Internship Paper

SESAME Project: Enhanced interaction techniques for an intuitive 3D modeling system

Supervised by: Prof. Dr. Wolfgang Stürzlinger
Prof. Dr.-Ing. Stefan Schlechtweg
Dipl.-Ing. Niklas Röber

Alexander Kuhn
Computervisualistik
Matrikelnr.: 168167
Magdeburg

January 2008

Abstract

This thesis deals with the issue of modeling three dimensional objects and the development of enhanced interaction methods for an existing 3D modeling system called SESAME. The aim is to create tools supporting the conceptual phase during a general shape creation process. For this reason it focusses on creating intuitive and expressive deformation methods, which should be fast to learn and easy to apply.

The development of the concepts is based on guidelines, which were derived from the experiences made with existing approaches. Furthermore, the presented methods are supposed to work conceptually together with existing modeling paradigms and should extend the range of possible shapes. In addition to these proposals possible extensions to existing modeling systems are discussed in general.

Zusammenfassung

Die vorliegende Studienarbeit beschäftigt sich mit dem Thema der Modellierung virtueller dreidimensionaler Objekte und mit der Weiterentwicklung eines existierenden Modellierungssystems namens SESAME. Das Ziel is dabei Methoden zu entwickeln, welche die konzeptionelle Gestaltungsphase in einem allgemeinen Modellierungsprozess unterstützen. Aus diesem Grund liegt der Fokus der Arbeit dabei auf der Entwicklung intuitiver und ausdrucksstarker Deformationsmethoden, welche sowohl einfach zu erlernen als auch anzuwenden sein sollen.

Desweitern sollen sich die zu entwickelnden Methoden konzeptionell in das existierende System einpassen, um die Vielfalt der zu erstellenden Formen zu erhöhen. Zudem werden allgemeine Erweiterungenmöglichkeiten für konzeptionelle Designanwendungen und Modellierungsprogramme erörtert.

Declaration

This is to certify that I alone compiled and completed this thesis using only the resources allowed. Quotations and references are indicated accordingly.

Magdeburg, January 2008

Alexander Kuhn

Contents

1 Introduction

1.1 Motivation

Defining complex 3D geometric shapes within a virtual environment can be considered a challenging task. Basically it contains of two main problems: First finding an flexible and precise description for an object shape and secondly an effective scheme how to interact with this description. This challenge is often represented in the complexity of the tools designed and commonly used for this purpose. In most cases these large software packages offer an extended shape creation, modification and processing functionality and include modules for geometric modeling, texturing, animation and rendering. This fact already implies the requirement for a certain level of experience and an extended learning phase in order to work with these features effectively.

For this reason during the last decade the attention and amount of publications towards the topic of *conceptual shape modeling* has considerably increased. Among other systems the SESAME modeling system [SOD06] attempts to facilitate this process and to explore alternative approaches towards expressive conceptual creation of shapes. Provide access to efficient shape modeling tools to a larger group of users might also support the exploration of new application areas. Besides these goals efficient modeling tools can be used to improve the workflow during creational processes in practical environments and increase the number and quality of the created solutions.

1.2 Task Description

The aim of this thesis is to develop interaction techniques for 3D modeling tasks, based on concepts of an existing modeling system called SESAME (Sketch, Extrude, Sculpt, And Manipulate Easily).

These techniques should aid the conceptual design phase within a shape creation process. As such they are supposed to be used with minimal previous knowledge and low cognitive effort performing the modeling steps. Therefore the focus of these concepts lies on intuitive interaction, which includes the support of incomplete input data given by the user. Despite this lack of information, the results should match general expectations of the user towards the created shape.

The concepts are supposed to extend the variety of shapes which can be created with the original system, improve the interaction with them and should enhance the modification and creation of 3D objects. Additionally the proposed interaction methods might allow for efficient exploration of possible design solutions and should be independent from the actual object representation.

1.3 Structure

The thesis itself is divided in four chapters: First a general introduction into the terms and definitions of 3D modeling aspects and relevant applications is presented. This is followed by an overview of currently existing modeling approaches and a brief description of these concepts. The second part introduces the existing SESAME modeling system and the development, description and theoretical evaluation of the conceptual ideas that were conceived out of the preceding observations. This evaluation represents the basis for the third part, where selected concepts are implemented and discussed in further detail. The focus lies on the technical description of the selected ideas and identifying possibilities how to implement the concepts followed by a brief overview of important implementation aspects. In the last part, the results are discussed and evaluated in order to explore further extension possibilities and improvement possibilities. Finally, the thesis concludes with a list of references and sources which have been used.

2 Fundamentals and related Work

2.1 Development of 3D Modeling Techniques

Since the creation of the first graphical computer displays in the 1960s and the spreading of personal computers, visual applications for computers have gained significant importance. Besides Moore's law, which predicted the development of the computational capabilities, the actual growth of the capacities (especially in the graphics area) exceeded these expectations [OLG+05].

This fast progression expanded the possibilities to present and to interact with virtual content to a large degree, also increasing the prospects for complex visual computations and various interactions techniques. To deal with the huge amount of additional information, not only the interaction techniques themselves had to be improved and adapted, also the creation of geometric content for visual applications, namely three-dimensional modeling, gains more importance.

From this point the question arises: What is actually digital 3D modeling?

Digital shape modeling or 3D shape creation nowadays is mainly understood as a technical construction process, which is reflected in the way most common modeling applications are designed: Split windows with grids and precise numeric modeling tools offering a large variety of different modeling paradigms and high functionality. This complex development process is necessary to describe exact digital representations, as they are mandatory for subsequent rendering steps, construction or processing tasks. Common examples for corresponding software tools are given in section 2.6.

But especially in the first phases of conceptual design, this software structure is not very appropriate. The issues that are important in these first steps of dealing with initial ideas are intuitive interaction, creativity and conceptual ideas. They are often seen as highly individual aspects involving the process itself, as well as used tools, context and the creator.

Indeed, all these terms cannot really be described very precisely and objectively, which makes it difficult to define guidelines and concepts that fit the requirements of this creational stage. Despite these challenges, there have been several approaches addressing exactly this problem during the recent decades. Among pioneering systems, like SKETCH [ZHH96] and Teddy [IMT99], a number of promising approaches have been developed, getting closer to the goal to efficiently support conceptual design. One of these Systems is SESAME (Sketch, Extrude, Sculpt And Manipulate Easily, [SOD06]), which will be described in more detail within section 3.1.

The basic features of these concepts always correspond to the final practical environment in which they are used. In order to get an idea about important aspects, these applications have to be considered.

2.2 Applications of 3D Modeling

In general there are various application areas for 3D modeling software and 3D object representations:

- **Technical engineering** The digital representation of physical objects is a very important aspect during a construction process of mechanical parts and has multiple special applications within this process: First of all during the conceptual phase, simple models can help product designers and -developers to communicate about the shape and functionality of objects and already determine properties in early stages. During the next phases more concrete models are required, which are derived from the preceding conceptual ideas, but usually have to be recreated in CAD/CAM Modeling Software tools. In this phase planing and determination of spacial relations and shape properties are required at a high level of exactness to avoid complication or ambiguities during the following stages. Another transformation is required in the last production phase, when the model needs to be transferred in a representation, that can be used within the manufacturing process (e.g. for CNC machines). Some of these aspects are collected within this case study towards conceptual tools [Sin06].

