# Event Recognition during the Exploration of Line-Based Graphics in Virtual Haptic Environments

Matthias Kerzel and Christopher Habel

Department of Informatics
University of Hamburg
D-22527 Hamburg, Germany
{kerzel,habel}@informatik.uni-hamburg.de

**Abstract.** Pictorial representations are widely used in human problem solving. For blind and visually impaired people, haptic interfaces can provide perceptual access to graphical representations. We propose line-based graphics as a type of graphics, which are suitable to be explored by blind and visually impaired people, and which can be successfully augmented with auditory assistance by speech or non-verbal sounds. The central prerequisite for realizing powerful assistive interaction is monitoring the users' haptic exploration and in particular the recognition of exploratory events. The representational layers of line-based graphics as well as of exploration-event descriptions are specified by qualitative spatial propositions. Based on these representations, event recognition is performed by rule-based processes.

**Keywords:** line-based graphics, virtual environment haptic representation, geometric diagrams, event recognition.

## 1 Line-Based Graphics

### 1.1 Pictorial Representations for Blind and Visually Impaired People

Pictorial representations play a major role in human cognition and especially in human problem solving [1]. We use maps for way-finding. We visualize data, i.e., we (re-)present data in a pictorial format more suitable for thinking, problem solving, and communication, namely in the representational modality of graphs. And we use diagrams of devices or machines for teaching how these artifacts work. For blind and visually impaired people, access to pictorial representations is limited. Therefore, haptic interfaces to graphical representations—named 'haptic representations' as well as 'tactile representations'—have been proposed for partially substituting vision in comprehending different types of pictorial representations, such as graphs, mathematical diagrams, maps, floor plans or line drawings of animals [2, 3, 4, 5, 6]. Whereas visual perception supports comprehension processes that switch quickly between global and local aspects of a pictorial representation, haptic perception is more local and in particular more sequential [7]. Thus, compared to visual representations, in exploring haptic representations users have to integrate distributed,

T. Tenbrink et al. (Eds.): COSIT 2013, LNCS 8116, pp. 109–128, 2013.

sequentially grasped local information over the course of exploration, i.e., over time, into a global ‘picture’ of pictorial representation.

Realizing haptic representations in virtual environments offers the opportunity of assistive interactions with the user. As haptic exploration without visual feedback is a difficult task, the user should not also be burdened with active control of the interaction, e.g., pressing keys or performing command gestures. Instead, the computer should act like an intelligent and proactive partner; see Beaudouin-Lafon [8] for the idea of computer-as-partner interaction. In agreement with this view, we propose an interaction situation in which concurrently (a) the user explores a pictorial representation via the haptic interface, and (b) the system observes, i.e., monitors the exploration process, in particular the user’s movements. The observations are related to an internal model of the system and, additionally, a propositional representation of the current exploration situation id build up. Based on these representations, the system can interact with the user: Additional information about entities that are relevant but not in the present focus of the user, as well as relations and properties of the depicted entities, are provided using speech or non-linguistic sound [5, 9, 10].

In order to foster the user’s active control over the interaction, future research might aim at integrating keyboard- or dialog-based information requests into the architecture. However, a certain degree of autonomy of the system is necessary to compensate the user’s missing overview, e.g., in some situations the user might not be aware that useful information could be requested.

### 1.2 Virtual Haptic Environments and Line-Based Graphics

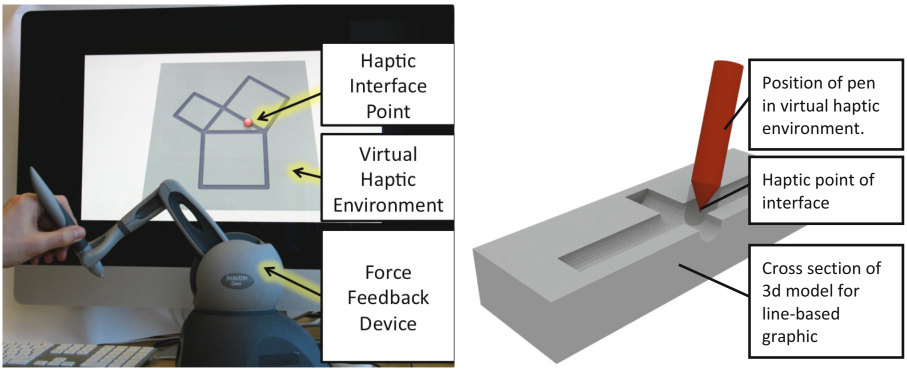
To increase the effectiveness of knowledge acquisition from pictorial representations provided by virtual haptic environments, we have developed prototypes of systems which provide access to different types of line-based graphics—such as maps, floor plans, graphs and diagrams—augmented with auditory assistance provided by speech or sonification [5, 9, 10, 11, 12]. In our approach virtual haptic perception is realized by using a Phantom Omni force feedback device (http://www.sensable.com), Fig. 1(a) depicts this device, as well as an exemplifying geometric diagram. To explore a virtual object haptically, a pen-like handle that is attached to a robot arm is used. The user’s pen movements are received by sensors in the robot arm and are evaluated with respect to the position of the tip of the pen. This position information is used to simulate a virtual pen tip, called haptic interface point (HIP), in a virtual environment. The force interaction between the HIP and the 3D model in the virtual environment is calculated as if the HIP and the 3D model were real objects, i.e., when the HIP collides with the 3D model, it is pushed back as if one would try to penetrate a solid surface with a real pen. This computed force feedback is made perceivable to the user through the motors in the robot arm that the pen is attached to. In short, the user feels as if probing a real 3D object with the tip of a real pen.

Virtual haptic exploration with a Phantom device can be characterized as *one-point interaction,* see Fig. 1(b). Thus, the exploration of an object is completely local and sequential. The haptic features of the object have to be explored sequentially and integrated, over time, into an internal model of the pictorial representation that should be consistent [13]. In order to design the interaction as user-friendly as possible, the haptic representation should consist of a small inventory of graphic atoms (see, for visual diagrams [14]) that are easy to explore haptically with the Phantom, and that additionally are cognitively adequate for constructing internal models.

As an alternative approach for haptic access to spatial information vibrotactile interfaces have been investigated. Giudice and colleagues [15] employ vibration to convey spatial information during exploring a touch screen using one fingertip. Like the force feedback realized by the Phantom, these vibrotactile approaches are based on one-point interaction. Thus, the formalism for event recognition, which is presented in sect. 3 and 4, could be extended to these vibrotactile interfaces.

Spatial information for several application domains of virtual haptics like street maps, public transit maps, geometric diagrams, graphs and charts can be based on *lines*. In this view, regions can be seen as second-order entities that are specified by their boundary. For example, a triangle (closed plane figure) can be characterized as well as it is depicted by three straight lines meeting at three vertices, see Fig. 1(a). Therefore, we propose to use line-based graphics as the conceptual basis for representing spatial information in virtual haptic environments. A line-based graphic consists of lines in a plane. These lines may be combined to form complex arrangements and shapes.

