TECHNISCHE UNIVERSITÄT DRESDEN

Formalization of Gestural Input for Multitouch-Systems

Summary

Dissertation Thesis

submitted to
Technische Universität Dresden
Faculty of Computer Science

submitted by
Dipl.-Medieninf. Dietrich Kammer
born 29.01.1984 in Dresden

Adviser and referee:
Prof. Dr.-Ing. habil. Rainer Groh (Technische Universität Dresden)

Referee:
Prof. Dr. Beat Signer (Vrije Universiteit Brussel)

Consultant:
Prof. Dr.-Ing. Raimund Dachselt (Technische Universität Dresden)

Dresden, September 2013

This summary is a translation of the officially submitted short version of the german dissertation »Formalisierung gestischer Eingabe für Multitouch-Systeme«. More details and references can be found in the original work. Please refer to the cited papers in english language to get more information.

1 Motivation and Goals

Thanks to novel input technologies different from keyboard and mouse, human-computer interaction is becoming richer, more intuitive, and more versatile. For instance, touch sensitive surfaces allow direct input with fingers without the use of additional peripheral devices (siehe Figure 1). Users can interact directly with the displayed content using their hands. For more precise input, digital pens are being used more frequently. Gestures that are performed freely in space increase the number of interaction possibilities for humans (see Table 1). However, by dispensing with additional input devices to interact with computers, processing input becomes more intricate and complex. Managing this compexity is a challenge not only for developers of the required tracking hardware, but also for programmers and designers of user interfaces. Existing models and tools for programming user interfaces are laid-out for keyboard and mouse. More complex gestures have to be recognized and processed by the programmer based on raw data supplied by the input hardware. This results in extensive programming efforts, little reuse of existing code, and failures while programming

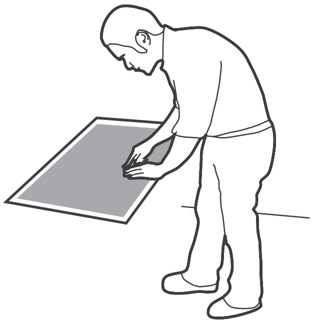


Figure 1: Large interactive surfaces allow a large variety of so-called planar gestures for input (cp. [Kammer u. a. 2010b])

The contributions of this thesis consist of a semiotic discussion in the area of gestural input for human-computer interaction. To this end, the instruments of semiotics are considered in an interdisciplinary discussion and insights from gesture research are reviewed. Hence, the thesis is situated in the intersection of three disciplines or areas of research (see Figure 2). For the formalization of gestural input, a general semiotic model is proposed. This model is applied to planar gestures in the area of multitouch interaction. The developed Gesture Formalization for Multitouch (GeForMT) defines a domain-specific language to describe planar gestures. In contrast to other formalization approaches, this gestures language follows a pictorial approach. Instead of defining detailed filters for raw input data, GeForMT uses geometrical and topological parameters to define gestures. The design of gestures is supported, for example by identifying conflicts between similar gestures at an early stage in the development process. The formalized gestures can be embedded in code and hence facilitate the programming. The corresponding framework is responsible to implement the necessary algorithms for gesture recognition.

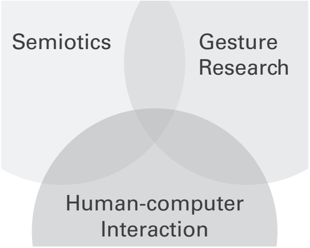


Figure 2: Related areas of research

First and foremost, the proposed formalization addresses developers of software for multitouch systems. They can be programmers as well as designers with different skill sets and expertises. This dissertation focuses exclusively the authoring process during gesture development. Hence, no usability studies are conducted for gestural user interfaces. The development of guidelines for the concrete and usable design of interfaces can be based on the foandations laid in this work. The emphasis in this dissertation is on aspects of gesture input rather than the visual design of user interfaces. Questions related to this topic are only touched upon and not answered in depth.

2 Interdisciplinary Perspectives and Theoretical Foandation

Gestures are subject of investigation in different scientific disciplines, for instance cognitive psychology, philosophy, communication studies and other natural sciences as well as human sciences. Due to this broad spectrum, finding a single definition of the term gesture is difficult. Since commonly only a specific branch of gestures is investigated, McNeill introduces a continuum to categorize human gestures, that was first described by Kendon [McNeill 1995, p. 37; Kendon 2004, p. 104ff]. Kendon's continuum is a tool to

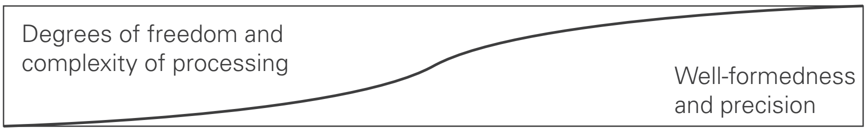
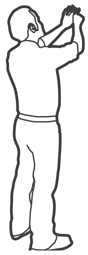
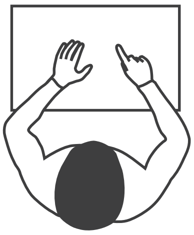
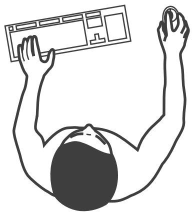
	Spatial Gestures	Planar Gestures	Sketching	Keyboard-Mouse-System
Spectrum				
Tracking data	Video or motion capture	Video or position data	Position and pressure	Keypresses and movements
Raw data	Skeleton (3D coordinates)	n-paths (2D coordinates)	1-path (2D coordinates)	Key combinations, mouse position
Context	Space	(Planar) surface	Pen and paper	Peripheral devices
Technical Setup				

Table 1: Forms of gestural input, the scope of the thesis are surface gestures (»planar gestures«)

clarify which type of gestures is investigated. Figure 3 shows the fandamental distinction between spontaneously produced gestures and codified signs. The role of speech is important for spontaneously produced gestures. In contrast, sign language can substitute speech entirely by gestures. While cognitive psychologists and sociologists are mainly concerned with gesticulation, gestures in human-computer interaction can rather be treated as so-called emblems. Emblems have to be performed in a well-formed manner to be anderstood properly. An example is the showing of a thumb to communicate agreement. For a proper anderstanding of this gesture, a cultural tradition or convention is necessary.

