rubber sheet, *x–y* distortion, and suppression (Spence, 2001).

Figure 27 shows an example of spatial repicturing of a VMR using distortion of the VMR's space. The figure displays four snapshots of a Euclidian and several non-Euclidean planes. The figure shows discrete moments of the progression from a traditional Euclidian

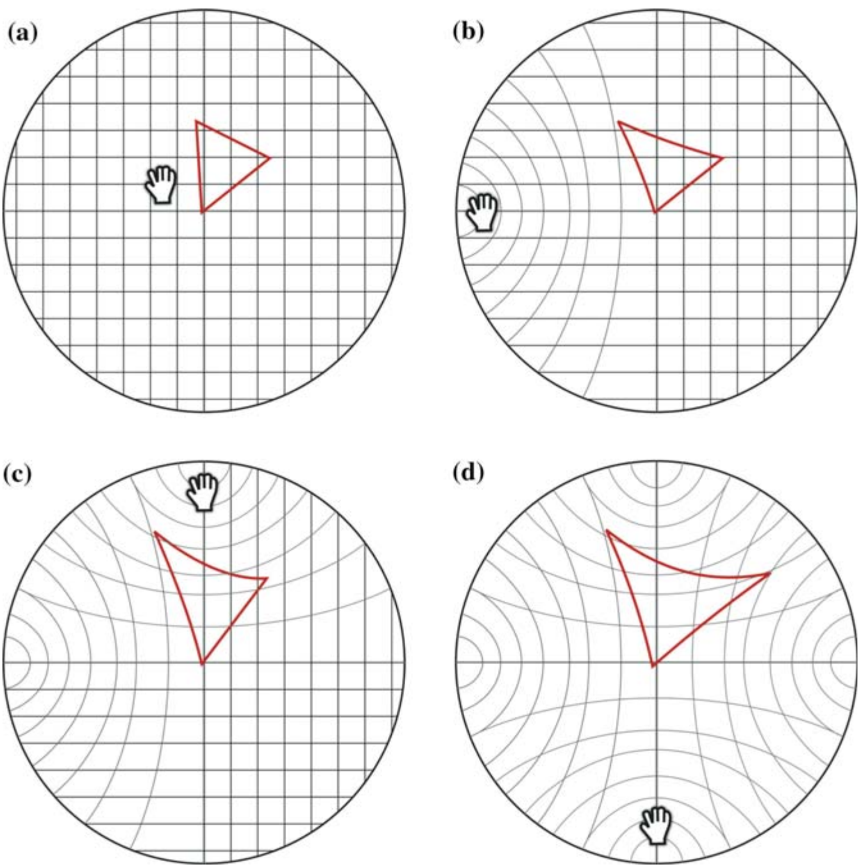

Figure 27. Repicturing a triangle by distorting the Euclidian plane.

coordinate system (a) to an infinite hyperbolic system (d). By dragging a pointer along either axis, a learner can investigate and compare these two geometries. The dragging motion stretches the Euclidian plane into an infinite Non-Euclidian space and redraws any figure that lies within the space to align with the new system. In this example, the learner has drawn a triangle (a). As the learner drags the pointer to the left within the left half of the circle, the coordinate system as well as the triangle is stretched. Repeating this step in each of the four directions, the learner has created a hyperbolic space (d) and the triangle has been transformed to conform to the properties of the new Non-Euclidian space.

Finally, as was stated, sometimes more than one form of representation is needed to achieve better reasoning and understanding when interacting with a VMR. Figure 2 (above) shows an example of semantic repicturing to produce two forms of representation of a K lattice (Sedig et al., 2005). The first (a) is a finite-state transition diagram representation of a K lattice, and the second (b) is a geometric representation of the VMR. Learners can experiment with K-lattice structures by navigating the Hamiltonian paths of their state-transition diagrams, thereby gaining insight into not only their geometric but also their procedural properties (ibid.). There are numerous VMRs that can be repictured semantically such as functions, graphs, structures and diagrams.

3.3.1. *Scoping*

Scoping refers to interacting with a VMR to adjust its field of view (Sedig and Morey, 2005). By dynamically increasing or decreasing the field of view (i.e., scope) of a VMR, learners can locate its starting point of growth and discover the process and sequence of its growth and construction from the starting point. The scope of a VMR is the amount of visual structure displayed or the number of elements that are in view at a given time. There may be instances where the learner may have difficulty making sense of the components or the process by which the whole VMR has come into existence. In such situations, a learner may want to adjust the scope of the VMR, based on its compositional chunks or its atomic elements, to discover how the whole structure is constructed or how the smaller building blocks are put together.

Figure 28 provides a simple example of scoping in a Logo-like application, where four snapshots of the scoping process are shown.

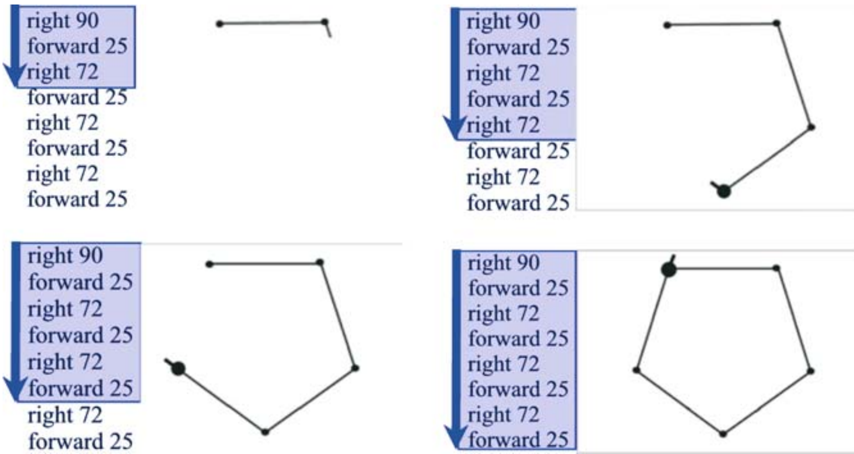

Figure 28. Varying the scope of a geometric shape to explore how it is constructed.