Lines are realized in a virtual haptic environment by using a planar 3D model with u-shaped grooves. Fig. 1(b) shows a cross section of such a model with a virtual HIP. On a conceptual level line-based graphics are used as planar spatial representations. The u-shaped grooves guide the user's exploration along a linear trajectory: Haptic groove-following leads to line-following on the conceptual layer and results in line recognition.




(a) (b)

**Fig. 1.** (a) Sensable Phantom Omni and visualization in the domain of geometry (b) Cross section of 3D model used to represent a line-based graphic

### 1.3 Event Recognition for Exploration Monitoring

As described in sect. 1.2, a crucial deficit of haptic representations compared with visual representations is determined by the sequential and local nature of haptic exploration. For example, when looking at an arrangement of geometric figures as depicted in Fig. 1(a), an observer will immediately be able to identify, distinguish and locate the individual figures. In haptic explorations each figure, and even each part of a figure, has to be explored and identified separately. Thus, at the start of a haptic exploration a line is perceived as a line, and not, for example, as side of a triangle.

To help users overcome this type of problems, the system can provide additional information. Habel, Kerzel and Lohmann [12] suggest event recognition as an interaction paradigm for the generation of verbal assistance during the exploration of virtual environment tactile maps. Based on principles of division of labor, haptic and audio modality can be combined [5]. Verbal assistance provided during haptic exploration can offer non-spatial information that would be given as textual annotation in a visual representation as well as spatial information that is difficult or time consuming to extract from the haptic representation.

Parkas [16], Moustakas et al. [17] and Wang et al. [18] describe touch-based approaches that realize verbal assistance during exploration of tactile presentations. In their systems, touching (using a stylus/pen device or a finger on a touch screen) certain parts of a haptic representation immediately triggers a predefined assistance. This, however, limits the set of possible assistances by not taking into account the history of exploration, thus limiting the interactive capabilities of these systems.

Especially in haptic representation, where several elements overlap or have direct contact to each other, identifying which element is currently explored and should be in the focus of assistance, requires the recognition of *exploration events* (pen movements during exploration). For example, a triangle can be explored haptically by following the three triangle-constituting lines. Without going into details here (but see sect. 3.2), exploration events include both ongoing processes (called extended events) as well as short momentary transitions (called non-extended events). Recognition of extended exploration events takes the history of the exploration into account. Extended events take place over a time interval and often have a complex structure, being composed of several smaller events (see sect. 3).

Thus, event recognition provides the possibility to assist the user in different kinds of situations. The event recognition component has access to the full spatial representation and can interpret exploration movements in this context. The user, on the other hand, only has a local perception and lacks an overview of the explored representation. In haptic perception of space, distances and sizes are perceived differently depending on position and direction of the haptic exploratory movements [19]. This complicates identifying and understanding structures during the exploration, but assistance based on event recognition can help the user to overcome these difficulties.

In the following, we exemplify this conception of event recognition with some exploration events, both extended as well as non-extended ones, during the exploration of a geometric figure illustrating the Pythagorean theorem; see Fig. 2(a).

The Pythagorean theorem is chosen for exemplifying haptic exploration of line-based graphics in an abstract domain. Moreover, we conducted empirical studies of haptic exploration of line based-graphics depicting more complex spatial arrangements that were novel to the participants, see Lohmann, Kerzel and Habel [10] for an example of haptic exploration of line-based graphics from the domain of maps.

The pictorial representation of the Pythagorean theorem, consisting of a triangle and three squares, illustrates the fact that the area of the square over the hypotenuse of a right-angled triangle is equal to the sum of the area of the two squares over the other two sides. In a haptic environment based on line-based graphics, which could for example be used in teaching geometry to blind students, the lines of the figures are realized as grooves. The user explores the outlines of the figures by following these grooves in order to judge shape and size of the explored geometric entities. Fig. 2(b) shows an exemplifying exploration: (1) a side is explored, which is the hypotenuse of the right-angled triangle, but is also a side of a square, namely, the square over this hypotenuse. Thus, this exploration movement can—in future exploration—lead to two extended exploration events. (2) The next exploration movement follows another side of the triangle, which is again also a side of a different square. So the process of exploring square A has ended, while the exploration of the triangle is still going on. (3) Finally, the third exploration movement ends where the exploration started, the third side of the triangle is explored, thus completing the process of exploring the triangle. This fact could be communicated to the user, who might not be aware that the exploration ended in a closed loop.

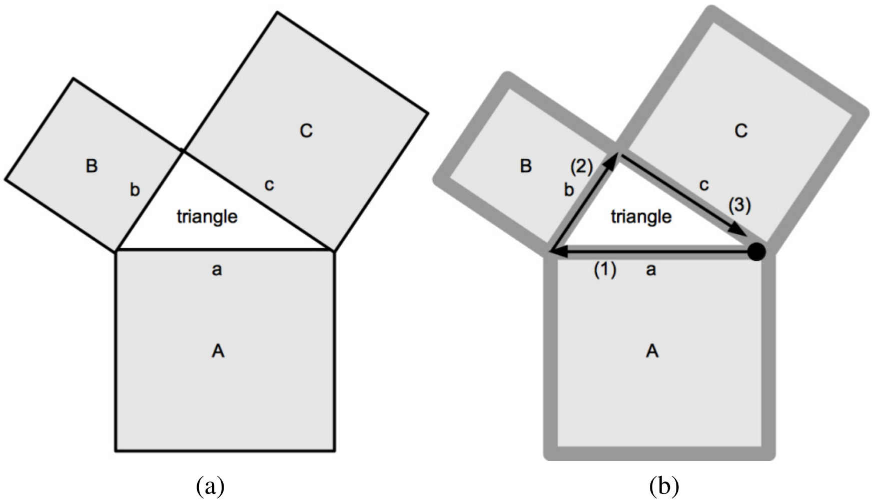



**Fig. 2.** (a) Visual representation for Pythagorean theorem, as used for teaching Geometry. (b) Visualization of haptic realization for Pythagorean theorem, lines are grooves (darker grey), the black arrows indicate exploration trajectories mentioned in the text

This example shows both extended events, like the process of exploring the side of a square or the triangle, and non-extended events, like completing the exploration of the triangle by arriving at the start point of the exploration. Exploration events are not unique with regard to their descriptions. The first exploration event depicted in Fig. 2(b) can be described both as ‘exploration of the hypotenuse of a triangle’ and as ‘exploration of the upper side of a square’. The exploration can also be described in terms of complex events. The exploration of the triangular shape, i.e., the sequence of exploring the three sides, coincides with the exploration of the sides of three different squares. In such cases, the time intervals during which the extended events take place overlap. This exemplifies that not only the internal structure of exploration events may be complex, but also the temporal relations among recognized events may form a variety of temporal relations, see [20] for a representation of qualitative temporal relations. When reacting to the user’s exploration, these temporal relations need to be captured during the process of event recognition.