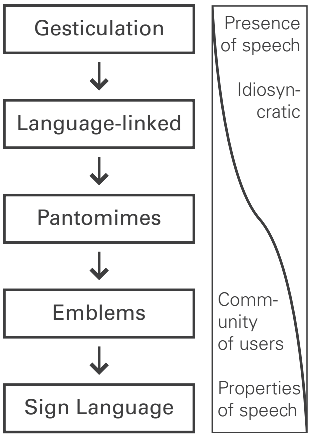


Figure 3: Human gestures can be categorized using Kendon's continuum (see [Kendon 2004, p. 104ff])

While semiotic terms [Nöth 2000] are being discussed in computer science, they are rarely used productively (vgl. [Andersen 2001]). Nake concluded that computer operations can be regarded as sign processes, that are represented by different means [Nake 1993, p. 31]. In contrast to existing analytical strategies, productive approaches based on semiotics have rarely been proposed (cp. [Souza 2005]). This thesis seeks to productively apply the means of semiotics in the area of planar gestures.

3 Related Formalization Approaches

The manifold representations of gestures in computer science can be situated on three levels: the algorithm level, the exchange level, and the user level. On the algorithm level, mathematical formalisms and statistical data structures are used for an efficient and precise recognition of gestural input. The user level is responsible for communicating gestural input to the user in an effective, but also informal way. Tutorials in natural language, pictures, icons, and animations can be used. The exchange level is situated between the algorithms of gesture recognition and the user. It is the bridge between mathematical algorithms and the dynamic and informal perspective of the user.

3.1 Spatial Gestures

Spatial gestures need to address three dimensions and hence six degrees of freedom. In addition, movements of fingers, arms, legs, head, and the whole body can be distinguished. This large diversity results in a high complexity of the required processing. The application scenarios of existing formalization approaches differ considerably: gesture vectors are used mainly to annotate gestures in video sequences [Trippel et al. 2004], detector nets are used to program gestures for virtual construction scenarios [Latoschik 2001], and the Behaviour Markup Language (BML) is focused on describing multi-modal applications [Kopp et al. 2006]. Since BML is based on XML, it is very detailed and human-readable. However, it tends to become bloated and very complex. Detector nets are rather tightly boand to gesture recognition algorithms and hence on the algorithm level instead of the exchange level.

3.2 Gestures in Sketching

Digital sketching does not only use handwriting recognition for text input, but also symbolic stroke gestures to execute commands and make corrections. They are similar to planar gestures, but can be performed with higher precision thanks to the pen as tool. Gestures are used to execute diverse commands, such as the selection of objects or text, deletion, and ando of input. Cho [Cho 2006] proposes a special formalism based on gesture function in order to correctly recognize correction signs while sketching. LADDER [Hammond and Davis 2005] and the Sketch Language [Bimber et al. 2000] use domain-specific languages to implement sketching interfaces. LADDER is based on geometric properties of stroke gestures.

3.3 Planar Gestures

Concurrently to the gesture formalization developed in this thesis, six further formalization approaches have been proposed. On the one hand, theoretical methods of computer science such as regular expressions or logics are used by Proton [Kin et al. 2012], Midas [Scholliers et al. 2011] and Framous [Görg et al. 2010]. On the other hand, GDL [Khandkar and Maurer 2010], GISpL [Echtler and Butz 2012] and GDML [@NUIGroup 2009a] use concrete formats such as JSON and XML or tools for domain-specific languages. Midas and GISpL explicitly address the description of further gestural input and are thus more abstract and less specialized for planar gestures. The related formalization approaches tend to offer filters for the events that are generated during input. The approach in this thesis relies (like LADDER for sketching) on the topological and geometrical aspects of gesture execution. Hence, a not previously available tool is developed to describe gestures in a pictorial way.

4 Semiotic Model for Formalization

The semiotic model provides the structure to approach the formalization of a specific form of gestural input for human-computer interaction. By addressing the three fandamental aspects, syntax, semantics, and pragmatics, the design space and challenges of using gestures are identified. The categories and concepts of semiotics are used as practical tools to structure a domain of knowledge.

The syntax is concerned with the formal relationships between signs. A general grammar of a certain form of gestural input for communication between human and computer has to be established. Semantics establishes the relationships between sign combinations and their meaning. It is necessary to address both the technical issues that are determined by the computer as well as the mental model of the user. Pragmatics addresses the relationship between human and sign usage. The meaning of a sign is placed in a certain context by the user. Figure 4 shows the proposed semiotic model for the formalization of gestural input. The human sphere of competence is situated in the area of pragmatics. The assignment of meaning in the area of semantics is at the interface between humans and the syntax which is determined by the processing of the computer.