In this example, given the linguistic code and the resultant VMR (i.e., the pentagon), the learner continuously interacts with a vertical slider to adjust the scope of the VMR. Through navigating the sequential code, as the learner moves the slider downward along the code, the VMR is dynamically constructed, i.e. the scope is increased. Conversely, as the slider moves upward, the VMR is deconstructed, i.e. the scope is decreased. In this way, the learner can analyze each step of the construction process. Figure 29 shows a similar but more complex example of scoping, taken from screen captures of PolygonR&D, a tool for visualizing and exploring tiling patterns (Morey and Sedig, 2004). In this figure a visual tiling pattern is displayed at three levels of the scoping process. Finally, Figure 30 shows scoping applied to a complex 4D polytope. The polytope is made of many 3D cells. Trying to visualize and make sense of 4D polytopes can be a difficult task, as seen in the last snapshot in Figure 30. Using a slider control provided in Polyvise (Morey and Sedig, 2004), learners can dynamically adjust the scope of 4D polytopes to analyze their structures. Scoping can be a powerful technique for allowing learners to explore the construction process of complex VMRs. It may be very difficult or impossible to explore the structural make-up of complex VMRs when viewing them in full scope. By providing continuous flow and control over the pace of interaction, scoping can complement composing.

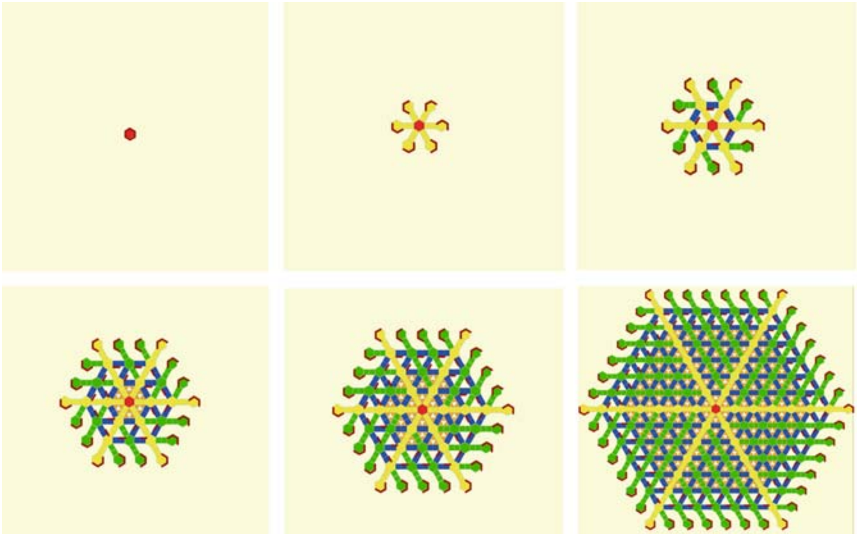

Figure 29. Scoping applied to a tiling pattern to explore its growth and construction.

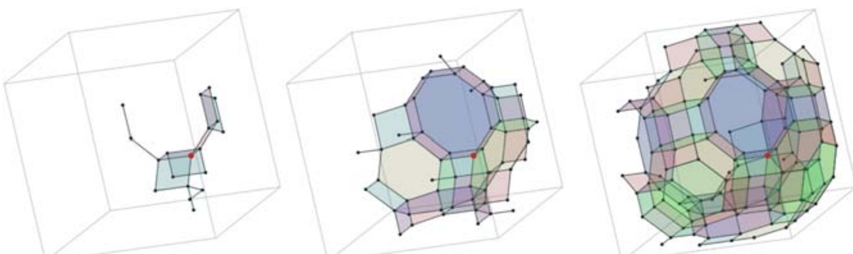

Figure 30. Scoping applied to a 4D polytope structure to investigate its 3D cells and 2D faces.

3.3.2. *Searching*

Searching refers to interacting with a VMR to seek out the existence of or locate the position of specific features, elements, or structures within the VMR. Searching, unlike browsing, is a directed activity, where the learner is actively attempting to answer questions or develop understanding around a particular question or idea (Bates, 1986). Although when searching a VMR the mathematical tool may perform the act of locating what the learner is searching for, nonetheless, learners must know what they intend to find and be able to express their intention via interaction. This requires a reasonable understanding of the VMR as well as the language of interaction that is used to query the VMR. Searching may be necessary when learners cannot readily detect the object of their search. When searching a VMR, learners must both generate the search and evaluate the results. Results of a search may be displayed with or without their associated context. Elements satisfying a query may be highlighted within the VMR or listed separately.

Figure 31 shows an example of searching where a learner can fill out a form to search for prime numbers in Pascal's Triangle. The result is displayed within the overall context of the VMR by highlighting the elements that satisfy the search query. Figure 32 shows a prototype of how a learner can search a number line for prime numbers. In this example, the learner navigates the VMR using a discrete attribute-walk along the number line by hopping from one prime number to the next, a technique that can support inductive reasoning about these numbers.

Finally, Figure 33 shows another possible example of searching. Searching for patterns and properties within graphs can sometimes be a difficult task, even for relatively simple graphs. This figure shows a straightforward graph structure. A learner, investigating the Traveling Salesman problem, may input a sequence of numbers specifying the Hamiltonian path of the graph. The learner may then converse with the VMR via a menu or otherwise to request it to search for and

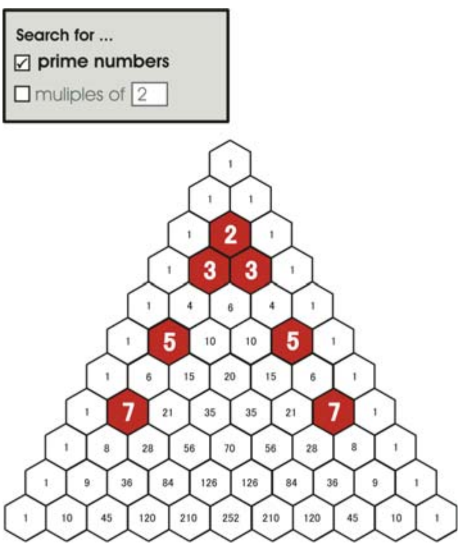

Figure 31. Searching Pascal's Triangle for prime numbers.