## 2 Propositional Representations for Spatial Information in Virtual Haptic Environments

Line-based graphics can be based on an inventory of geometric concepts that can be used for qualitative spatial descriptions and reasoning. These propositional qualitative representations are the basis for both representing exploration movements in virtual haptic environments (sect. 3) and a rule-based approach to recognize events (sect. 4). In this section we give a rough overview on the basic concepts and on their use for specifying line-based graphics. (The underlying geometrical concepts belong to incidence geometry and ordering geometry, see [21, 22, 23].)

**Lines and Configuration Points.** *Lines* in line-based graphics are specified as maximal straight linear entities, i.e., lines are not part of other lines, but lines may have linear parts, namely *segments* (see below). Points on different lines may coincide; these points are called *configuration points,* as they are determined by configurations of lines. The *endpoints* of lines also are configuration points: They are defined as points on the line, which do not lie between two other points on the line [22]. Configuration points can be differentiated as a *junction*, a *tee* or a *crossing* of two lines; these subtypes depend on cardinality- and shape properties of the determining line configurations (see [24, 25] for kindred approaches).

**C-Segments and Planar Embedding.** Each line may be decomposed by a set of configuration segments. A *configuration segment* (c-segment) is a part of a line that lies between two adjacent configuration points on that line. The segmentation predicate describes which c-segments belong to a line. A line that contains no tee or crossing configuration point consists of only one c-segment.

Sets of c-segments may be considered to be sets of edges of a planar graph. Consequently, the nodes of the planar graph may be seen as counterparts of configuration points. Configuration points are embedded in the plane by defining their position in absolute coordinates. Thus, a line-based graphic equals an embedded planar graph.

**Other Segmentation Criteria.** The approach to insert segmentations into lines can be extended to other criteria depending on the application domain. A line segment may for example represent the part of a line determined by the projection of another entity onto the segmented line. In the application domain of street maps this is used to represent the part of a street that is next to a landmark. Segments defined by projections are called *p-segments*.

**Line Complexes.** Line complexes can be formed by a set of transitively connected c-segments. A c-segment may belong to multiple line complexes. All configuration points of c-segments also belong to the line complex(es) that the c-segment belongs to. This model follows an intuitive understanding of typical application domains of virtual haptic representations: When two streets intersect, each street can be said to have a crossing with another street, the crossing does not belong exclusively to just one of the streets. The same applies to two lines in a graph chart.

Depending on the configuration of c-segments, a line complex may have several properties: A line complex is branching if there exists a configuration point that is shared by more than two c-segments. A line complex is closed if the c-segments in the line complex form a closed path (according to graph theory). A non-branching closed line complex defines a region that is bounded by the c-segments. Thus, line complexes may form second order elements of line-based graphics.

**Domain-Specific Properties.** Each structure in a line-based graphic may also have properties relevant to the application domain. For example, the proper name in the context of the application domain is used as a predicate. This approach can easily be extended to provide other relevant information about the element in the domain.

**Table 1.** Subset of propositions defining elements of line-based graphics

| **Propositional description** | **Verbal description** |
|---|---|
| line($l$) | Element $l$ is a straight line on a plane. |
| endPoint($l$, $p$) | Point $p$ is an endpoint of line $l$. |
| configurationPoint($c$) | Configuration point $c$ is a point on a line, $c$ can either be an endpoint of a line or a point where two lines meet. |
| located($c$, $x$, $y$) | Configuration point $c$ is embedded in a plane at the coordinates ($x$, $y$). |
| c-segmentation($seg$, $l$, $\{s_1, .., s_n\}$) | Line $l$ is segmented by segmentation $seg$, dividing $l$ into the set of c-segments $\{s_1, .., s_n\}$. |
| c-segment($s$, $seg$, $c_1$, $c_2$) | C-segment $s$ is created by segmentation $seg$ due to configuration points. C-segment $s$ is the part of a line that lies between configuration point $c_1$ and configuration point $c_2$. |
| lineComplex($lc$, $\{s_1, .., s_n\}$) | Line complex $lc$ is formed by the set of transitively connected c-segments $\{s_1, .., s_n\}$. |
| domainName($e$, $name$) | Element $e$ has is called $name$ in the domain. |

# 3 Exploration Events Based on Propositional Spatial Knowledge

## 3.1 Exploration Events in Line-Based Graphics

According to Lederman and Klatzky, manual exploration plays the core role for haptic perception [19]. In the case of line-based graphics in virtual haptic environments the manual exploration is mediated through a haptic interface point (HIP) that the user moves within grooves provided by a virtual haptic model. Thus, exploration movements are mainly restricted to movements along linear structures similar to movements of vehicles within a network of roads; see Pfoser and Jensen [26]. (Since the users of our prototypical line-based-graphics systems are instructed to explore a graphic based on lines, only in very few cases a user left the grooves and started exploratory movements in regions outside the grooves; compare Yu and Habel [5] for a similar process of border-following during the haptic exploration of regions depicting rooms.)

Thus, the exploration events to be recognized by the system are movements along the lines of line-based graphics. Storing of and reasoning about observed events presupposes a systematic notation for representations of events. These event representations are the basis for user interaction. They describe what is currently happening in the exploration or what has just happened. The event representation also serves as a history of the exploration, which is utilized to recognize more complex exploration patterns.

According to Shipley and Maguire [27] an observed sequence of movements may have an event representation on different levels of granularity. During a haptic exploration, a recognized exploration event may be the tracing of a single geometric figure. But this exploration also consists of tracing each single edge of the figure. Thus, the observed exploration can also be represented as a sequence of finer grained exploration events. The presented approach does not try to solve the problem of granularity by defining an optimal level of granularity. Instead, a *hierarchical event representation* is used where events are parts of more complex events. Which events are used to generate assistance, depends on the importance of the explored elements in the line-based graphic in the context of the application domain and will not be discussed in this article.

As an example, completely exploring a line complex having the shape of a triangle requires exploring all edges of the triangle, i.e., completely exploring all c-segments that are part of the line complex forming this triangle. In turn, completely exploring a c-segment requires an exploration movement from one configuration point of the c-segment to the configuration point at the other end of the c-segment. This movement is then in turn broken down into non-extended events that describe position changes during the exploration, like leaving or arriving at a configuration point. Each recognized event in this hierarchy offers an opportunity to assist the user during the exploration.

### 3.2 Types of Exploration Events

Exploration events are distinguished by temporal duration: *Non-extended events* answer the question “What has just happened?” while *extended events* answer the question “What is currently happening?” Non-extended events take place in a single moment. Extended events on the other hand describe exploration processes. Fig. 3 shows a taxonomy of the different event types developed for the representation of haptic exploration of line-based graphics. This taxonomy of exploration events is not exhaustive, i.e., a user may perform exploration movements that have not been classified. Yet the proposed set of exploration events enables the assisting system to react adequately to the user’s current exploration movements by focusing assistance on currently explored elements of the line-based graphic.