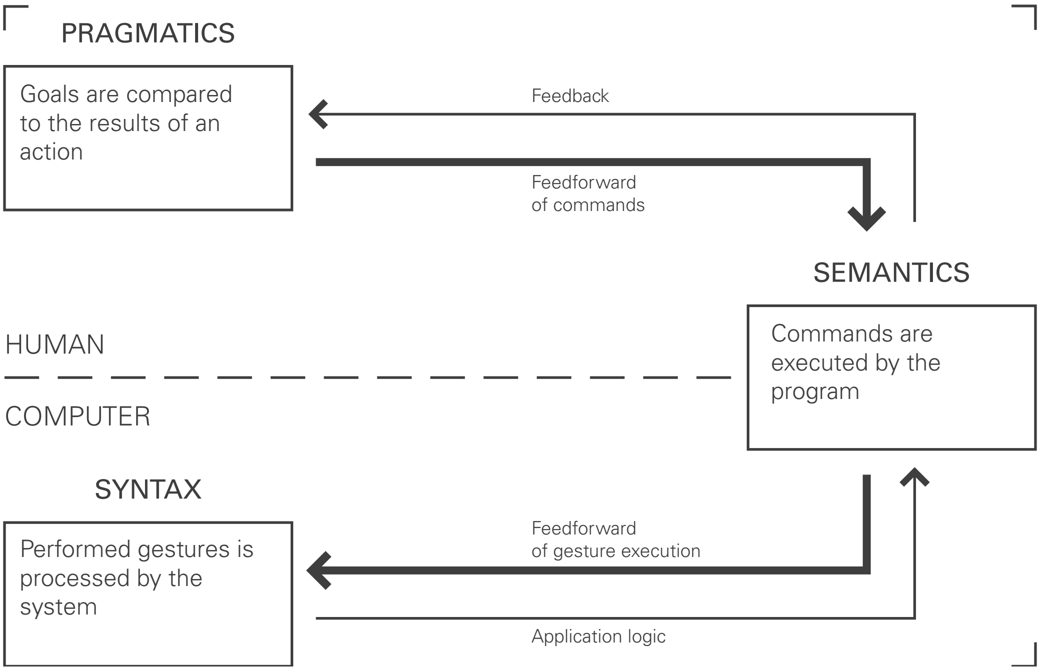

PRAGMATICS
Goals are compared to the results of an action
Feedback
Feedforward of commands
SEMANTICS
Commands are executed by the program
HUMAN
COMPUTER
SYNTAX
Performed gestures is processed by the system
Feedforward of gesture execution
Application logic

Figure 4: The semiotic model describes the interaction between human and computer by means of gestural input

The requirements for a gesture formalization according to the semiotic model are manifold and address both technical issues of recognizing gestural input as well as the application in user interfaces. The following questions need to be answered when applying the semiotic model in order to formalize gestural input:

- What are the atomic syntactical units of a form of gestural input?
- How can these atomic units be recognized reliably?
- How can the atomic units be combined in more complex gestures?
- How can the combinations be validated in order to avoid inconsistencies?
- How are gesture definitions managed and linked with the application logic?
- Which structural relationships exist between syntactical and semantic aspects?
- What are possible feedforward mechanisms, that support users during interaction?
- What are possible feedback mechanisms, that afford intuitive interaction for the user?

5 Gesture Formalization for Multitouch

The semiotic model has been applied for multitouch interaction, resulting in the Gesture Formalization for Multitouch (GeForMT) [Kammer et al. 2010b]. The three areas, syntax, semantics, and pragmatics, are addressed in this section.

5.1 Syntax of Planer Gestures

Planer gestures are represented as simultaneous touch contacts on a touch sensitive surface that are produced by the fingers and hands of users. In order to formalize these contacts, basic forms and movements are identified, which can be combined to a certain extent. Temporal and spatial constraints between simultaneous touch contacts are another important concern of GeForMT. In addition, the object focus and processing modes of gestures are addressed.

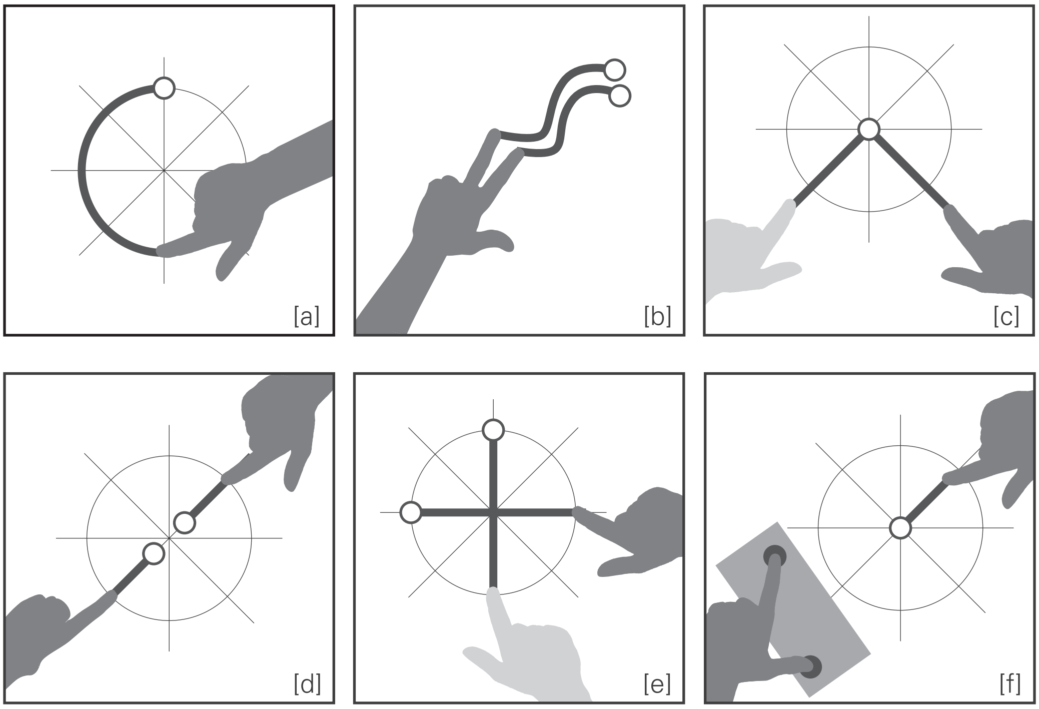


Figure 5: Ausführungsformen flächiger Gesten: Dunkelgraue Linien oder Kurven (Gestenzüge), geometrische Grundformen werden zur Verdeutlichung auf einer Windrose idealisiert angeordnet, kleine Kreise (Startpunkt), dunkelgraue Hände (Ende eines Gestenzugs), ausgefüllte Kreise (Haltegesten), hellgraue Hände (bereits beendeter Gestenzug der zur Geste gehört), rechteckige Interface-Elemente (Objektbezug)
[a] 1F(SEMICIRCLE_S_CCW)
[b] 2F(MOVE)
[c] CONNECT_START[
LINE_SW;LINE_SE]
[d] SPLIT[LINE*LINE]
[e] CROSS[LINE_S;LINE_E]
[f] 2F(HOLD(rectangle)) + LINE
(nach [Kammer u. a. 2010b])

The Extended Backus Naur Form (EBNF) is a formal language, for instance to describe the syntax of programming languages (cp. [Schöning 2003, S. 25f]). GeForMT consists of 17 production rules, that are shown in Listing 1. These production rules can be further reduced. However, the current list is easier to extend and the rules are more comprehensible. Gesture definitions consist of a unique identifier and a list of alternative complex gestures, separated by a vertical bar (rule 1). Additionally, the gesture can be explicitly flagged to be evaluated continuously (online) or discretely (offline). For online processing, continuous events are being produced once the gesture is recognized. Discrete gestures only produce one event after the gestures is completely performed (rule 17).