- **Architectural design** Architectural representations are usually concerned with technical visualizations and spatial planning issues. As such the conceptual process has certain similarities to the product development process and can be divided in the same phases. Besides spatial properties, the context related visual appearance is essential, as it can be produced with digital scenes and corresponding realistic lightning models. Corresponding approaches are illustrated by the commercial Google SketchUp system[1] and available physics simulations modules.

- **Earth Science** Simulating the physical processes with virtual objects is also useful in this area, as well as abstract 3D representations for visualizations purposes. One example within this context is [CCH+02], emphasizing the shared availability of interactive 3D content for educational purposes. Furthermore abstract physical simulations can already be performed as soon as rough shape information is available. This might be helpful for conveying abstract coherence in complex systems, which are common within this context. Although there is no direct conceptual shape design required, the creation of adequate virtual content also requires efficient and intuitive modeling environments.

- **Game industry** Game applications usually make heavy use of 3D models and virtual content exploration. For this reason, they pushed the development of fast, interactive and realistic 3D visualizations over the last decade and also lead to the development of several industrial standards [OLG+05]. Most of the current modeling software packages are geared to the needs of this application area and often

[1] Google SketchUp, http://www.sketchup.com

represent the first broad range user studies of modeling techniques and environments. The support of conceptional shape modeling will especially also be useful for abstract shape design processes in in this area.

- **Movie industry** Like the Game industry, there is also a high need for conceptual and specialized modeling tools in this area. Usually development processes in this area include an extended conceptual design phase incorporating the creation of storyboard, set designs or even complete digital characters. The focus in this area lies on interactivity, but is directed towards realism or certain stylistic appearances.

- **Medical applications** Also within the medical science there are also several applications for 3D models: First of all the anatomy of the human body and special organic structures can be represented and explored in an virtual environment, which is especially useful for educational purposes or as additional data to support surgical planning. Although most of the models are acquired automatically as described in section 2.5 these methods usually introduce artifacts due to limited spatial and temporal resolutions (e.g. [HPSP01]). For presentation purposes[2] and the development of visualizations or anatomic structure detection algorithms, usually additional 3D content is created manually.

  More specific applications are techniques deforming anatomic models based on simulations of the physical behavior of organic objects during dissection and resulting haptic effects [RMT05]. Therefore flexible volumetric 3D model representations are also required.

- **Chemistry** The modeling of chemical compounds and internal reactions can help to abstract and visualize these complex processes and offer possibilities for quantitative visual analysis. This fact can be used for educational software applications in general. Therefore, efficient specialized modeling tools already have been proven to bee useful in this area ([LXJY05], [MTG+04]).

  The experiences made with conceptual design tools towards efficient virtual content interaction might also help to improve these specialized modeling applications.

This list can not be considered being complete and certainly many more application areas of applied 3D Modeling could be found. As already pointed out, in most of these application areas a large amount of specialized software tools have been developed for the last decades and the question arises if there are some common requirements for shape and 3D content creation.

2.3 Modeling in General

From this application oriented discussion, which outlines some basic practical applications, the initial question can be refined:

[2] examples can be found on commercial websites like http://www.3Dscience.com or http://catalog.nucleusinc.com/

What are common properties and requirements of 3D shape modeling and are there common aspects and improvement opportunities among these applications?

As also stated in [Sin06], despite numerous approaches, a simple answer to this question is hard to find. So in order to give a meaningful answer, it makes sense to discuss this topic from several different points of view.

2.3.1 Shapes in natural Environments

One possibility is to interprets the modeling process as an evolutionary development, which is characterized by two important steps: *modification* and *selection*.

Starting with a simple shape, by making slight modifications of its structure or adding new properties, a huge amount of different shapes is created depending on the type of modification. With the help of a selection pattern (so called fitness criteria) stated by different rules or constraints, after a limited number of these modeling steps the resulting shapes will approximate different solutions for the given selection parameters. As an important effect, there will be multiple different approximations, that all fit the quality criteria of the selection requirements (further information on this topic can be found in [BC02]).

One example how these kind of abstract concepts can be applied within modeling systems are suggestive systems. They propose a number of the modification steps and select the altered shapes either manually in suggestive interfaces like in Chateau [IH01], GIDeS++[3] [FCAD03] or automatic which results in techniques similar to procedural modeling systems [Wat07]. Procedural modeling systems are able to apply specified operations after defined patterns making far more complex shapes achievable than via manual shape creation.

Similar abstract shape creation processes could be derived from observing physical interactions of objects with their environment. One example would be gravity or behavior of different materials under applied forces. The human mind is familiar with a large diversity of these natural processes and are able to predict the results up to certain degree, but can also make use of them to create a variety of intended shapes. Typical examples would be sculpting with clay, drawing on a piece of paper or modeling with LEGO[4]. By learning and continuous interaction humans are capable to get essential skills in predicting the result of this interaction and in creating shapes.

2.3.2 Modeling as perceptional Transformation Process

On the other hand shape modeling can also be interpreted as a transformation process of perceptional information: This perceptional process starts with physical objects that can absorb, reflect or emit light. The resulting light intensity can be perceived by the human eye and is transmitted to the brain. Due to further processing differences in light intensity are interpreted and essential abstract features are derived.

[3] Sketch based CAD modeling system, http://immi.inesc.pt/ tfdc/gides++/
[4] Building toy produced by the LEGO Group (http://www.lego.com)

The goal of shape modeling, whether digital or not, can be understood as an inversion of this process. A similar, more specified formulation of this aspect has also been stated in [NSACO05] and [Sin06]. Starting with an abstract concept of shape and its encoded representation in the brain, certain modeling tools and resources (e.g., human hands, tools and clay, or a pointing device and a set of modeling operations) enable the shape designer to make a conceptual plan how to transform this abstract concept into an real or virtual three dimensional object.

As shown by [HS97] the human visual system also uses silhouettes as key index to memorize shapes. For further processing also the shading and texturing of the object resulting from its surface properties can be considered as being important [ZTCS99]. Within the area of non-photorealistic rendering (NPR) this definition has been abstracted towards suggestive contours by identifying lines that might get silhouette features in nearby viewing positions. So the recent approaches also directly try to use these encoding features for specifying 3D shape information and will be described in further detail in section 2.6.

By this point of view another important aspect has to be considered: The described abstract mental concept is diverse, enriched by individual subjective information, incomplete and can change dynamically during the modeling process. So finding an one to one digital representation of such an concept is impossible. In contrast the modeling process is also necessary to constrain, limit and objectify this representation to a certain degree, while over-constraining and limiting the creation process too much, will influence the structure and number of the final results. This consideration leads towards an approach, which is centered to the individual creator.

2.3.3 Modeling as creative Design Process

From the designers point of view, modeling can be understood as searching for an appropriate solution within a certain solution space. This solution space consists of all shapes that fit the designers aesthetical claim and the general requirements towards the shape of the object.