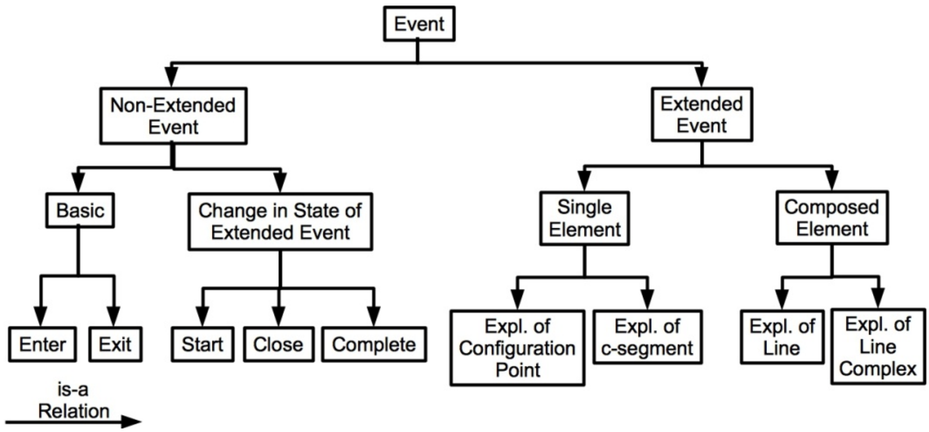
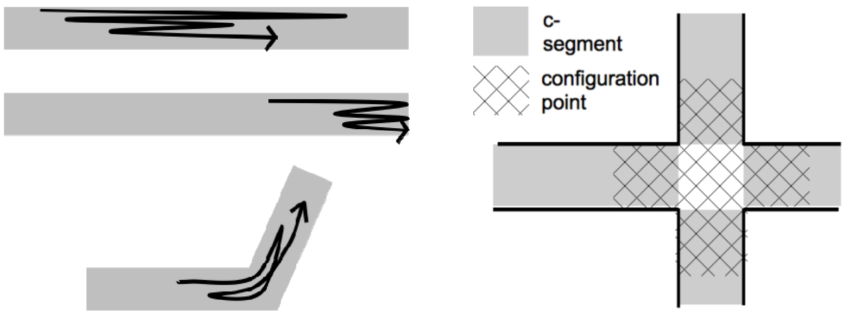



**Fig. 3.** Taxonomy of exploration events concerning line-based haptic graphics (Expl. stands for exploration)

**Basic Non-extended Events.** *Basic non-extended events* can be directly observed from the sensor data provided by the haptic device. The detection of basic non-extended events represents the transition from continuous perception to discrete events related to the geometry of a line-based graphic. For each c-segment and configuration point, a space in the virtual haptic environment is defined. Although a point has no area on the conceptual level, configuration points are perceived through active movement in the haptic environment, i.e., repeated movements against the end of a line or repeated tracing of the corner between two lines. Fig. 4(a) shows typical exploration movements for lines and configuration points. Basic non-extended events describe *entering* or *exiting* non-composed elements of the line-based graphics, i.e., c-segments (or other types of segments) and configuration points.

**Extended Events: C-Segment and Configuration Point Exploration.** *C-segment* and *configuration point exploration* events describe the process of exploring a single element in a line-based graphic. Exploration movements in c-segments often go back and forth. These repeated exploration movements allow the explorer to perceive the angle at which the c-segment is oriented. Likewise, configuration points are explored through movements in its vicinity. The event representation abstracts these movements into a single exploration event.

The spaces associated with configuration points and c-segments overlap. Thus, while exploring a c-segment, the configuration points at its ends can be explored without leaving the c-segment, and vice versa the adjoining part of c-segments can be explored while not leaving a configuration point. Fig. 4(b) shows a 2D sketch of the spaces of c-segments and a configuration point.




(a) (b)

**Fig. 4.** (a) Typical exploration movements (depicted by black arrows) within a line segment, at the end of a line segment, and at a configuration point (2-way junction) (b) Overlapping areas associated with c-segments and a configuration point at a crossing

**Extended Events: Line and Line-Complex Exploration.** *Line* and *line complex explorations* describe the exploration of more complex elements in line-based graphics that consist of c-segments. The user might not sequentially explore the c-segments but move back and forth between c-segments to gain a better understanding of the spatial layout. As long as the explored element is not left during the exploration, repetitions are abstracted in the event representation.

**Non-basic Non-extended Events: Start, Closure, Completion and Subordination.** *Non-basic non-extended events* describe changes to ongoing, extended events. In contrast to basic non-extended exploration events, they are not directly observable. Arriving at a configuration point can be directly observed, but realizing that arriving at this configuration point completes exploring a side of a triangle cannot. Recognizing this event relies on knowledge about the history of the exploration.

The recognition of an exploration process causes the *start* of a corresponding exploration event. Likewise, the end of an extended event represents the moment when the extended event is irrevocably over. For instance, the user might start to explore a triangle, but before finishing it, he moves on to explore a different part of the line-based graphic. This is called *closure* as the exploration event is kept in the exploration history, but can no longer be influenced by the current exploration.

If the user later starts to explore the triangle again, this will be modeled as a new exploration event, unrelated to the former exploration event.

Extended exploration events are recognized before they are *completed*. This is useful for the purpose of assistance generation. The user might be interested in knowing that a triangle is explored before the complete contour of the triangle is traced. The completion of an element is recognized separately; this can be used to inform the user that a complete element has been explored and nothing was missed.

Events can be integrated into the hierarchical structure of an open extended event. This is called *subordination*. When an event $e_1$ is subordinated under an event $e_2$, this can be interpreted as $e_2$ growing not only in a temporal, but also in a conceptual way. For instance, the user explored the first side of a triangle. The extended event of exploring the triangle is started. When the user moves to the next side of the triangle, the exploration of this side is subordinated under the exploration of the triangle, thus the exploration event of the triangle now consist of two exploration events of its sides.

### 3.3 Propositional Representation for Exploration Events

Events are represented using propositions. For each exploration event, propositions are used to specify the type (see, for the taxonomy of exploration events, Fig. 3), the unique name and other properties.

**Hierarchical Properties: List of Subordinated Events.** Events are recognized by hierarchically aggregating events into more complex events. Basic non-extended events are used to conceptualize extended events describing the exploration of single elements in a line-based graphic; these extended events are in turn used to conceptualize events describing explorations of composed elements. When an event is aggregated from simpler events, these simpler events become subordinated to the more complex event. In the propositional representation each extended event features a list of events subordinated to it.

**Temporal Properties: Time Stamp, Qualitative Temporal Relations and Closure.** Basic non-extended events have a pseudo timestamp, representing the time point they were observed. By recursively accessing the information about the hierarchical aggregation down to the level of basic non-extended events, temporal information about any event can be computed and compared in terms of qualitative temporal relations. An extended event that is ongoing has the property *open*.