Complex gestures (rule 3) consist of simple gestures, that are separated by an operator for temporal relations (rule 8). Optionally, a spatial relation can be added (rule 9). Simple gestures in turn consist of optional functions that describe the nature of the touch contact and a set of combined atomic gestures including their object focus (rule 4). Alternatively, an already defined identifier can be reused in a complex gesture in rule 4, which is specified in angled brackets.

The declaration of an object focus for atomic gestures is optional (rule 5). The object focus can be indicated by a list of comma-separated identifiers (rule 6). Functions consist of a number and an identifier which designates the nature of the touch contact, for example `F` for Finger or `H` for Hands (rule 7).

```
1   definition  ::=  id '=' complex ('|' complex)*[':'time]';'
2   id          ::=  [A-Za-z0-9]+
3   complex     ::=  gesture (operator gesture)* |
                     relation'['gesture (operator gesture)*']'
4   gesture     ::=  function'('atomfocus(','atomfocus)*')' |
                     atomfocus(','atomfocus)* | '<'id'>'
5   atomfocus   ::=  atom | atom'('focuslist')'
6   focuslist   ::=  id(','id)*
7   function    ::=  [integer]'F' | [integer]'H' | [integer]'B'
8   operator    ::=  '*' | '+' | ',' | ';'
9   relation    ::=  'SYNC' | 'JOIN' | 'SPLIT' | 'CROSS' |
                     'CONNECT_START' | 'CONNECT_END' | 'ASIDE' |
                     'AMONG' | 'CLOSE' | 'ADJOIN_'direction
10  atom        ::=  identifier | vector['_'direction] |
                     shape['_'direction]['_'rotation]
11  identifier  ::=  'MOVE' | 'POINT' | 'DEPOINT' | 'HOLD'
12  vector      ::=  'LINE'
13  shape       ::=  'CIRCLE' | 'SEMICIRCLE'
14  direction   ::=  'NORTH' | 'N' | 'NORTHEAST' | 'NE' | 'EAST' |
                     'E' | 'SOUTHEAST' | 'SE' | 'SOUTH' | 'S' |
                     'SOUTHWEST' | 'SW' | 'WEST' | 'W' |
                     'NORTHWEST' | 'NW' | 'S~' | 'W~' | 'N~' | 'E~'
15  rotation    ::=  'CLOCKWISE' | 'CW' | 'COUNTERCLOCKWISE' | 'CCW'
16  integer     ::=  [1-9][0-9]*
17  time        ::=  'ON' | 'OFF'
```

Listing 1: EBNF grammar of the Gesture Formalization for Multitouch (GeForMT) (see [Kammer u. a. 2010b])

Atomic gestures (rule 10) consist of either keywords without a direction to describe freeforms, tap, and hold gestures (rule 11), or vector based gestures with a direction (rule 12), or further shapes with optional specification of direction and rotation (rule 13). Examples of formal gestures are shown in Figure 5.

5.2 Semantics of Planer Gestures

Planar gestures serve different interaction goals and actions in an application. A strict categorization is sometimes difficult, as McNeill already posited for the area of speech-accompanying gestures [McNeill 2007, p. 38]. Also planar gestures possess different dimensions, that determine their semantic function to different degrees. However, we propose five main categories to distinguish gestures. The classes shown in Figure 6 are subclasses of the emblem category in Kendon's continuum.

The term deictic gestures that is often foand in the literature addresses the selection of objects and object groups. Manipulative gestures serve to alter objects in their visual properties such as form, size, orientation, or location. Navigational gestures serve the movement of a user in an application. This usually results in the change of the whole appearance of the application and not only a single object. Figure 6 shows an exampe of a navigational gesture, where the corner of the application is tapped, thereby jumping to another layer. Iconic gestures encompass the drawing of arbitrary forms, for instance in a sketching or modeling application. However, the graphical form is not furhter interpreted by the system. Hence, no concrete meaning is associated or a certain function is