In some sense this also relates to the already described evolutionary concept of modeling given in section 2.3.1, with some important differences:

First the number of possible solutions that can be created and evaluated is limited and therefore much lower than those created with procedural techniques. Second altering and selecting possible solutions is not only oriented towards objective requirements, but also strongly depends on the experience and the knowledge of their designer.

Due to this lack of concrete information and creation rules conceptual design is also an iterative and dynamic process. After finishing one modeling step the designer evaluates the achievements and makes a new plan how to reach the desired shape or how to close gaps of missing information, which was not included in his abstract concept. This might also mean that the designer can reject solutions, add new aspects during the modeling and take single steps backwards by erasing and modifying certain parts of his conceptional design.

These preceding considerations already hint at some of the main conflict points while supporting this process with a computer. They also can be seen as some of the reasons, why traditional tools like clay and sketching still dominate conceptual and inductive shape creation [Sin06].

2.3.4 Supporting conceptual Shape Modeling

Although supporting this complex and dynamic creative process is a difficult task there are some key motivations which drive the development of software tools for this purpose.

As already stated most of the conceptual modeling is done with traditional media and parts of this work have to be redone in later stages to translate these into digital concepts. These precise numeric descriptions are necessary to fulfill further processing requirements and this conversion between the conceptual design- and the constructional data is often accompanied by the loss of already created information and redundant design steps within different descriptions.

On the other hand providing more flexible conceptual software, allows to get more control and further creation tools within the overall design process. Moreover it reduces unnecessary loss of information and hint at problematic configurations in early design phases. Thereby efficient tools can help to communicate about individual conceptual shape ideas in a more efficient way, by providing a more objective view on certain shape configurations. This can help to extend the range, diversity and quality of possible solutions. In addition to this also far more complex considerations, like physical simulations, can be introduced in earlier stages extending creative possibilities during this phase as well as opportunities for optimization towards later requirements.

A list of current approaches to tackle this problems is given in section 2.6.3.

2.3.5 Modeling as algorithmic Approach

Besides these more abstract and general considerations a formal definition of the term three dimensional geometric modeling can be stated as following:

In respect to create digital content on a computer the term 3D Modeling describes the process of creating or modifying a digital representation of a 3D object in an geometric form, in order to store, manipulate or process it for further use. This digital representation must contain a description of the shape of the object and its spatial properties, but can also contain further processing information (e.g., about texturing, rendering or animation). This modification process is often carried out with the help of a specialized 3D modeling software and results in a so called 3D model.

This definition already hints at the two main aspects of a modeling program:

The digital *object representation* and the definition of creation and deformation operations.

2.4 3D Model Representation

The main purpose of a digital 3D model is to contain information about the geometric properties and visual representation of a virtual object. Therefore a classification into boundary representations and volume representations is used to present some of the basic concepts. Which type of representation is chosen mainly depends on the application, but also on the availability of algorithms and hardware support. It must be pointed out, that this classification is only one possibility and cannot capture all the currently available approaches (an alternative classification into procedural descriptions and geometric primitive combinations is proposed in [Hav05], P.18.) Within the following section a brief overview of common representations and their properties is given.

2.4.1 Boundary Representations

Boundary representations often address applications with the main focus on the outer surface and visual appearance of the object. Therefore these representations contain data describing position, orientation, and additional properties of the object surface. Due to an intensive research there has been developed a variety of different representations that could fall into this category:

- **Polygonal Representations** Polygonal surface descriptions represent a discreet surface approximation of the object consisting of a set of connected polygons. The polygons itself is described by a set of points, which store the *geometric information*, and connecting edges, representing *topological properties*. Usually triangles are used, because they have a number of geometric properties, which make them ideal for further processing: They are always planar, convex, cannot self-intersect and are simple to compute. Furthermore every other polygon can be divided into a set of triangles and also every 3D shape can be approximated by triangular meshes, whereas in most cases the more accurate this approximation is supposed to be the more triangles will be needed (e.g., curved surfaces). Due to this properties this is the currently most widespread object representation technique and there is a large variety of algorithms and supporting hardware, that have been developed for processing triangular meshes.

- **Point based Representations** Point based representations store geometric information only within point positions (in contrast to polygonal descriptions, where shape features are also included in connecting edges). Further shape information is derived by interpolating these discrete sample points and additionally stored information. For displaying point based representations *splatting* can be used, which employs disks instead of triangles as rendering primitives. The main advantage of this approaches is the fact, that during the transformation of the model no topological information has to be maintained. This allows more sophisticated modeling operations, that usually do not depend on any connectivity information. Within the last few years there are also a number of promising approaches pointing out, that point based methods can keep up with triangular meshes in terms of rendering performance and flexibility. An overview of the state of the art in this area is given in [KB04]. Due to missing topological information the direct connectivity and determination of neighboring points is not given explicitly within the representation. Moreover most of the common algorithms for rendering and processing polygonal meshes will not be compatible with this representation, which makes intermediate conversion steps necessary.

- **Signal oriented Representations** Mathematically the description of a shape can be seen as function, defining a certain subset of the 3D space as belonging to the object surface or not. As known from the signal theory, every function in 2D as well as in 3D can be approximated by a combination of weighted basis functions. Therefore there a number of approaches using these descriptions for shape representation and manipulation. As an advantage from these kind of representations high-level properties can be derived much faster than from explicit descriptions (e.g., high frequency areas on the object surface which encodes surface features described in 2.3.2) and the knowledge and insights from signal theory can be applied. Therefore signal based representations can be very useful for storing and transmitting the final model (e.g., as illustrated in [BM03]). On the other hand direct interaction with signal based representations and applying local changes cannot be considered as very intuitive and the conversion to other standard representations takes specialized software and is usually associated with a loss of detailed shape information.

- **Implicit Surfaces** An object surface can also be directly described by an implicit mathematical function, defining a subset of the 3D space and dividing it in an inner and outer area. The strength of these kind of representations is high accuracy and compact definition for special cases (like spherical objects). Additionally they do not depend on the spatial resolution, consume less space than explicit descriptions and volumetric descriptions can be easily derived (additional references can be found in [Hav05], P.19). On the other hand it is a non-trivial problem to describe arbitrary shapes with the help of implicit functions, usually resulting in high degree and complex descriptions, which do not fulfill the requirements of efficient and fast interaction techniques anymore.

- **Constructive Solid Geometry** CSG is based on the assumption that even very complex and structured shapes can be described by combining simple primitives. Therefore a set of combination functions, so called boolean operators is defined which usually includes union, subtraction and intersection of objects. The result of the combinations is usually stored in tree-like structures for further processing and rendering evaluation. One of the first approaches towards interactive conceptual design aided by CSG operators is illustrated in [RS97]. With the help of this concept the advantages of implicit surfaces can be used and extended towards more expressive shapes without getting to complicated mathematical descriptions. Although there is a wide variety of shapes that can be produced, they are still not arbitrary. Reproducing or designing arbitrary objects would require from the designer to plan the construction process very well or a unacceptable high number of simpler shapes. One characteristic of this kind of representation is, that the shape information is not only stored within the explicitly representing primitives but also in the procedural description how to combine them with each other.