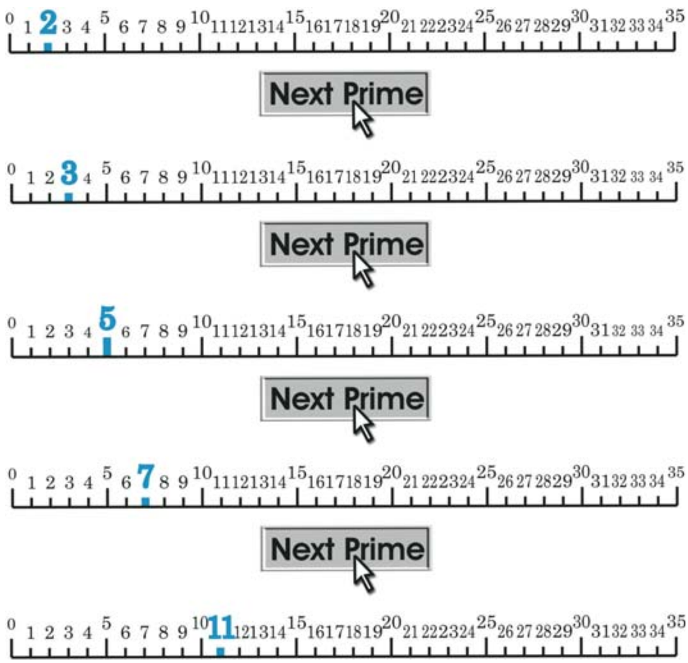

Figure 32. Searching a number line for prime numbers using attribute walking.

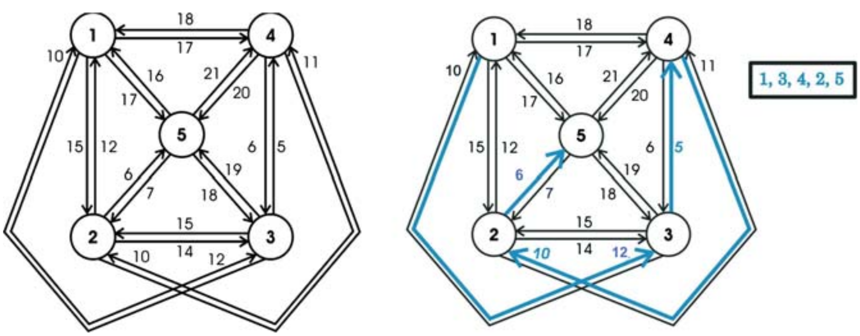

Figure 33. Searching for the Hamiltonian path of a graph.

display this path. The resulting path can then be highlighted within the graph and the ordering of the nodes be displayed so that the learner can validate the input values.

3.4. *Combining Task-based Interactions*

As was stated before, task-based interactions can be combined in mathematical cognitive tools to allow learners to perform coordinated and integrated tasks. In such cases, the tool provides learners with several interactions so as to support diverse cognitive activities of the learners (Sedig et al., 2003). This section provides two brief examples to show how these interactions can be combined.

The first example is from a tool called TileLand (Sedig et al., 2002; Travaglini, 2003). This tool provides learners with 3 interactions: composing, automatic annotating of actions, and scoping. By clicking on command buttons (Figure 34a), polygonal tilings are composed (Figure 34c). Learner's actions, as they are being performed, are automatically added to a code panel (Figure 34b). The learner can then review the annotated code and interact with it via an arrow (Figure 34b) to scope the composed VMR.

The second example is shown in Figure 35 where probing and annotating are combined to explore the first derivative of a function. The figure shows snapshots of three stages of interaction. As the learner navigates the function graph, continuously using a sliding control, two events occur simultaneously: (1) each point on the graph is probed for its derivative, displaying its value, and (2) the VMR is

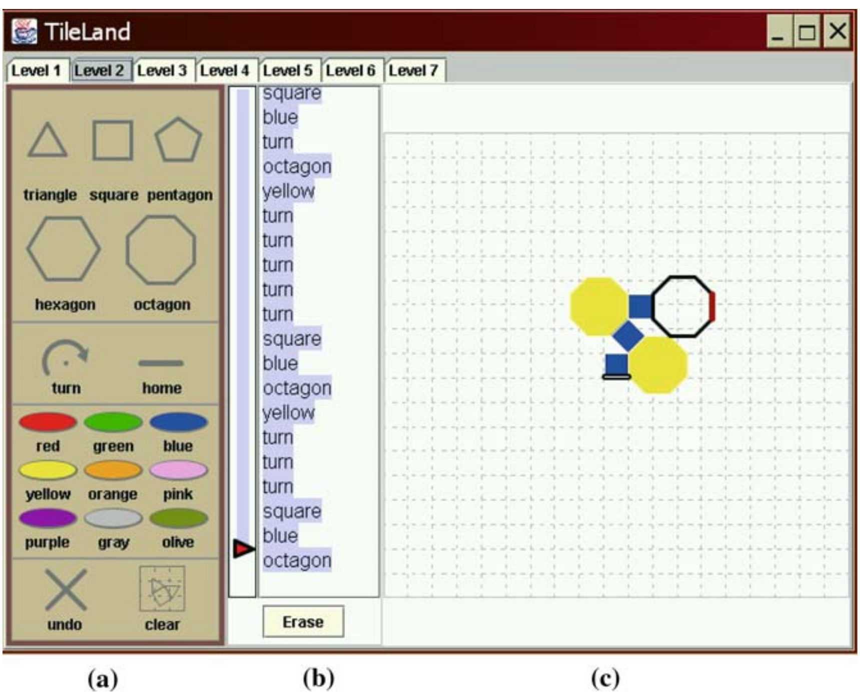

Figure 34. Combining interactions in TileLand.