**Progress Property: Completion.** An extended event is completed when its subordinated events form a defined pattern. For instance, the exploration of a c-segment is completed when both configuration points at its ends have been explored without leaving the c-segment. In that case, the exploration events of both configuration points would be subordinated to the exploration of the c-segment. A completed event has the property *completed*.

**Interaction of Closure and Completion.** *Open* and *completed* are independent properties of an extended event. An extended event might be both *open* and not *completed*, i.e., the user might be exploring a triangle, but might not yet have finished exploring all parts of the triangle. If the user quits exploring the triangle before all parts have been explored, the exploration event becomes closed, while not being *completed*. The user might also have completed exploring all parts of the triangle, but still continue to trace its contour to get a better understanding of its spatial layout, the exploration of the triangle thus being complete and open. Finally, after the complete exploration of the triangle, the user might continue the exploration at a different figure; the exploration of the triangle thus being still *complete* but no longer *open*.

**Event Representations.** For each event the criteria for start, closure and completion are listed. The term *e* is used as a unique name to identify the event. Using the properties *open* and *complete* the state of an extended event is specified. The list [$e_1$, .., $e_n$] contains names of all events that are subordinated to event *e*.

1) The event *e* is the basic non-extended event of entering or exiting the zone associated with c-segment or configuration point *c*. Event *e* takes place at time point *t*.

enter(*e*, *c*, *t*), exit(*e*, *c*, *t*)

2) The extended event *e* describes the exploration of a c-segment *s*. Event *e* starts when an enter event for *s* is detected. Event *e* is completed once both configuration points at the ends of the c-segment have been explored, that is, exploration events for both configuration points are contained in [$e_1$, .., $e_n$]. During *e*, no element of the line-based graphic that is not *s* or one of its configuration points may be entered, otherwise *e* is closed.

cSegmentEx(*e*, *s*, *open*, *completed*, [$e_1$, .., $e_n$])

3) The extended event *e* describes the exploration of a configuration point *c*. The event starts when an enter event for *c* is detected. The event is completed once [$e_1$, .., $e_n$] contains explorations of all c-segments meeting in *c*. During *e* the exploration may not exit *c*, otherwise *e* is closed.

cpEx(*e*, *c*, *open*, *completed*, [$e_1$, .., $e_n$])

4) The extended event *e* describes the exploration of a line *l*. The event starts, when a c-segment belonging to *l* is explored. The event is completed once [$e_1$, .., $e_n$] contains completed explorations of all c-segments in *l*. During *e* no element that is not part of *l* may be explored, otherwise *e* is closed.

lineEx(*e*, *l*, *open*, *completed*, [$e_1$, .., $e_n$])

5) The extended event *e* describes the exploration of a line complex *lc*. The event is completed once [$e_1$, .., $e_n$] contains completed exploration events of all c-segments in line complex *lc*. During *e* no element that is not part of *lc* may be explored, otherwise

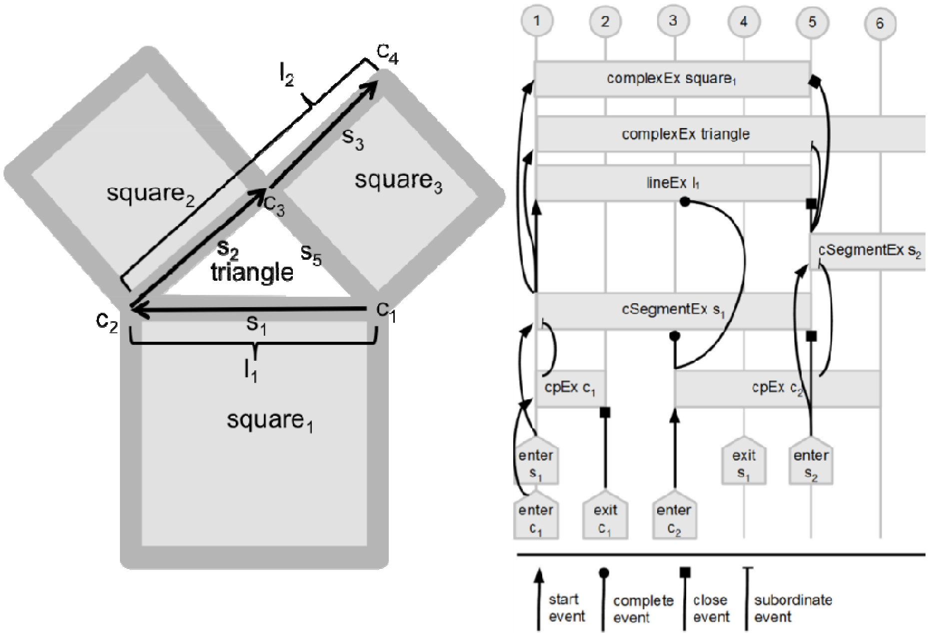
*e* is closed. The list of events [$e_1$, .., $e_n$] lists all line- and c-segment exploration events that are part of *e*.

ComplexEx(*e*, *lc*, *open*, *completed*, [$e_1$, .., $e_n$])

### 3.4 Example: Exploration Events during Exploration

Fig. 5(a) shows a line-based graphic illustrating the Pythagorean theorem; the black arrows indicate the exploration trajectory of the user. The following 11 steps explain how the event representation is updated, each step corresponding to an enter event or to an exit event with regard to a c-segment or a configuration point.