- **Subdivision Surfaces** These kind of representations are usually incorporated with polygonal meshes and the actual shape is derived out of a so called base mesh and an applied subdivision scheme. Like polygonal representations, nearly every shape can be approximated and the results usually offers smooth surfaces depending on the kind and repetitions of the applied subdivision scheme. One important property is that the actual model is derived by a combination of explicitly giving geometry data and an implicit transformation function reducing the amount of space required for this representation depending on the complexity of the model. Additionally the shape information created of the subdivision algorithm can be influenced and combined with procedural shape generation [LVKY02]. Although usually yielding nice results, smooth surfaces are not required in every case and for complex models the direct interaction with the control mesh cannot be considered as effective anymore.

- **Parametric Patches** Basing on the 2D Counterpart of Higher Polynomials, like Bernstein Polynomials and B-Splines, 3D Surfaces can be described by combining multiple boundary splines to one surface represented by their corresponding control point network. The properties of so called freeform surfaces are high flexibility towards the approximation of shapes and the local intuitive behavior towards interaction with the control points. Although controlling B-Splines in 2D are very expressive and can be used to describe nearly every 2D shape, this process cannot be transferred to 3D that easy. The additional degree of freedom results in a lot more controlling parameters, similar to the base mesh of subdivision approaches the control polygons can get very complex and problems due to occlusions have to be resolved.

2.4.2 Volumetric Representations

Besides pure boundary representations also a variety of volumetric shape descriptions do exist. These descriptions are usually chosen, when inner parts and structures of the 3D model are important, like in medical applications or physical simulations. Some examples of this class of representations are:

- **Implicit Surfaces and CSG** As already stated for boundary representations these kind of representations also indirectly include volumetric information about the object, which can be used. Despite this easy derivation they share the same disadvantages as their surface representation counterpart.

- **Voxel based Representations** As it is done with pixels for 2D images, objects in 3D space can also be rasterized and defined by simple quadratic primitives so called cells or voxels. They usually store information about material, opacity and various optical or physical properties. As an advantage these kind of representations directly store the object information within their spacial context and information about object properties in close regions can easily be derived. Hence this is often used in complex optical simulations (e.g., raytracing). On the other hand they are characterized by high memory consumption and computationally more expensive processing of the data, which makes further optimization necessary. The application of these models is discussed in further detail in [CA06].

- **Space Partition Techniques** Of course the 3D volumetric representation is not limited to quadratic primitives, but can also be given by a set of other primitives like planes dividing the surrounding space into two subspaces. Usually this optimizes the memory consumption and size of the object representations, but also makes interaction and deformation of the object more complex towards its algorithmic description. One of the first attempts in using space partition techniques in virtual environments is described in [Nay95].

In summary the choice of the object representation usually depends on the application and processing requirements. Note that the classes given as example cannot be considered as being disjunct and most of them can be combined or transformed into each other, although this transformation is usually associated with the loss of data, depending on properties of the descriptions.

One major issue of most of these representations is the lack of semantic information stored with the object representation. Although there are often additional information about surface properties or texture, there is usually no information about the functionality of a certain part within an object and usually has to be added in later stages. In technical applications and major CAD systems there are shape descriptions, which are coupled with semantic information (so called constraints) but there are no suitable exchange standards. In most cases this additional data can only be used within the system the object was modeled in as stated in ([Hav05], p.8).

Another common problem among recent modeling approaches can be seen in the fact, that deformation methods and modeling approaches are often derived directly from the actual description, which is discussed within the next section.

In order to successfully create suitable tools for conceptual shape creation often aspects of multiple shape representations have to be considered.

2.5 Digital Shape Creation and Deformation

Once a digital representation for an 3D object is found, the question is how to interact with this kind of description.

There are two basic operation classes: the *creation* of new geometric content from the scratch and the *manipulation* of shape and additional information of existing objects (e.g., texture, surface properties). In general there are also two possibilities how to create

3D geometric content: automatically acquiring it from physical objects or by creating the models out of various input data with the help of specialized software tools.

In practical application the majority of the automatic acquisition methods are based on optical methods like Laser Range Scanning. Further possibilities are reconstructing the 3D object out of a series of 2D images (Photogrammetry). Specialized methods are also used in medical and technical engineering applications, which are using magnetic resonance imaging (MRI) scanning devices as stated in [BP07]. An important aspect is, that this data contains noise out of measurement errors depending on the properties of the acquisition system as well as temporal and spatial scanning resolution. Therefore the resulting very large data sets have to be processed in order to identify important surface features. Additionally these models neither contain any semantic information or about their construction process.

Creating 3D shapes manually with the help of special input information forms the second class of approaches. These input data can include position information by pointing devices or any other kind of data transforming the content into a 3D representation (e.g., optical tracking information like in [SBS06] or any other scalar signal). The efficient mapping of this input data to an intuitive modeling operation, in form of a geometric modeling technique, is a non trivial problem offering a wide variety of possible solutions.

The main focus in this work is at interpreting input information from common pointing devices like a mouse or a digital pen and apply them on specified modeling operations. As already mentioned these operations can either start from scratch and directly create new 3D objects or can be used for editing already existing geometrical objects to transform their shape.

2.5.1 Surface Deformation Techniques

Geometrically deforming a surface can be defined by a transformation function mapping an original surface to a modified version. In contrast to global procedures local deformations are limited to an area of influence wherein different points on the surface can be affected in several ways. Therefore constraints can be modeled and usually are defined by the user to modify the shape. Furthermore the variety of shape modification techniques can be classified in a similar fashion as the object representations within the previous section. Additionally it complies with the overview given in [BP07]. During the last decades a number of different techniques have been developed:

- **Tensor Product Spline Surfaces**

  Motivated by a spline representation for objects interacting with this representation already implies one class of deformation methods. A surface is directly constructed or approximated by a set of spline functions and the direct interaction with them already yields deformations. The effect of these modifications is usually limited towards very smooth surfaces due to continuity constraints and a rectangular region of influence. As already stated, complex deformations raise the complexity of the control structure and therefore hamper intuitive modeling. Additionally also the number of control points is crucial for representing surface details and to avoid deformation artifacts.

- **Transformation Propagation**

  Based on the deformation description by means of a function this transformation is usually limited to a certain area, the so called support region. This area can be directly limited to a certain region of interest on the object surface and controlled with a second defined region usually called the Handle. Expressed in numeric values the amount of deformation within the handle region is 1 and equals 0 at the border and outside of the support region. Between those regions the values are determined according to the corresponding interpolating function. The fact that this function usually only depends on a scalar distance field can lead to unexpected results and smooth surface interpolation cannot be guaranteed.

- **Variational Energy Minimization**

  To avoid unexpected changes within the surface description global properties (e.g., surface area or curvature) of the shape can be used to determine the properties of the deformed region. This global minimization is inspired by real physical interaction of a thin elastic surface with applied forces and certain boundary constraints. Usually the aim of these minimization approaches is to reduce occurring bending and stretching energies on the object surface. One shortcoming of this approaches is that normally local details are not preserved due to the global optimization. Global transformations do not result in local rotation of details on the object surface, which might result in visual artifacts or unwanted deformations of distortion of local features.