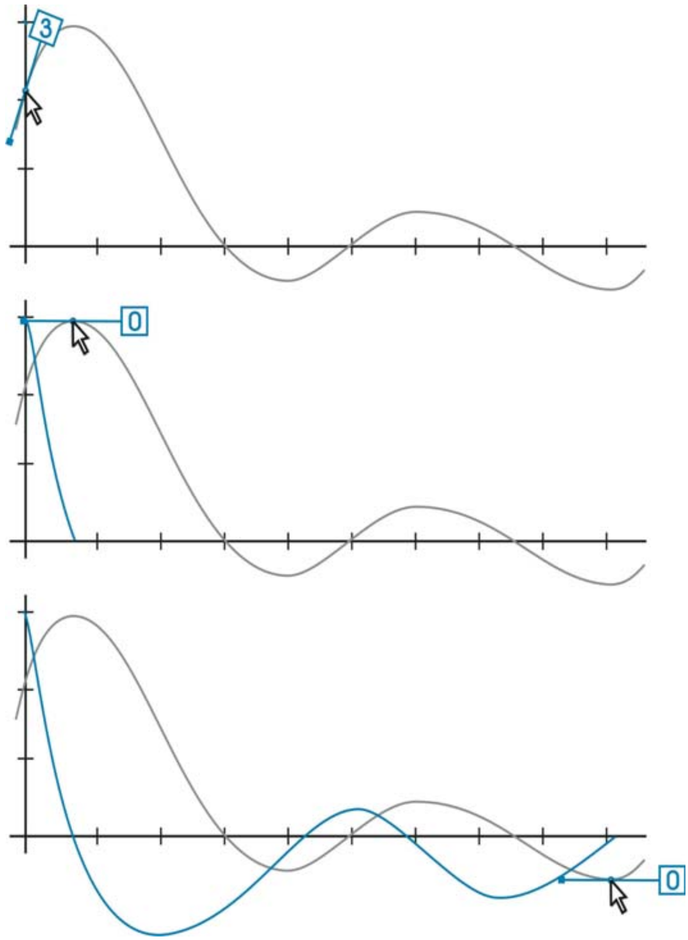

Figure 35. Combining continuous probing and automatic annotation to explore a function graph.

automatically annotated with another graph representing its derivative. In this example, while navigating the graph, one can think of the tangents with value 0 as the landmarks of the newly-generated annotation graph.

4. SUMMARY AND FUTURE WORK

Visual mathematical representations (VMRs) are used in many areas of mathematics. Interacting with VMRs means allowing learners to act upon VMRs and receive some form of reactive feedback. Many micro-level interaction techniques have been devised in the context of disciplines such as human–computer interaction and information visualization to interact with on-screen representations. These techniques can be used to render VMRs interactive. However, currently, these techniques are scattered across and reported in different disciplines and are often described and characterized in terms of the domain for which they were developed. This can make it difficult for designers of mathematical cognitive tools to know how they can be applied to VMRs. Additionally, these techniques are not organized and categorized according to their common goals and characteristics.

This paper has presented a categorization and characterization of the different ways learners can interact with VMRs. Two categories of interaction were presented: basic and task-based. Basic interactions are based on root metaphors of mouth/talking, hands/handling, and feet/walking. The three basic interactions are

1. *Conversing*: Talking to a VMR using symbolic, lexical expressions or commands
2. *Manipulating*: Handling a VMR using a pointing cursor
3. *Navigating*: Moving on, over, or through a VMR

Task-based interactions are based on the low-level cognitive tasks in which learners engage when thinking and reasoning about visual representations. Task-based interactions are built on top of the basic interactions – that is, one or more variations of the basic interactions contribute to the creation of these interactions. Additionally, task-based interaction can themselves be combined. The task-based interactions are

1. *Animating*: Generating movement within a VMR
2. *Annotating*: Augmenting a VMR by placing notes or marks on it
3. *Chunking*: Grouping a number of similar or related, but disjointed, visual elements
4. *Composing*: Putting together separate visual elements to create a VMR
5. *Cutting*: Removing unwanted or unnecessary portions of a VMR
6. *Filtering*: Showing, hiding, or transforming a selected subset of the visual elements of a VMR according to certain characteristics or criteria
7. *Fragmenting*: Breaking a VMR into its component or elemental parts

8. *Probing*: Focusing on or drilling into some aspect, property, or component of a VMR for further analysis and information
9. *Rearranging*: Changing the spatial position and/or direction of elements within a VMR
10. *Repicturing*: Displaying a VMR in an alternative manner
11. *Scoping*: Changing the degree to which a VMR is visually constructed/deconstructed by adjusting its field of view
12. *Searching*: Seeking out the existence of or position of specific features, elements, or structures within a VMR

As can be observed, this paper is not prescriptive. That is, given certain tasks, VMRs, and learners, the paper does not provide guidelines as to which interactions are most appropriate to use. This is because research and findings in this area are limited and *ad hoc*. For instance, Sedig et al. (2001) conducted an empirical study comparing different manipulation styles and evaluated their effects on student cognition and concept learning. They found that adding scaffolding to direct manipulation of representations of transformation geometry concepts significantly improved student learning. But, further semiotic analysis showed that this type of interaction is appropriate for and lends itself to certain types of representations (de Souza and Sedig, 2001). Hence, it is difficult to make generalized statements about the findings. Part of the challenge of analysis, evaluation, and comparison of the techniques is the lack of a proper conceptual framework and language in this area. Knowing what interactions exist and developing a language to describe them, as presented in this paper, can provide the first step in the creation of a systematic framework. The next step is to know when to use the interactions, and how to effectively operationalize them so as to maximize cognitive task support and learning. Currently, we do not have access to such knowledge. To help this situation, a possible line of future research is to develop a prescriptive taxonomy (Sedig, 2004). This type of taxonomy prescribes design rules and guidelines for the elements of a problem space and can provide best-practice examples of existing systems and techniques. Given different VMRs, cognitive tasks, and learners, the taxonomy would guide a designer of a mathematical tool to select and use interaction techniques that may best support the design requirements and specifications. Some elements of such a taxonomy may include: (1) organization of VMRs according to their features, (2) analysis of the cognitive tasks involved in mathematics problem solving and learning, and (3) general rules for when and how to use which interactions, suggesting best-practice tools and designs to be consulted.