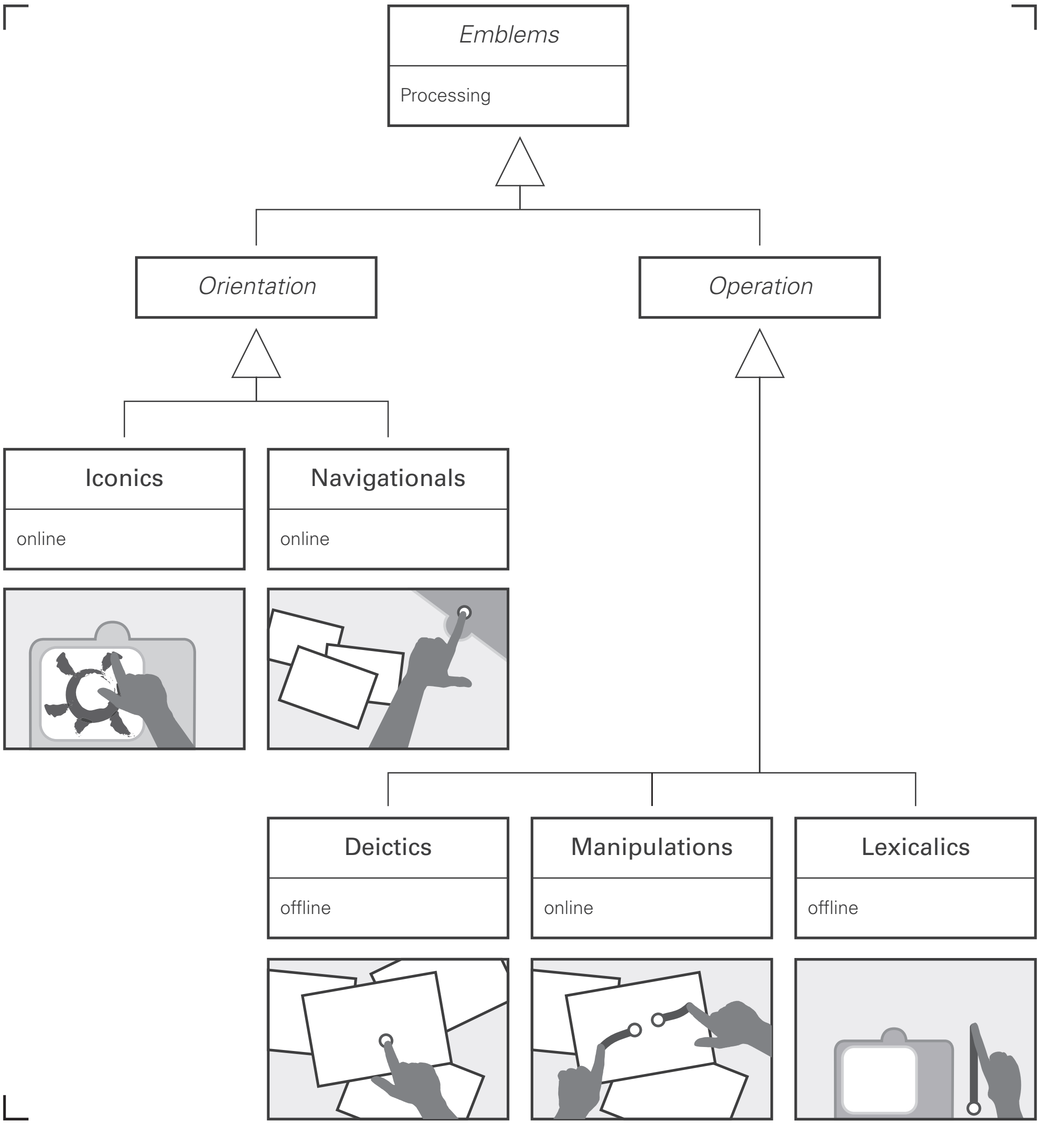


Figure 6: Semantic dimensions of planar gestures, organized according to the interaction goals operation and orientation (see [Kammer u. a. 2010b])

executed. Figure 6 shows how a drawing is used as graphical icon in an application. If a gesture has an arbitrary function assigned, it is called lexicalic. Abstract commands are possible, such as activating a help function by drawing a question mark. The term refers to the possibility of creating a lexicon where each syntax is assigned a certain function. Lexicalic gestures are different from manipulative and navigational gestures because those possess a more direct relation of execution and action. The lexicalic gesture shown in Figure 6 activates a menu by performing a pulling movement from the edge of the application.

The semantic dimensions can be grouped in a hierarchy of abstract interaction goals, namely operation and orientation (cp. [Groh 2007, p. 155]). Figure 6 shows that navigational gestures are used for orientation, in order to experience and search through a digital space. Iconic gestures can also be used to create personal marks that can later be used to find already used interaction paths or identify individually tagged objects. Operational gestures encompass deictic, manipulative and lexicalic gestures. These gestures produce state changes or perform selections. However, lexicalic gestures can also be used for orientation, for instance by changing views in an application.

As mentioned before, a single gesture can encompass various semantic dimensions (see [Kammer et al. 2011a]). Syntactically equivalent gestures can be used for direct manipulations, but also for arbitrarily assigned functions as in lexicalic gestures. The computer can constrain manipulations, check for inconsistencies or add functional features.

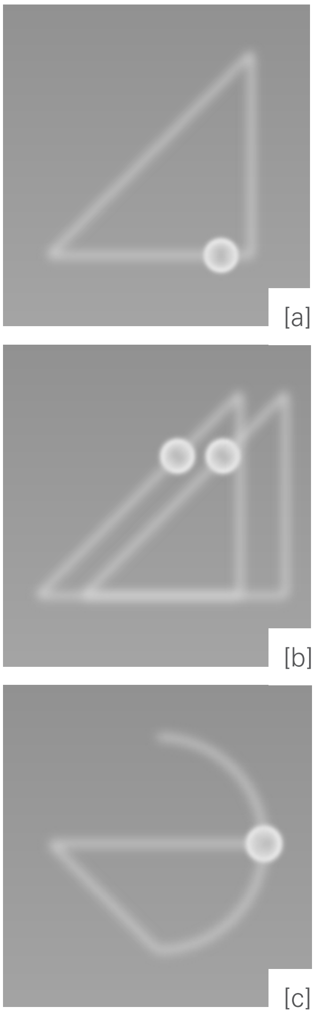


Figure 7: Animations of complex gestures, that are combinations of atomic gestures
[a] 1F(LINE_NE,LINE_S,LINE_W)
[b] 2F(LINE_NE,LINE_S,LINE_S)
[c] 1F(SEMICIRCLE_S_CW,
LINE_NW,LINE_E)

5.3 Pragmatics of Planar Gestures

Describing all gestures in an application with GeForMT does not only improve and facilitate gesture recognition. On the one hand, feedback can be generated if a current input does not match any of the registered gestures. On the other hand, animations can be generated based on the formal descriptions. To this end, an Animation Framework for GeForMT (AnFraGe) has been conceived. For the generation of the feedforward animations, a module system based on the compass rose is used, on which the atomic gestures of GeForMT are based (see Figure 7).