- **Multiresolution Deformation**

  The key elements of this techniques are shape decomposition and reconstruction. Therefore the sampled surface is decomposed into an set of different subversions with varying surface detail in means of a coarser, usually smoother description with less details by applying operations like mesh smoothing or -fairing to polygonal surfaces. The deformation itself is then applied to the so called base mesh, with the least number of surface details. After the deformation, the mesh details are reconstructed on the deformed mesh, reconstructing the original surface details. Usually the representation of the displacements, that occur during the decomposition is crucial for the overall result. This kind of techniques has its limitations especially for large mesh deformations, where the displacement of the surface details is larger than the local curvature, which results in self intersection of the surface and unintuitive results. Besides numerical stability cannot be guaranteed anymore.

- **Differential Coordinates**

  As already stated a large amount of the shape information is coded within local properties of the object surface. Mathematically this change of the local surface behavior depends on the derivatives of the function, that describes the surface. The key idea of this approach is to transform the surface while maintaining the secondary derivative of the original shape. The constraints formulated by the derivatives can be used to restore local detail, that would have been lost otherwise. This usually means to solve a set of linear equations, which can be represented in different ways. The advantage of these kind of transformations is, that they can also deal with large deformations on quite complex models but can also yield unexpected results especially for large rotations on the object surface.

2.5.2 Space Deformation Techniques

A similar development can be observed for space deformation techniques. Due to their higher complexity, they are more likely to be used in special applications like realistic rendering, interacting with volumetric medical data [RMT05] or digital sculpting techniques [CA06]. Most of them represent extensions of the surface based approaches, often also sharing their limitations. The main difference is that in comparison to the surface based techniques the space around the object is transformed. This is resulting in an important property: The deformation itself does not depend on the object description anymore and can be applied to any object representation.

- **Freeform Deformation**

  Similar to spline surface representations, polynomial curves can also be used to describe a 3D space patch. Despite the spatial description and extended shape variety, the additional dimension usually makes the interaction with the given control structures even more complex. Therefore direct manipulation is usually is not feasible anymore and more abstract procedures are required.

- **Transformation Propagation**

  With the help of the euclidian distance field deformations can also be described within a higher dimensional space with the difference, that the support region is a 3D subspace and the interaction influence has to be interpolated over a volume. In analogy to the 2D case in most application an additional global energy minimization produces more predictable results. This concept often is incorporated within sculptural modeling tools as described in section 2.6.

- **Non-Linear Space Deformation**

  Geometric modification problems are usually described and computed using a linear equation system depending on the complexity, consideration of global shape properties and number of parameters of the deformation. There are also more complex techniques, especially global energy minimization approaches with multiple degrees of freedom, which take properties of the surrounding geometry into account in order to improve the result or to create context related and more intuitive deformations. In general a larger computational effort is required, but theses procedures also yield better results in terms of intuitive parametrization and surface quality.

The described techniques only represent a general overview about the existing approaches, which have been developed. Designing a conceptual design system often requires the incorporation of multiple and specialized representations as will be shown within the next section.

One general demand for deformation concepts should be the independency from the object representation for several reasons: On the one hand a general description can be applied to several different mathematical descriptions in different systems and should only depend on the existing shape parameters. In the ideal case a modeling technique should yield the same results for all representations, which in general cannot be fulfilled. On the other hand standardized modeling operations can help to develop and improve modeling systems towards their intuitive use and required knowledge in order to produce the intended result in a more efficient way. This will also support the further deployment of more abstract high level deformations. Finally new deformation concepts do not have to be developed multiple times in order to apply them to different representations.

2.6 Existing 3D modeling Systems

The amount of application areas, representations and shape modification methods already implies the requirement for different and specialized software for various purposes. For this reason there are multiple modeling approaches, utilizing different deformation paradigms and object representations shown in chapter 2.5.1 and 2.4 for interacting with three-dimensional content.

Before creating modeling approaches addressing the conceptual design phase it is necessary to get an brief overview over the main approaches and common systems to identify drawbacks and positive properties, which are useful to be avoided or incorporated into the conceptual tools and to derive guidelines for their creation.

2.6.1 Modeling Packages and technical CAD Modelers

These large Software packages like 3D Studio Max [5], Maya [6] and Blender [7] usually contain a very high functionality, that is not only limited to the modeling process itself but also contain modules for texturing, animation and rendering. Often theses programs include various modeling techniques in one software package and the software supports internal conversion between different object representations in order to apply different modeling operations.

An important property of the majority of these tools is the separation between the actual shape creation process and the viewing of the object. Although there are usually small rendering windows they are used for previewing the spatial relations and evaluating

[5] Autodesk, www.autodesk.com/3dsmax
[6] Autodesk, www.autodesk.com/maya
[7] Open source 3D content suite, www.blender.org

the surface quality as shown in figure 2.1. The main reason often is the complex rendering process and the initial orientation towards technical construction purposes. Although all packages usually support the personalization of the interface, the time spend for menu operations (especially among users with lower experience) is considerably higher than the actual time spend for modeling operations. This will be shown in section 3.1.

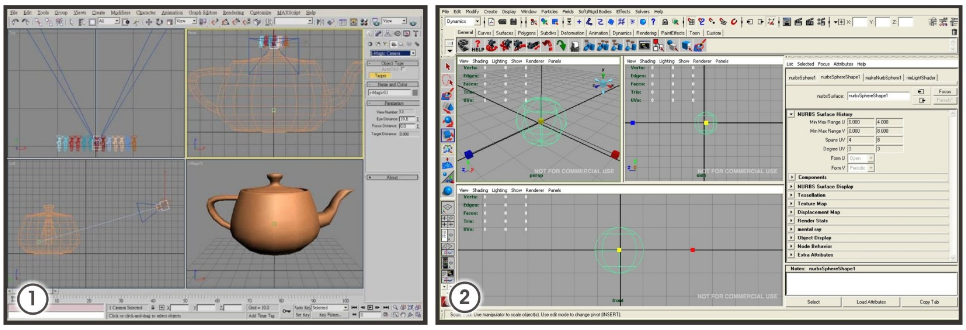

Figure 2.1: Examples for the standard interface of 3D Studio Max(1) and Maya(2).

Technical Computer Aided Design modelers like CATIA [8], Pro/ENGINEER [9] or SolidWorks [10] are usually oriented towards the later construction of the object. Therefore aspects which are important in later production stages already influence early modeling processes. These first stages are in most cases still very close to the traditional approach, using technical drawings from multiple perspectives to resolve constructional ambiguities. This also means that construction related information is created in early stages, including information about the functionality of an object (e.g., movement constraints, structural dependencies), construction trees to track the development, version creation and authoring functionality in so called EDM/PDM Systems.

Especially these technical construction packages are often designed as stand-alone systems, supporting the complete creation process. Therefore the exchangeability and common data format support is rather weakly developed. This means, that during the conversion of data for further use in external systems additional semantic information usually gets lost.