(Step 1) The exploration movement starts in the lower right corner of the triangle. The start position of the exploration movement is both within the space of configuration point $c_1$ (the lower left corner of the triangle) and c-segment $s_1$, the hypotenuse of the triangle. In this step, $s_1$ and $c_1$ are entered. Therefore, corresponding exploration events for $c_1$ and $s_1$ are ongoing. As c-segment $c_1$ is a common part of line $l_1$, line complex *triangle* and line complex *square*$_1$, the exploration of c-segment $s_1$ contributes to all three exploration events of these elements which begin with this step. (Step 2) The black arrow in the hypotenuse shows the first exploration movement. The hypotenuse is traversed from right to left. First the vicinity of the lower left corner (configuration point $c_1$) is exited. The corresponding exploration event of $c_1$ is closed. It has not been completed, as not all c-segments that meet in $c_1$ have already been explored. From the incomplete exploration of $c_1$ it can be deduced that the user is not aware of how this configuration point is connected to other lines and therefore he should be given suitable assistance. (Step 3) At the end of the first exploration movement, the lower left corner of the triangle is reached, configuration point $c_2$ is entered. Reaching this point completes the exploration of c-segment $s_1$. As $s_1$ is the only c-segment belonging to line $l_1$, the hypotenuse of the triangle, the exploration of this line is also completed. (Steps 4 to 6) In the second exploration movement, the user traces the left side of the triangle, which is an exploration of c-segment $s_2$. C-segment $s_1$ is exited and c-segment $s_2$ is entered thereafter. Up to this point, the explorations of line complexes *triangle* and *square*$_1$ were ongoing. As c-segment $s_2$ is part of line complex *triangle* and *square*$_2$, but not of *square*$_1$, the exploration of line complex *triangle* continues, exploration of *square*$_1$ ends, and a new exploration event for *square*$_2$ begins. C-segment $s_2$ is also part of line $l_2$, therefore a corresponding exploration event begins. When the exploration movement exits the vicinity of configuration point $c_2$, the exploration of $c_2$ ends without having been completed. (Step 7) Configuration point $c_3$, the top of the triangle, is entered. This completes the exploration of c-segment $s_2$, the left side of the triangle. (Steps 8 to 10) The third exploration movement continues in a straight line along one side of *square*$_3$. C-segment $s_2$ is exited and $s_3$ is entered. C-segment $s_3$ is part of line complex *square*$_3$, but not of *triangle*. The exploration of *triangle* ends and a new exploration event of *square*$_3$ begins. The user is still moving along line $l_2$, the corresponding exploration event still goes on. As the user continues the exploration in a straight line, it can be deduced that the user might not be aware that the exploration of *triangle* has stopped. This fact could be the basis for assistance to the user. (Step 11) Finally, the exploration movement reaches the topmost corner of *square*$_3$, configuration point $c_4$. This completes the exploration of c-segment $s_3$. As line $l_2$ consists of the c-segments $s_2$ and $s_3$ and both have been completely explored without leaving line $l_2$, the exploration of line $l_2$ is also completed. Fig. 5(b) shows selected events created and modified during the exploration process.




(a) (b)

**Fig. 5.** (a) Visual depiction of a haptic exploration trajectory (depicted by black arrows) of the Pythagorean theorem diagram (b) Selected events that are created and modified during the first steps of the exploration

## 4 Rule-Based Event Recognition

Section 2 and 3 dealt with the representation of spatial information and exploration events. This section will discuss the procedural aspects of event recognition using a rule-based formalism.

### 4.1 Complex Event Processing

Luckham [28] coined the term *complex event processing* (CEP), to describe the process of recognizing meaningful events from a continuous stream of less complex events in a timely manner. Event recognition during the exploration of virtual haptic line-based graphics is a case of complex event processing. While the stream of position information from the haptic device is monitored for basic non-extended events, these non-extended events are processed in order to recognize more complex, extended exploration events. The computational challenge is to identify complex event patterns in a fast changing knowledge base of events that is subject to unpredictable input.

Because of the ability to perform well in fast changing environments, *rule-based systems* have been established as a paradigm for complex event processing; see Eckert and Bry [29]. Rule-based systems are used in artificial intelligence to codify human problem solving skills as a set of situation-action rules, see Hayes-Roth [30]. In [31], Anderson shows how complex cognitive abilities can be achieved by rule-based systems that not only encompass solving predefined problems, but also reaction to unpredictable input. In event recognition, during the exploration of virtual haptic environments, this unpredictable input is the user's exploration movement, to which the system that monitors the exploration must adapt quickly.

Rule-based systems realize their abilities through interaction of *declarative knowledge* (propositions in a knowledge base) and *procedural knowledge* (a set of rules). The knowledge base contains both static and dynamic information. In the case of event recognition during the exploration of line-based graphics, the spatial information is static. The dynamic part of the knowledge base consists of the representation of events. The procedural knowledge defines how the dynamic part of the knowledge base is altered. These rules consist of preconditions and actions that modify the declarative knowledge, once the preconditions of a rule are fulfilled.

The *inference engine* of a rule-based system constantly monitors the knowledge base, evaluates the preconditions of rules and executes the actions specified by the rules. This process can in return modify the knowledge base, triggering recursive execution of other rules. Forgy [32] developed the Rete algorithm that is based on arranging rule preconditions in a network to handle matching of preconditions against a fast changing knowledge base with a manageable time complexity. Walzer at al. [33] show how the Rete algorithm can be extended to allow for qualitative temporal relations in rule conditions, which are often used in event recognition.

### 4.2 Architecture for Event Recognition during the Exploration of Line-Based Graphics

The rule-based event recognition is embedded into an architecture for event recognition and assistance generation, see Fig. 6. The architecture of this system is based on an architecture suggested by Habel, Kerzel and Lohmann [12]. While the user interacts with the haptic interface, the virtual haptic environment provides force feedback based on 3D spatial information (i.e., on a 3D model of the line-based graphic). Exploration movements of the user are sent to a component for *basic non-extended event detection*, where the current position of the exploration is related to spatial information of the line-based graphic. Basic non-extended events are added to the knowledge base upon detection. The *inference engine* of the rule-based system constantly monitors the knowledge base taking into account both static spatial information and dynamic information about exploration events. If the precondition of a rule is fulfilled, the rule is executed, creating new exploration events, or modifying existing ones. Based on recognized exploration events, (verbal) assistance is generated and passed to the user.

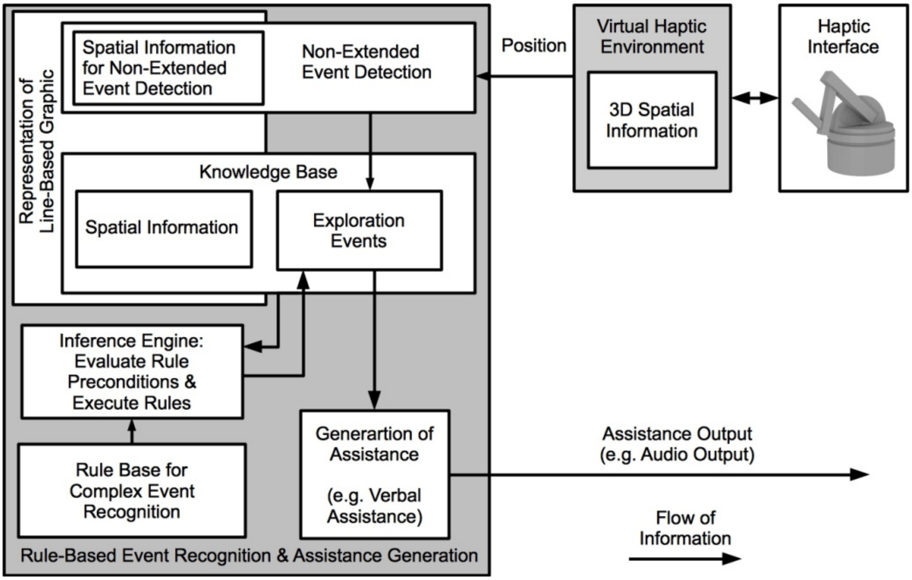



**Fig. 6.** Architecture for rule-based event recognition and assistance generation

### 4.3 Rules for Event Recognition

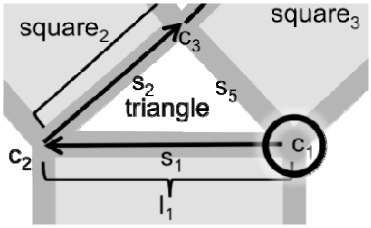
The complex events introduced in sect. 3.3 can be described by formal patterns using first order predicate logic as preconditions of rules. For example, completing the exploration of a line complex requires two main conditions: For all c-segments of the line complex an exploration event has to be completed; and all these exploration events have to happen without the exploration being interrupted by leaving the line complex. However, the user is allowed to explore the line complex in any order and perform repetitions during the exploration as long as the first two criteria are fulfilled.