Evaluation of the formalization approaches for planar gestures with student developers showed that GeForMT was rated more concise and readable. Compared to GDL and GISpL, GeForMT was rated equivalent for conciseness, accuracy, readability, and complexity.

6 Reference Architecture

In Echtler and Klinker's multitouch software architecture, GeForMT is situated in the interpretation layer, which processes calibrated finger and hand position from the anderlying transformation layer [Echtler and Klinker 2008]. Events are propagated to the widget layer above, that registers gestures and interface elements (regions) in the interpretation layer. Figure 8 shows the general components and processing steps that are needed when implementing GeForMT. The parser is responsible for processing formal gestures, a unified data model stores gestures, the gesture recognizer processes raw input data, and the matching is responsible to compare formal gestures and user input. Optionally, feedforward and feedback mechanisms can be provided by the library.

The reference implementation »GeForMTjs« of the described architecture has been implemented in JavaScript [Kammer et al. 2012a]. With the help of the JavaScript implementation, tools for supporting the process of gesture definition have been developed. At the heart of these tools that allow the pictorial programming of gestures is the compass rose, which visualizes the available basic geometrical units. It is used as an interactive element in an editor, to define atomic gestures by example (see Figure 9). A different number of fingers can be used as well.

7 Practical Examples and Further Development

First, protypical applications were analyzed with the help of GeForMT that were developed in student courses (see Figure 10). In these courses, a multitouch framework without the possibility of declaratively describing gestures was used. The expressive power of GeForMT could be validated by formalizing gestures that were developed independently of the formalization approach. Second, two further applications have been extended with a multitouch interface based on the reference architecture. DelViz is a search interface for information visualizations based on mouse input, which has been

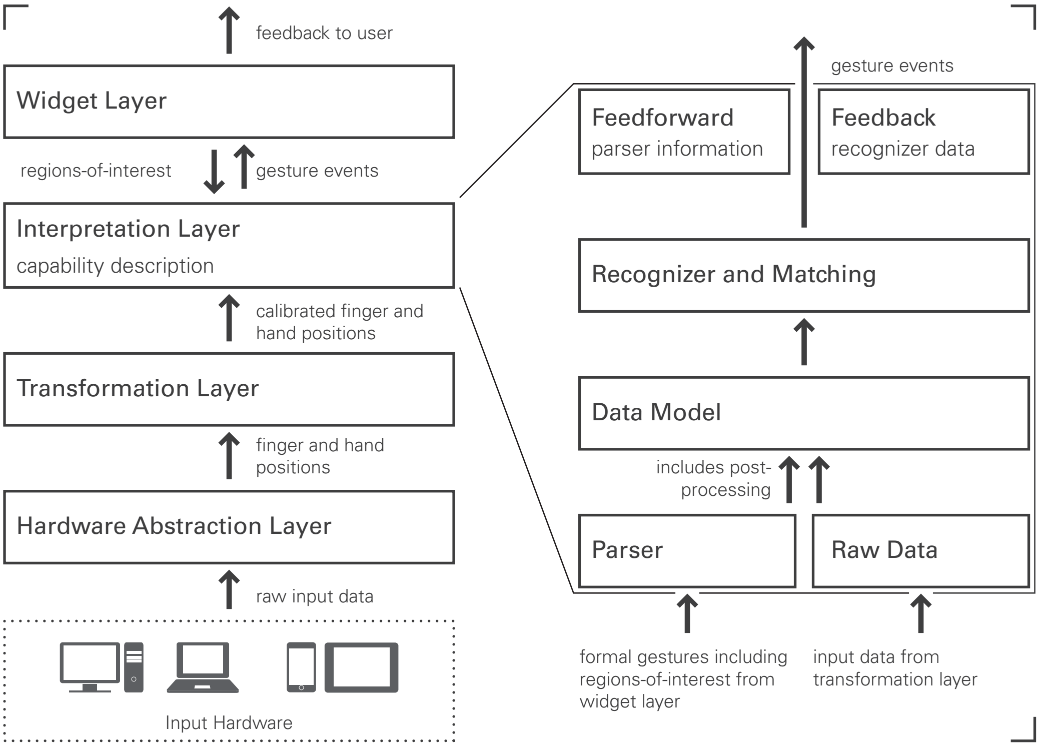


Figure 8: Multitouch architecture by Echtler and Klinker (see [Echtler und Klinker 2008]) with reference architecture for GeForMT in the interpretation layer

adapted and extended for multitouch interaction [Keck et al. 2011a]. The component-based 3D framework Bildsprache LiveLab [Kammer et al. 2012d] has also been extended by a modular implementation of the reference architecture in order to investigate planar gestures in 3D visualizations. Based on the formalized gesture sets, statistical analyses were conducted in order to compare applications according to the gestures they employ.

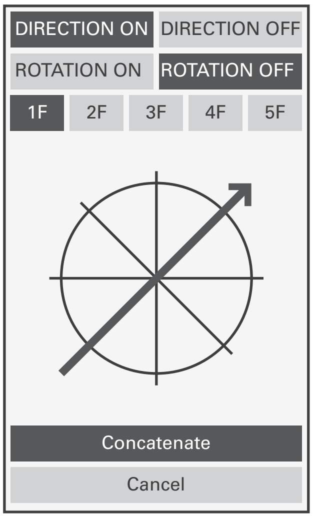


Figure 9: Interactive compass rose with additional properties to adjust direction and rotation intolerance and number of fingers to be used

Further forms of gestural input are suited as application domains for the proposed semiotic model. The tentative formalization approaches developed in this thesis are subject of future work. In contrast to planar gestures, spatial gestures need to address more degrees of freedom. Hence, the extension of the existing formalization approach to this form of gestural input represents a challenge. Spatial gestures that are performed over a multitouch tabletop are most likely to be combined with planar gestures, that are formally described with GeForMT [Kammer et al. 2011b]. Here, the introduction of namespace can help to disambiguate and separate the formal gesture languages. The established semiotic model is also suitable to explore and consolidate the invention of novel interactive elements that can be manipulated using planar gestures. These interactive elements can be developed by conducting experiments with natural everyday substances [Brade et al. 2011a]. The handling of these substances is fandamentally different from the object centered and geometrical approach of GeForMT [Groh 2011, p. 78].