Furthermore all of the referred modeling tools demand a high levels of exactness and completeness of the input data during the whole creation process. Working with surface representation this often means detailed editing of single control points or surface patches even for very complex models.

[8] Applied PLM, www.appliedgroup.co.uk
[9] PTC, www.ptc.com
[10] Dassault Systems, www.solidworks.com

2.6.2 Sculpting and Volumetric Modelers

Besides technical oriented modeling systems during the last decade also various volumetric modeling systems have been established, too.

Some of the main systems like ZBrush [11], MudBox [12] and volumetric modeling modules in Blender use so called volumetric sculpting metaphors originating from abstract observations like in chapter 2.3.3.

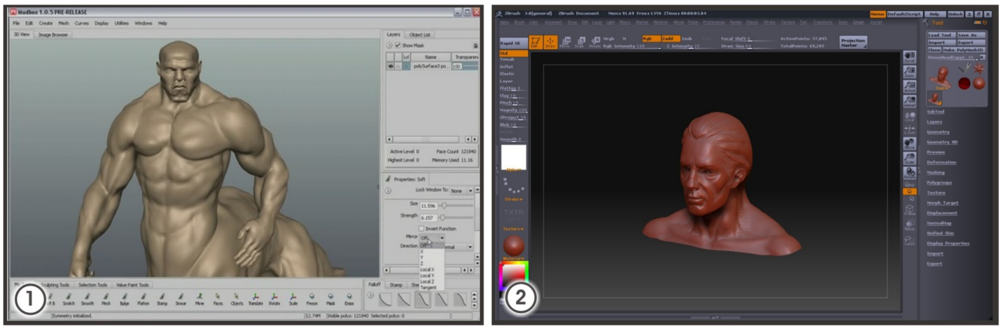

Figure 2.2: Example interfaces of two volumetric modeling interfaces namely MudBox(1) and ZBrush(2). (The screenshots and models are taken from software tutorials)

The main modeling operation has its origin in the modeling of real clay and usually consists of the local displacement of the surface within a circular support region following a transformation propagation concept described in section 2.5.2. Therefore various displacement behaviors can be defined depending on the given input and usually mesh refinement or space subdivision techniques are incorporated. Most of these tools are used for creating smooth and detailed organic shapes for example in the area of modeling organic structures and character modeling, but also to support rapid shape prototyping. A short review over some of these techniques and their application in conceptual design is given in [CA06].

The main advantage of these systems is the low number of operations that is needed to create a large variety of shapes within a reasonable amount of time, the resulting lower complexity of the user interface and usually no separation between creating and viewing the model, which can be seen in figure 2.2. Due to their volumetric representation most of these systems also support CSG like operators described in chapter 2.4.

Furthermore the deformation concept itself often does not depend on the actual surface representation as it is just defined in means of a displacement vector field and allows the conversion into traditional polygonal surface models. In addition to this, there are approaches which offer a high numeric stability and shape consistency towards extreme

[11] Pixologic, www.pixologic.com
[12] Autodesk, www.mudbox3d.com

shape deformations as shown in [AMPC+04] and [vFTS06].

On the other hand there is still a gap between the capabilities and formal expressiveness of these volumetric tools and the more technical oriented modeling approaches. Usually the modeling process is separated by creating initial parts of the model in a technical modeling system and refining the converted result in a volumetric system or vice versa. Despite most of the volumetric modeling tools offer complex deformation possibilities it is often hard to get precise control over the appearance of the result and usually this is achieved by a high amount of modeling operations.

2.6.3 Sketch Modeling Software

Strokes are already known quite well in the conceptual design phase as most of the work created in this phase is contributed by traditional sketch on paper artwork. For digital interaction the paradigm of input strokes is an flexible and versatile tool as systems like SKETCH [ZHH96],GIDeS++ [FCAD03], Teddy [IMT99], ShapeShop [SWSJ05] and FiberMesh [NISA07] have proven.

Most of the systems make use of the concepts stated in section 2.3.2 and utilize the definition of contours and feature lines as main input for the shape reconstruction which is often done by global energy minimization approaches like described in 2.5.1. Despite this complex internal representation most of these systems can be controlled by a very simple user interface accompanied by a set of command gestures. Some of these interfaces are shown in figure 2.3.

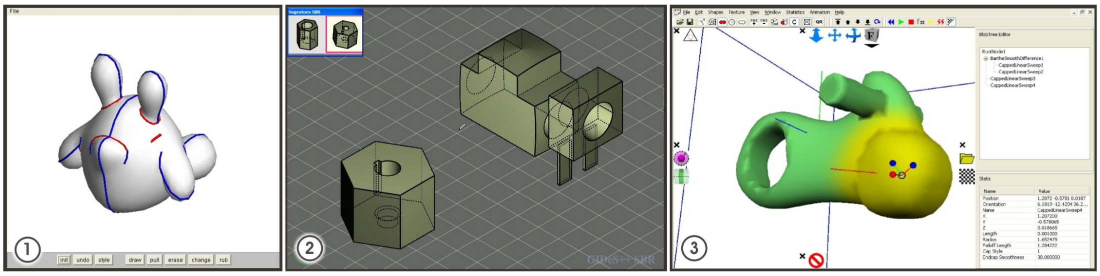

Figure 2.3: Sketch modeling interfaces highlighting sharp features (red) and smooth surface-curves (blue) in FiberMesh(1), suggestion based CAD sketching in GIDeS++(2) and CSG like sketch-model assembly in ShapeShop(3)

The main limitation of the first two systems stated in the beginning is the limited shape variety, complexity and appearance of the shapes that can be created. Additionally the underlying structural representation often influences the modeling operations which are available and the way they can be used. This refers to the mainly analytic, technical shapes in SKETCH [ZHH96] and GIDeS++ [FCAD03] and very smooth spherical shapes in Teddy [IMT99]. This problem is addressed by ShapeShop [SWSJ05] by enriching the formal expressiveness by a number of different modeling methods and an underlying Hierarchical Implicit Volume Model, which allows CSG-like modeling operations to achieve more complex shapes and to store the complete construction history of the object.

But still the range of shapes is limited and offers no smooth transition between the defined modeling operations. To tackle this problem FiberMesh [NISA07] has proposed another promising paradigm by defining on-surface strokes which might be smooth on surface lines or sharp edge features.

One alternative way of creating 3D shapes out of 2D sketches is by extruding 2D sketches like in SESAME [SOD06] and SketchUp[13]. Using intersections and resulting shape information given in a 2D drawing these structures are extruded along the surface normal of the drawing plane. This concept will be described in further detail in section 3.1. Another experimental system basing on this approach is CB Model Pro [14] allowing the extrusion of single freeform strokes towards smooth surface descriptions and fast shape exploration via appropriate parametrization controlled by sliders.

Despite some limitations the variety of the created surfaces in relation to the required system knowledge, speed and efficiency during the shape creation and exploration process outperforms larger modeling packages. Additionally they offer the possibility to combine properties of volumetric deformation methods and technical constructions processes into one interface without the need to shift and convert between different systems.