For each type of extended event, a separate rule specifies when an event starts, ends, is completed (if appropriate) and when another event is subordinated to it. In figure 7 and 8, two rules out of a set of about 20 rules are shown [34].

The conditions of rules are given as a conjunction of propositions. In order for the rule to be executed, all conditions of the rule have to evaluate to *true* according to the knowledge base. Several functions are used in these rules: id() creates a new and unique identifier for an event; in($l$, $e$) evaluates to *true*, if event $e$ is contained in the list of events $l$ and append($l$, $e$) adds event $e$ to the end of list $l$. Temporal relations are expressed in qualitative terms. The underscore (_) is used to denote that any value is acceptable or that no change is made to a value.

IF
  enter($e_{enter}$, $s$, $time$)
  $\neg\exists\ e_{cp}$: cSegmentEx($e_{cp}$, $cp$, _, _, $list$)
  in($e_{enter}$, $list$)
THEN
  Create:
  cSegmentEx($id()$, $s$, $t$, $f$, [$e$])

(a)




(b)

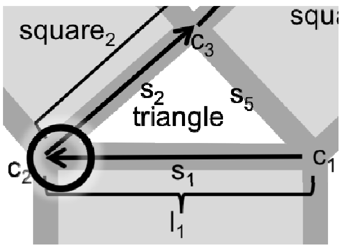
**Fig. 7.** (a) Rule for recognizing the start of an exploration event of a c-segment (b) Starting situation of exemplifying exploration (sect. 3.4)

Fig. 7(b) depicts the starting situation of the exploration from sect. 3.4. C-segment $s_1$, the hypotenuse, is entered. This creates a corresponding entering event. There is no c-segment exploration event in the knowledge base that is based on this entering event. Therefore, an exploration event of $s_1$, which is currently ongoing and not completed, is added to the knowledge base by the rule in Fig. 7(a). This falsifies the second condition of the rule, thus preventing the rule to be executed repeatedly. At the same time, configuration point $c_1$, the lower right corner of the triangle, is entered. A different rule creates a corresponding exploration event of $c_1$ in the knowledge base.

The rule for subordinating the ongoing exploration of a configuration point, in this case the exploration of $c_1$, causes this exploration event to be subordinated to the ongoing exploration of c-segment $s_1$. Spatial knowledge from the propositional representation of the line-based graphic is used to check whether the configuration point is actually one of the endpoints of the c-segment, which holds true in this case.

IF
  cSegmentEx($e_{seg}$, $s$, $t$, $f$, $list$)
  c-segment($s$, $seg$, $cp_1$, _)
  cpEx($e_{cp1}$, $cp_1$, _, _, _)
  in($e_{cp1}$, $list$)
  c-segment($s$, $seg$, _, $cp_2$)
  cpEx($e_{cp2}$, $cp_2$, _, _, _)
  in($e_{cp2}$, $list$)
THEN
  Modify:
  cSegmentEx($e_{seg}$, _, _, $t$, _)

(a)




(b)

**Fig. 8.** (a) Rule for recognizing the completion of an exploration event of a c-segment (b) Completion of the first exploration movement along c-segment $s_1$ (see sect. 3.4)

Fig. 8 (b) depicts the situation after the first exploration movement from the example in sect. 3.4. has been completed. The exploration movement has just reached configuration point $c_2$ in the lower left corner of the triangle. An exploration event for the configuration point is added to the knowledge base and subordinated to the ongoing exploration of the traversed c-segment, i.e., it is added to the list of events being subordinated to the exploration of c-segment $s_1$.

Three exploration events have already been subordinated to the exploration event of $s_1$, the first is entering $s_1$, the second is the exploration event of configuration point $c_1$ where the exploration started; the third is an exploration event for configuration point $c_2$ which has just been reached. Thus, the preconditions stated in the rule in Fig. 8(a) are fulfilled; the configuration points on both ends of $s_1$ have been explored now. The state of the exploration event of $s_1$ is changes to *completed*.

## 5 Conclusion

Line-based graphics have been introduced as a formal basis for conveying spatial information from different application domains, such as maps, graphs, charts or mathematical diagrams, to blind or visually impaired people through virtual haptic environments. The specification of line-based graphics follows the criteria outlined by Mackworth [35]. This offers the advantages that, firstly, design principles for haptic line-based graphics can be applied to a variety of application domains and that, secondly, users familiar with haptic line-based graphics in one application domain can successfully use their exploration experience in a new domain.

To assist blind or visually impaired people in their exploration of pictorial representations, assistive communication using speech or non-verbal sound has been proved to be helpful. Powerful assistance not only reacts locally to the current point of interaction, but also considers the context of exploration as well as the history of the exploration process. As a core component for such assistance system, we have proposed an event monitor, which recognizes exploration events using rule-based mechanisms. A system prototype, which uses a prior version of the event-recognition component for *Verbally Assisted Virtual-Environment Tactile Maps*, has been successfully evaluated by Lohmann, Kerzel and Habel [10] in the domain of street maps.

**Acknowledgement.** We like to thank the anonymous reviewers for their thoughtful comments and valuable suggestions. The presented research has been partially supported by DFG (German Science Foundation) in the international research training group ‘Cross-modal Interaction in Natural and Artificial Cognitive Systems’ (CINACS, IRTG 1247).