Developing such a taxonomy is not easy. Much empirical evaluation of tools and techniques is needed to validate and refine such a framework. It also requires collaboration and interdisciplinary research on the part of mathematics educators and human–computer interaction designers. As interactive mathematical tools in the form of online applets are now being delivered on the Internet, they can influence the mathematical thinking of millions of learners. It is important that these tools be designed properly. In an interdisciplinary analysis of a VMR-based applet from the National Council of the Teacher's of Mathematics' Illuminations Web site, Gadanidis et al. (2004) observe that "many applets, as is the case with the NCTM applet, do not appear to be well designed, neither from a pedagogical nor from an interface design perspective." It is hoped that this paper – presenting a diverse set of interactions, categorizing, describing and characterizing these interactions, and demonstrating their use in exploration of VMRs – stimulates future systematic research in this area to improve the design of mathematical cognitive tools.

ACKNOWLEDGMENTS

We would like to thank the reviewers of this article for their valuable comments. We would also like to thank Jim Morey, Hai-Ning Liang, and Sonja Rowhani for their ideas and contributions. This research has been funded by the Natural Sciences and Engineering Research Council of Canada.

NOTES

[1] In this paper, the terms learner, user, problem solver, explorer, and investigator convey the same meaning.

[2] These tools, also called cognitive technologies or mindtools, are intended to support human cognitive processes and thinking. Examples of these tools include interactive visualization software to explore patterns in a body of information, mind mapping tools to help externalize and organize thoughts and concepts, and online interactive mathematical applets to investigate how velocity and position graphs relate.

[3] The nodes of this diagram represent Ks having different orientations, and its links represent how these Ks can be connected.

[4] The 8-puzzle is a game consisting of a 3 × 3 square grid. Eight of the squares have numbers from 1 to 8, and one of the squares is empty. This allows for moving the other 8 squares around into different positions until the squares are arranged in an ascending order, with the last square empty.

REFERENCES

Ainsworth, S.E., Bibby, P.A. and Wood, D.J. (1997). Information technology and multiple representations: New opportunities – new problems. *Journal of Information Technology for Teacher Education* 6(1): 93–105.

Arcavi, A. and Hadas, N. (2000). Computer mediated learning: An example of an approach. *International Journal of Computers for Mathematical Learning* 5: 25–45.

Banks, D. (1992). Interactive manipulation and display of two-dimensional surfaces in four-dimensional space. *Computer Graphics* 25(2): 197–207.

Barwise, J. and Etchemendy, J. (1998). Computers, visualization, and the nature of reasoning. In T.W. Bynum and J.H. Moor (Eds), *The Digital Phoenix: How Computers are Changing Philosophy* (pp. 93–116). London: Blackwell.

Bates, M.J. (1986). An exploratory paradigm for online information retrieval. In B.C. Brookes (Ed.), *Intelligent Information Systems for the Information Society*, Amsterdam: North-Holland.

Beynon, M., Nehaniv, C.L. and Dautenhahn, K., (Eds.) (2001). *Cognitive Technology: Instruments of Mind – Proceedings of the International Cognitive Technology Conference*. Warwick, UK: Springer-Verlag.

Botsmanova, M.E. (1972). On the role of graphic analysis in solving arithmetic problems. In J. Kilpatrick and I. Wirzup (Eds), *Soviet Studies in the Psychology of Learning and Teaching Mathematics* (Vol. 6, pp. 119–123). Chicago: University of Chicago Press.

Bridger, M. and Bridger, M. (2001). Mapping diagrams: Another view of functions. In A.A. Cuocoand and F.R. Curcio (Eds), *The Roles of Representation in School Mathematics: 2001 Yearbook*, Reston, VA: National Council of Teachers of Mathematics.

Card, S., MacKinlay, J. and Shneiderman, B. (1999). *Readings in Information Visualization: Using Vision to Think*. San Francisco: Morgan Kaufmann Publishers.

Chen, C. (1999) *Information Visualization and Virtual Environments*. London, UK: Springer-Verlag.

Cheng, P. (2002). Electrifying diagrams for learning: Principles for complex representational systems. *Cognitive Science* 26: 685–736.

Cuoco, A.A. and F.R., Curcio, (Eds.) (2001). *The Roles of Representation in School Mathematics: 2001 Yearbook*. Reston, VA: National Council of Teachers of Mathematics.

Davidson, J.E. and R.J., Sternberg, (Eds.) (2003). *The Psychology of Problem Solving*. Cambridge, UK: Cambridge University Press.

de Souza, C.S. and Sedig, K. (2001). Semiotic considerations on direct concept manipulation as a distinct interface style for learnware. In Proceedings of the

Brazilian Human–Computer Interaction Conference (IHC2001) (pp. 229–241), 15–17 October. Florianopolis, Santa Catarina, Brazil.

Diezmann, C.M. and English, L.D. (2001). Promoting the use of diagrams as tools for thinking. In A.A. Cuocoand and F.R. Curcio (Eds), *The Roles of Representation in School Mathematics: 2001 Yearbook*, Reston, VA: National Council of Teachers of Mathematics.

Dix, A.J. and Ellis, G. (1998). Starting simple – adding value to static visualization through simple interaction. In Proceedings of the 4th International Working Conference on Advanced Visual Interfaces (AVI' 98) (pp. 124–134). L'Aquilla, Italy. New York: ACM Press.