2.6.4 Modeling using specialized Input Devices

One of the main problems of the shape modeling tools is the reconstruction of 3D geometric information out of the given 2D input data which usually is created by common interface devices like a mouse or digital pen. To avoid this problem, there are a couple of systems that explore possibilities to use alternative input devices to facilitate the modeling process. Among other systems, examples are FreeDrawer [WS01] and Bender [LPRS05]. In FreeDrawer strokes can be drawn directly into the 3D space on their actual position within an virtual environment, whereas Bender uses two separated input devices to track the position, orientation and their relative position to each other. The resulting translations are linked to corresponding handles on the object surface to define local deformations.

Another interesting approach is SCULPROX [SBS06] incorporating various deformation techniques which are controlled by interacting with an physical object (in this case a sponge and attached optical markers), tracked by an optical system and applied towards previously defined surface handles. Two approaches are shown in figure 2.4.

A couple of systems is also working with the help of a combination of optical and haptic tracking as input parameters that can be mapped onto deformation operations like iSphere [LS04] and [HQ02]. An alternative way is to use a simple modeling blocks like known from LEGO as input to acquire spatial information as done in this approach

[13]Google SketchUp, http://www.sketchup.com
[14]Dassault Systems, www.cbmodelpro.com

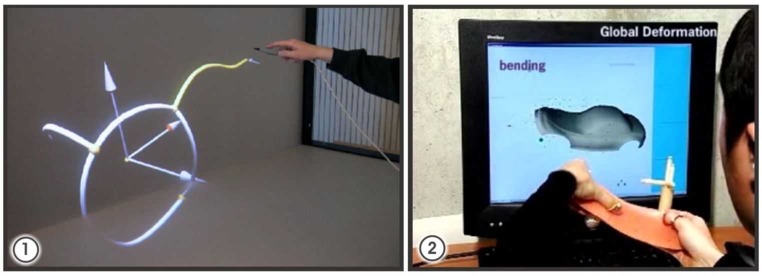

Figure 2.4: Alternative modeling input like a tracking device within a virtual environment in FreeDrawer(1) or tracking visual markers on various input tools used for simulating haptical feedback in SCULPROX(2).

[AFM+00]. In this publication it has also been shown that certain configurations of input commands can be interpreted automatically to construct more complex structures in an easy and efficient way.

Especially in the further development of sketch systems and conceptual modeling tools alternative input devices are definitely an interesting option for further improvements although the focus in this work mainly is on commonly available devices.

3 Conceptual Design of Interaction Techniques

3.1 The existing SESAME System

The SESAME System (for Sketch, Extrude, Sculpt And Manipulate Easily) was developed by Ji-Young Oh at York University under supervision of Prof. Wolfgang Stürzlinger ([SOD06],[SOD05]) and is the basic system for the extensions presented within this thesis.

The aims of the SESAME system are to support especially early concept design phases of the shape modeling process and facilitate the efficient exploration of possible design solutions. Originally it is oriented towards analytic shapes as they can be found in the area of product development or architectural design. Additionally it tries to reduce the initial knowledge, which is required in order to produce expressive and creative 3D models and to focus the work with the system on shape exploration and support a creative modeling process. The basic interface of SESAME is shown in figure 3.1.

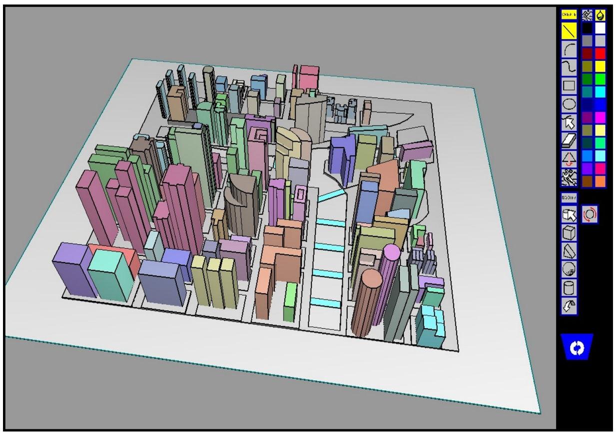

Figure 3.1: Example scene modeled within the SESAME system.

3.1.1 Development Guidelines

In order to implement a system, that fits the demands of this early conceptual phase, a list of general guidelines is presented, which provides a basic orientation of how well modeling tools can be incorporated in this process:

1. **Non-intrusive Interface**: Due to the limited cognitive resources of the user, an appropriate modeling system interface should be easy to use with a low cognitive demand and support the user to efficiently externalize his ideas. Often the modeling process can be reduced to a limited amount of efficient and expressive modeling operations.

2. **Easy creation**: As the design process is usually iterative, dialectic and cyclic as stated in section 2.3.3, it is important, that the system supports efficient and repetitive creation and manipulation of design solutions. Therefore interactivity and plausibility of these methods are essential.

3. **Easy combination and restructuring**: As studies [VH98] have shown combination and restructuring are two important techniques during the process of creative invention and thus should be supported by a modeling system. While combination of shapes is a rather simple activity, restructuring is a more complex process and usually requires externalization or the aid of a computer system.

4. **Tolerance towards ambiguity and incompleteness**: Especially in the conceptual phase a clear idea of shape is not existing. This is problematic for a digital representation within a computer, because they usually demand a high level of exactness. Therefore assumptions have to be made in order to produce a result, but it also has to be clear *where* these assumptions were made and how they influence the resulting shape.

5. **Range of levels of abstraction**: During a design process usually there are various detailed versions of one object to emphasize different aspects. So it might be useful to reduce or add details to an object in order to focus on different features. Abstraction facilitates its interpretation and the decision-making process towards features of the object during its creation. A system supporting this aspect should therefore possess the ability to change and present the object on various levels of detail and should offer fast access to these presentations.

6. **Ability to edit various forms of information**: This aspect refers to the way objects and transformation methods are represented as well as to other information that could be added to facilitate the creation process and might also support the interpretation of the final result. Additional semantic information can also be used in later design stages to determine functionality and parameters of certain parts, but therefore standardization of exchange interfaces is essential.

7. **Supporting evaluation (simulation)**: Creating an object also means repeated testing against certain design criteria which often means simulating or to 'think through' the result in its final environment (e.g., behavior scenarios, optical or physical aspects). Usually this testing is performed as a mental process and becomes more concrete in later stages. At this point a computer can already support this process by offering a framework for simulating different environment influences on the produced concepts.

These guidelines can help to evaluate the usefulness and applicability for conceptual modeling tools in general and point out where the focus of these tools should lie on. They also represent guidance principles for concepts proposed in section 3.2.

3.1.2 Modeling Interaction

In order to efficiently produce 3D shape representations in early design stages SESAME implements a *sketch reconstruction* approach using a sketch extrusion paradigm. Therefore the whole construction process is split up into 2D and 3D interaction techniques implementing a smooth transition between these two interaction methods during the design process.

- The **2D interaction mode** offers tools for sketching directly on the objects surface and gives visual support information (suggestions) during the sketching process. In this mode the user can efficiently define segmented areas which can be used for later interaction as shown in figure 3.2 (1). Within the interface the user can choose between a given set of 2D primitives (lines, arcs, circles and freehand curves) and use them to draw directly on any surface plane within the scene.