## References

1. Tversky, B.: Visualizing Thought. Topics in Cognitive Science 3, 499–535 (2011)
2. De Felice, F., Renna, F., Attolico, G., Distante, A.: A Haptic/Acoustic Application to Allow Blind the Access to Spatial Information. In: World Haptics Conference, pp. 310–315 (2007)
3. Shimomura, Y., Hvannberg, E., Hafsteinsson, H.: Haptic cues as a utility to perceive and recognise geometry. Universal Access in the Information Society Online First, doi: 10.1007/s10209-012-0271-2

4. Sjöström, C., Danielsson, H., Magnusson, C., Rassmus-Gröhn, K.: Phantom-based Haptic Line Graphics for Blind Persons. Visual Impairment Research 5, 13–32 (2003)
5. Yu, J., Habel, C.: A Haptic-Audio Interface for Acquiring Spatial Knowledge about Apartments. In: Magnusson, C., Szymczak, D., Brewster, S. (eds.) HAID 2012. LNCS, vol. 7468, pp. 21–30. Springer, Heidelberg (2012)
6. Yu, W., Brewster, S.A.: Evaluation of Multimodal Graphs for Blind People. Journal of Universal Access in the Information Society 2, 105–124 (2003)
7. Loomis, J., Klatzky, R., Lederman, S.: Similarity of Tactual and Visual Picture Recognition with Limited Field of View. Perception 20, 167–177 (1991)
8. Beaudouin-Lafon, M.: Designing Interaction, not Interfaces. In: Proceedings of the Working Conference on Advanced Visual Interfaces, pp. 15–22. ACM (2004)
9. Lohmann, K., Habel, C.: Extended Verbal Assistance Facilitates Knowledge Acquisition of Virtual Tactile Maps. In: Stachniss, C., Schill, K., Uttal, D. (eds.) Spatial Cognition 2012. LNCS, vol. 7463, pp. 299–318. Springer, Heidelberg (2012)
10. Lohmann, K., Kerzel, M., Habel, C.: Verbally Assisted Virtual-Environment Tactile Maps: A Prototype System. In: Proceedings of the Workshop on Spatial Knowledge Acquisition with Limited Information Displays, pp. 25–30 (2012)
11. Alaçam, Ö., Habel, C., Acartürk, C.: Towards Designing Audio Assistance for Comprehending Haptic Graphs: A Multimodal Perspective. In: Stephanidis, C. (ed.) UAHCI/HCII 2013, Part I. LNCS, vol. 8009, pp. 409–418. Springer, Heidelberg (2013)
12. Habel, C., Kerzel, M., Lohmann, K.: Verbal Assistance in Tactile-Map Explorations: A Case for Visual Representations and Reasoning. In: Proceedings of AAAI Workshop on Visual Representations and Reasoning 2010, pp. 34–41. AAAI, Menlo Park (2010)
13. Tversky, B.: Cognitive Maps, Cognitive Collages, and Spatial Mental Models. In: Campari, I., Frank, A.U. (eds.) COSIT 1993. LNCS, vol. 716, pp. 14–24. Springer, Heidelberg (1993)
14. Tversky, B., Zacks, J., Lee, P., Heiser, J.: Lines, Blobs, Crosses and Arrows: Diagrammatic Communication with Schematic Figures. In: Anderson, M., Cheng, P., Haarslev, V. (eds.) Diagrams 2000. LNCS (LNAI), vol. 1889, pp. 221–230. Springer, Heidelberg (2000)
15. Giudice, N.A., Palani, H., Brenner, E., Kramer, K.M.: Learning Non-Visual Graphic Information using a Touch-Based Vibro-Audio Interface. In: Proceedings 14th International ACM SIGACCESS Conference on Computers and Accessibility, pp. 103–110. ACM (2012)
16. Parkes, D.: NOMAD - An Audio-Tactile Tool for the Acquisition, Use and Management of Spatially Distributed Information by Partially Sighted and Blind People. In: Tatham, A., Dodds, A. (eds.) Proceedings of the 2nd International Conference on Maps and Graphics for Visually Disabled People, pp. 24–29. King's College, London (1988)
17. Moustakas, K., Nikolakis, G., Kostopoulos, K., Tzovaras, D., Strintzis, M.G.: Haptic Rendering of Visual Data for the Visually Impaired. IEEE Multimedia 14, 62–72 (2007)
18. Wang, Z., Li, B., Hedgpeth, T., Haven, T.: Instant Tactile-Audio Map: Enabling Access to Digital Maps for People with Visual Impairment. In: Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility, pp. 43–50. ACM, Pittsburg (2009)
19. Lederman, S.J., Klatzky, R.L.: Haptic Perception: A Tutorial. Attention, Perception, and Psychophysics 71, 1439–1459 (2009)
20. Allen, J.: Maintaining Knowledge about Temporal Intervals. Communication of the ACM 26, 832–843 (1983)

21. Eschenbach, C., Kulik, L.: An Axiomatic Approach to the Spatial Relations Underlying 'left'–'right' and 'in front of'–'behind'. In: Brewka, G., Habel, C., Nebel, B. (eds.) KI 1997. LNCS (LNAI), vol. 1303, pp. 207–218. Springer, Heidelberg (1997)
22. Eschenbach, C., Habel, C., Kulik, L.: Representing Simple Trajectories as Oriented Curves. In: Kumar, A.N., Russell, I. (eds.) FLAIRS 1999, Proceedings 12th International Florida AI Research Society Conference, pp. 431–436. AAAI, Menlo Park (1999)
23. Habel, C.: Representational Commitment in Maps. In: Duckham, M., Goodchild, M., Worboys, M. (eds.) Foundations of Geographic Information Science, pp. 69–93. Taylor & Francis, London (2003)
24. Klippel, A.: Wayfinding Choremes. In: Kuhn, W., Worboys, M.F., Timpf, S. (eds.) COSIT 2003. LNCS, vol. 2825, pp. 301–315. Springer, Heidelberg (2003)
25. Reiter, R., Mackworth, A.: A Logical Framework for Depiction and Image Interpretation. Artificial Intelligence 41, 125–155 (1989)
26. Pfoser, D., Jensen, C.: Indexing of Network Constrained Moving Objects. In: Proceedings of the 11th ACM International Symposium on Advances in Geographic Information Systems, pp. 25–32. ACM (2003)
27. Shipley, T., Maguire, M.: Geometric Information for Event Segmentation. Understanding Events: How Humans See, Represent, and Act on Events. In: Shipley, T.F., Zacks, J.M. (eds.) Understanding Events: How Humans See, Represent, and Act on Events, pp. 415–435. Oxford University Press, New York (2008)
28. Luckham, D.C.: The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Addison-Wesley, Reading (2002)
29. Eckert, M., Bry, F.: Complex Event Processing (cep). In: Informatik-Spektrum 32, pp. 163–167 (2009)
30. Hayes-Roth, F.: Rule-Based Systems. Communications of the ACM 28, 921–932 (1985)
31. Anderson, J.R.: ACT: A Simple Theory of Complex Cognition. American Psychologist 51, 355–365 (1996)
32. Forgy, C.L.: Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem. Artificial Intelligence 19, 17–37 (1982)
33. Walzer, K., Breddin, T., Groch, M.: Relative Temporal Constraints in the Rete Algorithm for Complex Event Detection. In: Proceedings of the Second International Conference on Distributed Event-Based Systems, pp. 147–155. ACM (2008)
34. Kerzel, M.: Rule Patterns for Event Recognition During Exploration of Hapic Virtual Environment Line-based Graphics. Technical Report, Department of Informatics, University of Hamburg, Germany (2013)
35. Mackworth, A.K.: Adequacy Criteria for Visual Knowledge Representation. In: Pylyshyn, Z. (ed.) Computational Processes in Human Vision, pp. 464–476. Ablex Publishers, Norwood (1988)