English, L.D., (Eds.) (1997). *Mathematical Reasoning: Analogies, Metaphors and Images*. Mahwah, NJ: Lawrence Erlbaum Associate.

Fisher, R. (1990) *Teaching Children to Think*. Oxford: Basil Blackwell.

Frederickson, G. (2002) *Hinged Dissections: Swinging and Twisting*. Cambridge, MA: Cambridge University Press.

Frederickson, G. (2003) *Dissection: Plane and Fancy*. Cambridge, MA: Cambridge University Press.

Gadanidis, G, Sedig, K, & Liang, H (2004). Designing online mathematical investigation. Journal of Computers in Mathematics and Science Teaching, 23(3), 275–298.

Glasgow, J., Narayanan, N.H. and Chandrasekaran, A.B., (Eds.) (1995). *Diagrammatic Reasoning: Cognitive and Computational Perspectives*. Cambridge, MA: The MIT Press.

Gobet, F., Lane, P., Croker, S., Cheng, P., Jones, G., Oliver, I. and Pine, J. (2001). Chunking mechanisms in human learning. *Trends in Cognitive Sciences* 5(6): 236–243.

Golightly, D. (1996). Harnessing the interface for domain learning. In M.J. Tauber (Ed), Proceedings of the CHI '96 Conference Companion on Human Factors in Computing Systems: Common Ground (CHI '96) (pp. 37–38), 13–18 April. Vancouver, British Columbia, Canada. New York: ACM Press.

Golledge, R.G. (1999). Human wayfinding and cognitive maps. In R.G. Golledge (Ed.), *Wayfinding Behavior: Cognitive Mapping and Other Spatial Processes* (pp. 5–45). Baltimore, London: The Johns Hopkins University Press.

Gonzalez-Lopez, M. (2001). Using dynamic geometry software to simulate physical motion. *International Journal of Computers for Mathematical Learning* 6: 127–142.

Greeno, J.G. (1978). Natures of problem-solving abilities. In W.K. Estes (Ed.), *Handbook of Learning and Cognitive Processes* (5pp. 239–270). Hillsdale, NJ: Lawrence Erlbaum Associates.

Halpern, D.F. (2003) *Knowledge and Thought,* 4th edn. Hillsdale, NJ: Lawrence Erlbaum Associates.

Hanna, G. (2000). Proof, explanation and exploration: An overview. *Educational Studies in Mathematics* 44: 5–23.

Hansen, Y.M. (1999). Graphic tools for thinking, planning, and problem solving. In R. Jacobson (Ed.), *Information Design* (pp. 193–221). Cambridge, MA: The MIT Press.

Hanson, A. and Ma, H. (1995). Space walking. In Proceedings of IEEE Visualization '95 (pp. 126–133), Atlanta, GA, USA.

Hanson, A., Munzner, T. and Francis, G. (1994). Interactive methods for visualizable geometry. *IEEE Computer* 27(7): 73–83.

Hazzan, O. and Goldenberg, P. (1997). Students' understanding of the notion of function in dynamic geometry environments. *International Journal of Computers for Mathematical Learning* 1: 263–291.

Healy, L. and Hoyles, C. (2001). Software tools for geometrical problem solving: Potentials and pitfalls. *International Journal of Computers for Mathematical Learning* 6: 235–256.

Hitt, F. (Ed.) (2002). Representations and mathematics visualization. In Proceedings of the Twenty-Fourth North American Chapter of the International Group for the Psychology of Mathematics Education, Georgia, USA.

Holst, S.J. (1996). Directing learner attention with manipulation styles. In Tauber M.J. (Ed.), Proceedings of the CHI '96 Conference Companion on Human Factors in Computing Systems: Common Ground (CHI '96) (pp. 43–44), 13–18 April. Vancouver, British Columbia, Canada. New York: ACM Press.

Holzl, R. (1996). How does dragging affect the learning of geometry?. *International Journal of Computers for Mathematical Learning* 1: 169–187.

Hutchins, E.L., Hollan, J.D. and Norman, D.A. (1986). Direct manipulation interfaces. In D.A. Norman and S.W. Draper (Eds), *User Centered System Design: New Perspectives in Human–Computer Interaction*, Hillsdale, NJ: Lawrence Erlbaum Associates.

Jackiw, N. (1995) *The Geometer's Sketchpad, v. 3.0*. Berkeley, CA: Key Curriculum Press.

Jonassen, D.H., Beissner, K. and Yacci, M., (Eds.) (1993). *Structural Knowledge: Techniques for Representing, Conveying, and Acquiring Structural Knowledge*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Jones, K. (2000). Providing a foundation for deductive reasoning: Students' interpretations when using dynamic geometry software and their evolving mathematical explanations. *Educational Studies in Mathematics* 44: 55–85.

Jones, S. and Scaife, M. (2000). Animated diagrams: An investigation into the cognitive effects of using animation to illustrate dynamic processes. In M. Anderson and P. Cheng (Eds), *Theory and Applications of Diagrams – Lecture Notes in Artificial Intelligence no. 1889* (pp. 231–244). Berlin: Springer-Verlag.

Kaput, J. (1993). Urgent need for proleptic research. In T.A. Romberg, E. Fennema and T.P. Carpenter (Eds), *Integrating Research on the Graphical Representation of Functions*, Hillsdale, NJ: Lawrence Erlbaum Associates.

Keller, P.R. and Keller, M.M. (1993). *Visual Cues: Practical Data Visualization*. Los Alamitos, CA: IEEE Computer Society Press.

Kordaki, M. and Potari, D. (2002). The effects of area measurement tools on student strategies: The role of a computer microworld. *International Journal of Computers for Mathematical Learning* 7: 65–100.

Labeke, N.V. (2001). Multiple external representations in dynamic geometry: A domain-informed design. In S. Ainsworth and R. Cox (Eds.), AI-ED 2001, Workshop Papers, External Representations of AIED: Multiple Forms and Multiple Roles (pp. 24–31), 20 May, Texas, USA.