The tabletop system »DepthTouch« uses an elastic surface to interact with digital content. This results in an increase of haptic qualities and an additional interaction dimension: depth [Gründer et al. 2013]. Pushing, pulling, and planar gestures can be used for manipulating the elastic surface. In this context, planar gestures can be formally expressed using GeForMT. However, pressure needs to be added as another parameter.

8 Contributions and Future Work

The goal of this dissertation was the development of models and tools for gestural input in human-computer interaction. To this end, semiotics as structured approach for analyzing systems of signs and communication processes was exploited. Furthermore, a literature survey regarding gesture reserach was conducted, in order to precisely define the object of investigation. The general semiotic model required for the formalization was tested in the area of planar gestures.

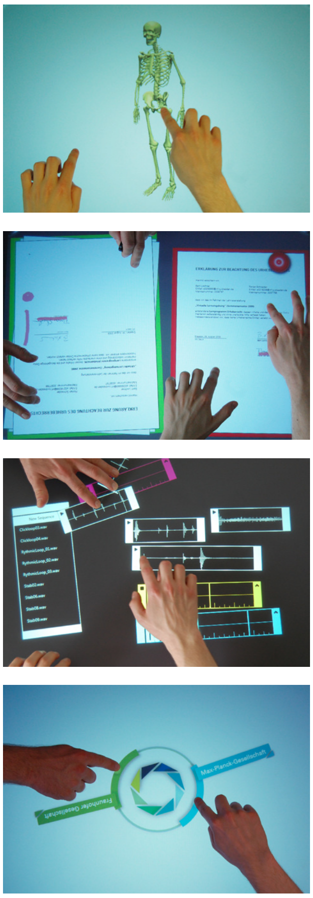


Figure 10: Prototypical applications for tabletops have been analyzed using GeForMT (see [Franke u. a. 2010; Kammer u. a. 2011a])

For the developed concepts, a reference architecture has been proposed and its applicability has been tested using practical examples. For future work, the further development of the formalization for spatial gestures, everyday substances, and elastic gestures was discussed.

Another research question concerns the mutual mapping of the existing formalization approaches for planar gestures. For instance, the conversion of the pictorial repertoire of GeForMT into the rule-based description of Midas can be pursued. This would result in more generic conclusions regarding the expressive power of the different gesture languages. The existing concepts to express planar gestures with XML have not been tested in practice. Such a format would be suitable to archive gestures described with GeForMT for the mapping to other frameworks. In the XML format, further and more precise parameters and constraints could be added that are not included in the short and precise DSL. In this context, the integration into further domain-specific approaches and frameworks could be investigated.

The proposed semiotic model for the formalization of gestural input does not impose a concrete structured procedure. The semiotic model does organize and structure the analysis of a given form of gestural input in the area of human-computer interaction. However, unlike process models for projects and software engineering, no temporal requirements or milestones are defined. Based on further experiences with semiotic formalization, more concrete standards can be developed.

9 References

[Andersen 2001] Andersen, P. B.: What Semiotics can and cannot do for HCI, *Knowledge-Based Systems*, 14(8), S, 419–424, 2001

[Bimber et al. 2000] Bimber, O.; Encarnacao, L. M. and Stork, A.: A multi-layered architecture for sketch-based interaction within virtual environments, *Computers & Graphics*, 24(6), S.851–867, 2000

[Brade et al. 2011a] Brade, M.; Kammer, D.; Keck, M. and Groh, R.: „Immersive data grasping using the explore table", In: *Proc. of the 5th TEI*, TEI'11, ACM, New York, NY, USA, S. 419–420, 2011 (ISBN: 978-1-4503-0478-8)

[Cho 2006] Cho, M.: A new gesture recognition algorithm and segmentation method of Korean scripts for gesture-allowed ink editor, *Information Sciences*, 176(9), S, 1290–1303, 2006

[Echtler and Butz 2012] Echtler, F. and Butz, A.: „GISpL: gestures made easy", In: *Proc. of the 6th TEI*, TEI'12, ACM, New York, NY, USA, S. 233–240, 2012 (ISBN: 978-1-4503-1174-8)

[Echtler and Klinker 2008] Echtler, F. and Klinker, G.: „A multitouch software architecture", In: *Proc. of the 5th NordiCHI: building bridges*, NordiCHI'08, ACM, New York, NY, USA, S. 463–466, 2008 (ISBN: 978-1-59593-704-9)

Franke et al. 2010] Franke, I. S.; Kammer, D.; Groh, R.; Happ, S.; Steinhauf, J. and Schönefeld, F.: Developing Multi-touch Software through Creative Destruction, *Mensch & Computer 2010 Entertainment Interfaces Track*, S. 45–51, 2010

[Görg et al. 2010] Görg, M. T.; Cebulla, M. and Garzon, S. R.: „A Framework for Abstract Representation and Recognition of Gestures in Multi-touch Applications", In: *Proc. of 3rd ACHI*, 3rd ACHI, St Maarten, Netherlands Antilles, S. 143–147, 2010

[Groh 2011] Groh, R.: „Was sieht die Hand?", In: Groh, R. and Zavesky, M. (Eds.): *Wieder mehr Sehen! - Aktuelle Einblicke in die Technische Visualistik*, TUDpress Verlag der Wissenschaft, Dresden, S. 75–80, 2011 (ISBN: 978-3942710398)

[Groh 2007] Groh, R.: *Das Interaktions-Bild : Theorie and Methodik der Interfacegestaltung*, TUDpress, Dresden, 2007 (ISBN: 9783938863053)

[Gründer et al. 2013] Gründer, T.; Kammer, D.; Brade, M. and Groh, R.: „Towards a Design Space for Elastic Displays", In: *ACM SIGCHI CHI - Workshop: Displays Take New Shape: An Agenda for Future Interactive Surfaces*, Paris - France, 2013