Lajoie, S.P., (Eds.) (2000). *Computers as Cognitive Tools*. New Jersey: Lawrence Erlbaum Associates.

Larkin, J. and Simon, H. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science* 11: 65–99.

Laurillard, D. (1993) *Rethinking University Teaching: A Framework for the Effective Use of Educational Technology*. London: Routledge Falmer.

Leung, Y.K. and Apperley, M.D. (1994). A review and taxonomy of distortion-oriented presentation techniques. *ACM Transactions on Computer–Human Interaction* 1(2): 126–160.

Leung, A. and Lopez-Real, F. (2002). Theorem justification and acquisition in dynamic geometry: A case of proof by contradiction. *International Journal of Computers for Mathematical Learning* 7: 145–165.

Marrades, R. and Gutierrez, A. (2000). Proofs produced by secondary school students learning geometry in a dynamic computer environment. *Educational Studies in Mathematics* 44: 87–125.

Matsuda, N. and Toshio, O. (1998). Diagrammatic reasoning for geometry ITS to teach auxiliary line construction problems. In Proceedings of the International Conference on Intelligent Tutoring Systems (pp. 244–253), Canada.

Morey, J. and Sedig, K. (2003). Archimedean kaleidoscope: A cognitive tool to support thinking and reasoning about geometric solids. In M. Sarfraz (Ed.), *Geometric Modeling: Techniques, Applications, Systems and Tools*, Berlin: Kluwer Academic Publisher.

Morey, J. and Sedig, K. (2004). Using indexed-sequential geometric glyphs to explore visual patterns. In Proceedings of Interactive Visualisation and Interaction Technologies, ICCS 2004 (pp. 996–1003), June, Krakow, Poland.

Morey, J. and Sedig, K. (2004). Adjusting degree of visual complexity: An interactive approach for exploring four-dimensional polytopes. The Visual Computer: *International Journal of Computer Graphics* 20, 1–21. Berlin: Springer-Verlag.

Morey, J., Sedig, K. and Mercer R. (2001). Interactive metamorphic visuals: Exploring polyhedra relationships. In Proceedings of the 5th International Conference on Information Visualization (pp. 483–488). England: IEEE Computer Society Press.

Morrison, J., Tversky, B. and Betrancourt, M. (2000). Animation: Does it facilitate learning? In Proceedings of the AAAI 2000 Spring Symposium Smart Graphics (pp. 53–60), Stanford, CA, USA.

Moyer, P., Bolyard, J. and Spikell, M. (2001). Virtual manipulatives in the K-12 classroom. In A. Rogerson (Ed.), Proceedings of the International Conference on New Ideas in Mathematics Education (pp. 184–187), Australia.

NCTM (2000). *Principles and Standards for School Mathematics*. Reston, VA: The National Council of Teachers of Mathematics.

Norman, D.A. (1993) *Things that Make Us Smart: Defining Human Attributes in the Age of the Machine*. New York: Addison-Wesley.

Norman, D.A. and Draper, S., (Eds.) (1986). *User Centered System Design: New Perspectives in Human–Computer Interaction*. Mahwah, NJ: Lawrence Erlbaum Associates.

Olive, J. (2000). Computer tools for interactive mathematical activity in the elementary school. *International Journal of Computers for Mathematical Learning* 5: 241–262.

Ormrod, J.E. (1995) *Human Learning*. Englewood Cliffs, NJ: Prentice-Hall Inc.

Otero, N., Rogers, Y. and du Boulay, B. (2001). Is interactivity a good thing? Assessing its benefits for learning. In Proceedings of the 9th International Conference on HCI (pp. 790–794), New Orleans, USA. New Jersey: Lawrence Erlbaum Associates.

Paivio, A. (1983). The empirical case for dual coding. In J.C. Yuille (Ed.), *Imagery, Memory and Cognition*, Hillsdale, NJ: Lawrence Erlbaum Associates.

Papert, S. (1980) *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books.

Park, O. (1998). Visual displays and contextual presentations in computer-based instruction. *Educational Technology, Research and Development* 46(3): 37–51.

Parker, G., Franck, G. and Ware, C. (1997). Visualization of large nested graphs in 3D: Navigation and interaction. *Journal of Visual Languages and Computing* 9(5): 299–317.

Peper, R.J. and Mayer, R.E. (1986). Generative effects of note-taking during science lectures. *Journal of Educational Psychology* 78: 34–38.

Peterson, D., (Eds.) (1996). *Forms of Representation*. Exeter, UK: Intellect Books.

Pettersson, R. (1989) *Visuals for Information Research and Practice*. Englewood Cliffs, NJ: Educational Technology Publications.

Pimm, D. (1995) *Symbols and Meanings in School Mathematics*. New York: Routledge Falmer.

Preece, J., Rogers, Y. and Sharp, H. (2002). *Interaction Design: Beyond Human–Computer Interaction*. New York: John Wiley and Sons.

Raskin, J. (2000) *The Humane Interface: New Directions for Designing Interactive Systems*. Reading, MA: Addison-Wesley.

Roth, S., Chuah, M., Kerpedjiev, S., Kolojejchick, J. and Lucas, P. (1997). Towards an information visualization workspace: Combining multiple means of expression. *Human–Computer Interaction Journal* 12: 131–185.

Salomon, G. (1979) *Interaction of Media, Cognition, and Learning: An Exploration of How Symbolic Forms Cultivate Mental Skills and Affect Knowledge Acquisition*. San Francisco, CA: Jossey-Bass Publishers.

Scaife, M. and Rogers, Y. (1996). External cognition: How do graphical representations work?. *International Journal of Human–Computer Studies* 45: 185–213.

Scher, D. and Goldenberg, E.P. (2001). A multirepresentational journey through the law of cosines. In A.A. Cuocoand and F.R. Curcio (Eds), *The Roles of Representation in School Mathematics: 2001 Yearbook* (pp. 1–1). Reston, VA: National Council of Teachers of Mathematics.