[Hammond and Davis 2005] Hammond, T. and Davis, R.: LADDER, a sketching language for user interface developers, *Computers & Graphics*, 29(4), S, 518–532, 2005

[Kammer et al. 2012a] Kammer, D.; Henkens, D. and Groh, R.: „GeForMTjs: A JavaScript Library Based on a Domain Specific Language for Multi-touch Gestures", In: *Web Engineering - 12th International Conference*, ICWE 2012, Berlin, Germany, July 23-27, 2012. S. 444–447, 2012

[Kammer et al. 2012d] Kammer, D.; Wojdziak, J.; Ebner, T.; Franke, I. S. and Groh, R.: A component-oriented framework for experimental computer graphics, *Computer Standards & Interfaces*, 34(1), S, 93–100, 2012

[Kammer et al. 2011a] Kammer, D.; Franke, I. S.; Groh, R.; Steinhauf, J.; Kirchner, M. and Schönefeld, F.: „The eleventh finger: levels of manipulation in multi-touch interaction", In: *Proc. of the 29th ECCE*, ECCE '11, ACM, New York, NY, USA, S. 141–148, 2011 (ISBN: 978-1-4503-1029-1)

[Kammer et al. 2011b] Kammer, D.; Henkens, D.; Wojdziak, J. and Freitag, G.: „Formalization and Combination of Touch and Point Interaction", In: *Workshop on Engineering Patterns for Multitouch Interfaces, Engineering Interactive Computer Systems (EICS) 2011*, Pisa, Italy, 2011

[Kammer et al. 2010b] Kammer, D.; Wojdziak, J.; Keck, M.; Groh, R. and Taranko, S.: „Towards a formalization of multi-touch gestures", In: *ACM ITS*, ITS'10, ACM, New York, NY, USA, S. 49–58, 2010 (ISBN: 978-1-4503-0399-6)

[Keck et al. 2011a] Keck, M.; Kammer, D.; Iwan, R.; Taranko, S. and Groh, R.: DelViz: Exploration of Tagged Information Visualizations Heiß, H.-U.; Pepper, P.; Schlingloff, H. and Schneider, J. (Eds.): *Tagungsband der INFORMATIK 2011 Lecture Notes in Informatics*, Band P192, Gesellschaft für Informatik e.V. (GI), S. 257–266, 2011 (ISBN: 978-3-88579-286-4)

[Kendon 2004] Kendon, A.: *Gesture: Visible Action as Utterance*, Cambridge University Press, 2004 (ISBN: 0521542936)

[Khandkar and Maurer 2010] Khandkar, S. and Maurer, F.: „A Domain Specific Language to Define Gestures for Multi-Touch Applications", In: Rossi, M.; Tolvanen, J.-P.; Sprinkle, J. and Kelly, S. (Eds.): *Proc. of the 10th DSM*, The 10th Workshop on Domain-Specific Modeling, Aalto University School of Economics, B-120, Aalto-Print, 2010 (ISBN: 978-952-60-1043-4)

[Kin et al. 2012] Kin, K.; Hartmann, B.; DeRose, T. and Agrawala, M.: „Proton: multitouch gestures as regular expressions", In: *Proc. of the ACM CHI*, CHI'12, ACM, New York, NY, USA, S. 2885–2894, 2012 (ISBN: 978-1-4503-1015-4)

[Kopp et al. 2006] Kopp, S.; Krenn, B.; Marsella, S.; Marshall, A.; Pelachaud, C.; Pirker, H.; Thórisson, K. and Vilhjálmsson, H.: „Towards a Common Framework for Multimodal Generation: The Behavior Markup Language", In: Gratch, J.; Young, M.; Aylett, R.; Ballin, D. and Olivier, P. (Eds.): *Intelligent Virtual Agents*, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, S. 205–217, 2006

[Latoschik 2001] Latoschik, M. E.: „A gesture processing framework for multimodal interaction in virtual reality“, In: *Proc. of the 1st AFRIGRAPH*, AFRIGRAPH'01, ACM, New York, NY, USA, S. 95–100, 2001 (ISBN: 1-58113-446-0)

[McNeill 2007] McNeill, D.: *Gesture and thought*, University of Chicago Press; University Presses Marketing [distributor], Chicago, Ill.; Bristol, 2007 (ISBN: 9780226514635 0226514633 0226514625 9780226514628)

[McNeill 1995] McNeill, D.: *Hand and mind : what gestures reveal about thought*, University of Chicago Press, Chicago, 1995 (ISBN: 9780226561349)

[Nake 1993] Nake, F.: *Die erträgliche Leichtigkeit der Zeichen: Ästhetik, Semiotik, Informatik*, Agis-Verlag, Baden-Baden, 1993 (ISBN: 9783870070380)

[Nöth 2000] Nöth, W.: *Handbuch der Semiotik*, 2., vollständig neu bearbeitete and erw. Aufl., Metzler, Stuttgart, 2000 (ISBN: 9783476012265)

[@NUIGroup 2009a] NUIGroup Authors: Gesture Recognition - NUI Group Community Wiki, 2009, Online: http://wiki.nuigroup.com/Gesture_Recognition/, Letzter Abruf: 01.08.2013

[Scholliers et al. 2011] Scholliers, C.; Hoste, L.; Signer, B. and De Meuter, W.: „Midas: a declarative multi-touch interaction framework“, In: *Proc. of the 5th TEI*, TEI'11, ACM, New York, NY, USA, S. 49–56, 2011 (ISBN: 978-1-4503-0478-8)

[Souza 2005] Souza, C. S. de: *The semiotic engineering of human-computer interaction*, MIT Press, Cambridge, Mass., 2005 (ISBN: 0262042207 9780262042208)

[Trippel et al. 2004] Trippel, T.; Gibbon, D.; Thies, A.; Milde, J.-T.; Looks, K.; Hell, B. and Gut, U.: „CoGesT: a formal transcription system for conversational gesture“, In: *Proc. of LREC 2004*, LREC 2004, S. 2215–2218, 2004