Sedig, K. (2004). Need for a prescriptive taxonomy of interaction for mathematical cognitive tools. In Proceedings of Interactive Visualisation and Interaction Technologies, ICCS 2004 (pp. 1030–1037), June, Krakow, Poland.

Sedig, K., Klawe, M. and Westrom, M. (2001). Role of interface manipulation style and scaffolding on cognition and concept learning in learnware. *ACM Transactions on Computer–Human Interaction* 1(8): 34–59.

Sedig, K. and Morey, J. (2005). A descriptive framework for designing interaction for visual abstractions. In G. Malcolm (Ed.), *Multidisciplinary Approaches to Visual Representations and Interpretations* 239–254. Elsevier Science.

Sedig, K., Morey, J. and Chu, B. (2002). TileLand: A microworld for creating mathematical art. In Proceedings of ED-MEDIA 2002: World Conference on Educational Multimedia and Hypermedia, July, Denver, USA.

Sedig, K., Morey, J., Mercer, R. and Wilson, W. (2005). Visualizing, interacting and experimenting with lattices using a diagrammatic representation. In G. Malcolm (Ed.), *Multidisciplinary Approaches to Visual Representations and Interpretations* 255–268. Elsevier Science.

Sedig, K., Rowhani, S., Morey, J. and Liang, H. (2003). Application of information visualization techniques to the design of a mathematical mindtool: A usability study. *Information Visualization* 2(3): 142–160.

Shedroff, N. (1999). Information interaction design: A unified field theory of design. In R. Jacobson (Ed.), *Information Design* (pp. 267–292). Cambridge, MA: The MIT Press.

Sherman, W.R. and Craig, A.B. (2003). *Understanding Virtual Reality: Interface, Application, and Design*. San Francisco, CA: Morgan Kaufmann Publishers.

Shneiderman, B. (1988). We can design better user interfaces: A review of human–computer interaction styles. *Ergonomics* 3(15): 699–710.

Shneiderman, B. and Kang, H. (2000). Direct annotation: A drag-and-drop strategy for labeling photos. In Proceedings of the International Conference on Information Visualization (pp. 88–95), July. Los Alamitos, CA: IEEE Computer Society Press.

Sims, R. (1999). Interactivity on stage: Strategies for learner–designer communication. *Australian Journal of Educational Technology* 15(3): 257–272.

Sims, R. (2000). An interactive conundrum: Constructs of interactivity and learning theory. *Australian Journal of Educational Technology* 16(1): 45–57.

Spence, R. (1999). A framework for navigation. *International Journal of Human–Computer Studies* 51: 919–945.

Spence, R. (2001) *Information Visualization*. New York: Addison-Wesley.

Stojanov, G. and Stojanoski, K. (2001). Computer interfaces: From communication to mind-prosthesis metaphor. In M. Beynon, C.L. Nehaniv and K. Dautenhahn (Eds), *Cognitive Technology: Instruments of Mind – Proceedings of the International Cognitive Technology Conference* (pp. 301–310). Warwick, UK: Springer-Verlag.

Straesser, R. (2001). Cabri-géomètre: Does dynamic geometry software (DGS) change geometry and its teaching and learning?. *International Journal of Computers for Mathematical Learning* 6: 319–333.

Strothotte, T. (1998) *Computational Visualization: Graphics, Abstraction, and Interactivity*. Berlin: Springer-Verlag.

Stylianou, D. (2002). On the interaction of visualization and analysis: The negotiation of a visual representation in expert problem solving. *Journal of Mathematical Behavior* 21: 303–317.

Tall, D. and West, B. (1992). Graphic insight into mathematical concepts. In B. Cornu and A. Ralston (Eds), *The Influence of Computers and Informatics on Mathematics and its Teaching* (pp. 117–123). 2nd edition, Paris: UNESCO.

Thompson, S. and Riding, R. (1990). The effect of animated diagrams on the understanding of a mathematical demonstration in 11- to 14-year-old pupils. *British Journal of Educational Psychology* 60: 93–98.

Travaglini, S. (2003). An exploratory study of interaction design issues using a tiling microworld. Unpublished Master's thesis. Department of Computer Science, The University of Western Ontario, London, Ontario, Canada.

Tufte, E.R. (1997) *Visual Explanations*. Cheshire, CT: Graphics Press.

Tweedie L., Spence R., Dawkes H. and Su H. (1996). Externalizing abstract mathematical models. Proceedings of CHI'96 (pp. 406–412). New York: ACM Press.

Ware, C. (2000) *Information Visualization: Perception for Design*. San Francisco, CA: Morgan Kaufmann Publishers.

Webb, R. (2003). Stella: Polyhedron Navigator. *Symmetry: Culture and Science* 11(1–4): 231–268.

Weisstein, E.W. (2003) *CRC Concise Encyclopedia of Mathematics*. 2Boca Raton, FL: Chapman and Hall/CRC.

West, T.G. (1995). Forward into the past: A revival of old visual talents with computer visualization. *ACM SIGGRAPH Computer Graphics* 29(4): 14–19.

Winn, W. (1993). Perception principles. In M. Fleming and W.H. Levie (Eds), *Instructional Message Design: Principles from the Behavioral and Cognitive Sciences*, 2nd edition, Englewood Cliffs, NJ: Educational Technology Publications.

Yacci, M. (2000). Interactivity demystified: A structural definition for distance education and intelligent computer-based instruction. *Educational Technology* 40(4): 5–16.

Yerushalmy, M. and Shternberg, B. (2001). Charting a visual course to the concept of function. In A.A. Cuocoand and F.R. Curcio (Eds), *The Roles of Representation in School Mathematics: 2001 Yearbook*, Reston, VA: National Council of Teachers of Mathematics.

Zhang, J. (1997). The nature of external representations in problem solving. *Cognitive Science* 21(2): 179–217.

Cognitive Engineering Laboratory
The University of Western Ontario
Canada