  While drawing suggestions are offered basing on geometric properties which are likely to occur for example parallelism and connections to intersection points. The number of the given suggestions is reduced further by measuring the distance to the current drawing location. To support the user perception and prevent misinterpretations due to perspective distortions, augmenting primitives are displayed in form of circles during the drawing process. This has been proven useful for estimating spatial relations in projected 2D drawings.

  After finishing the drawing step, the system automatically clips the line segments against each other and performs an automatic closed segment detection. Furthermore a snapping technique is implemented basing on already drawn 2D segments. The resolution of the snapping points corresponds to the current zoom level, which means the closer the virtual camera comes to the object more detail is provided.

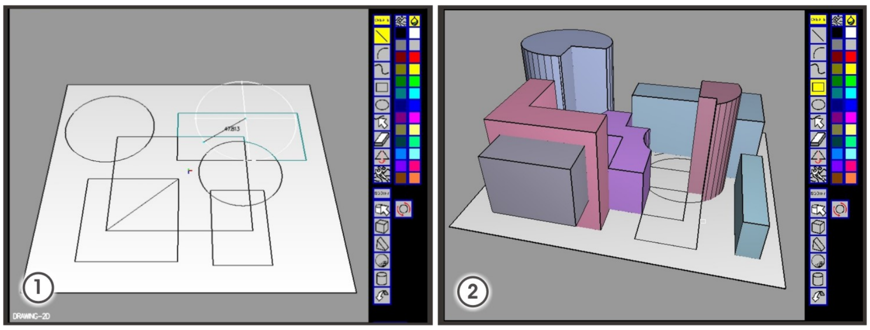

Figure 3.2: Modeling steps in SESAME. Sketching directly on the surface (1) followed by the extrusion of intersected areas (2)

- In **3D interaction mode** fast creation of volumes out of the 2D content is supported and abstract recombination and restructuring methods support the efficient interaction with the created content. After drawing in the scene providing closed segmented 2D geometric elements, the user can extrude them into the direction of the surface plane normal, the curves where drawn on as shown in figure 3.2 (2).

  For interacting with the extruded solid volumes the user can select them and move them around within in the 3D scene. This movement is constrained by snapping the object to the backward plane, viewing from the current camera position. The result is a smooth and predictable movement of the object and prevents free-floating in space and intersections with other objects.

  Once multiple objects are created automatically the so called contact graph is created. Within in this data structure all the objects are represented in a gravitational hierarchy and their corresponding contact relation. If the user selects one object in this hierarchy all objects on top are selected, grouped and further interaction is carried out on all of them (e.g., a table moved towards the room also moves all the objects placed on this table). The 3D mode provides a snapping method for other scene objects too, depending on the current zooming level, in analogy to the 2D interaction.

All of the previously described interaction techniques in both dimensions are available using a two button mouse and the shift and control buttons on a common input devices or directly within the graphical interface.

3.1.3 SESAME Comparison

In order to compare the usability of the described systems within the conceptual design phase, several user studies where performed and the program was compared to a common Modeling package (3D Studio Max [1]) and to traditional sketching on paper.

According to [SOD05] the studies where focussed on different aspects:

1. The amount of time that is spend to create conceptual solutions
2. The possibility to explore different solutions in the design space
3. How far does SESAME support creativity in the conceptual phase
4. The expressiveness of provided modeling operations

To rate these aspects among a practical task, in the studies the users where told to create an architectural model for a typical urban environment and rated among the given criteria.

It's important to note that due to the size of the study and the development state of SESAME, general conclusions have to be drawn carefully, but can get confirmed by observations made in similar systems like ShapeShop [SWSJ05].

[1] Autodesk, http://www.autodesk.com/3dsmax

According to the evaluation done in the study there was no significant difference in values representing the creativity, quality of the results, but practicability was rated slightly lower. In general it could be observed, that due to the direct inclusion in the content, the participants showed a better understanding of scale and context inclusion and the resulting solutions where designed less abstract. Additionally working with SESAME the users focussed on the modification of existing shapes more than experimenting with sketches and resulting shapes. It is also stated that working with the system can help to reflect the design problem under different circumstances.
Compared to the commercial CAD Software tool it can be said, that in general the SESAME allows more creation and modification operations in less time, which results in more productivity during the conceptual phase. Instead of dealing with mode-switching and menu operations this allows the user to create more content and also to explore more possible solutions. Furthermore the participants made extensive use of the drawing operations, in contrast to the lower than expected number of applied 3D interaction methods. On the other hand especially in later design phases a higher level of exactness and functionality is necessary, which is provided by the commercial packages, therefore exchangeability of the produced 3D content is an important factor.

3.1.4 SESAME Discussion

As seen in the previous section tools like the SESAME system definitely support and speed up the conceptual design phase. Together with its implementation and evaluation a number of important development issues already get clear and give useful hints, which features are important for such tools.

On the one hand there are several aspects which can be considered as being advantageous: First of all the performance of the SESAME system in the conceptual phase in terms of fast and creative content creation and design exploration outperforms common CAD modeling systems significantly. This is explained by its low interaction overhead and efficient interaction methods. Also the modification and reusability possibilities of the produced results in comparison with traditional sketching is much better. Having a digital representation in this early stage of the shape development offers also various advantages towards possible simulations of lightning conditions or physical properties and overall processing of the data.

On the other hand there are also some issues which are in need of further improvement: The current system does not provide any tools for numeric input or exact reproducibility of results, which might not be that essential in the early conceptual phase, but especially in later phases this aspect will gain significant importance. Therefore the system should at least support the exchangeability within standardized object formats in order to perform further processing with more specialized tools.

In addition to this the diversity of shapes that can be created with the system is limited. Up to now it is oriented towards architectural or technical domains, but in the majority of the applications a bigger variety of achievable shapes is desired. Additionally the technique in its current implementation is limited to certain object representations, although it might be extended towards a more general approach.

Another very important aspect is the visual representation of the created objects. The current rendering implies quite fixed and static solutions of ideas which are actually not that concrete in this phase. As stated in [SOD06], there already have been attempts to incorporate non-photorealistic rendering techniques. It turned out that the applied method of enhancing and stylizing object borders produces problems towards the expected result of selection operations. Additionally this influences the perceived accuracy of these operations in a negative way. Therefore further development in this direction seems to be promising to represent the conceptual character of the solutions.

In summary the SESAME system provides essential insight into the structural requirements for conceptual modeling tools and concepts. Together with the development guidelines given in section 3.1.1 the requirements for the new techniques are outlined and specify important aspects.

3.2 Modeling Concepts

To expand the functionality of SESAME basing on the preceding assumptions, there will be proposed a list of concepts in order to discuss them and select a smaller portion for realization. These concepts, which have been developed during the internship, are categorized in 3 classes:

1. *Interaction techniques* for enhancing virtual interaction with the scene
2. *Modeling using object interaction*, using existing objects and their relations to describe deformations
3. *Sketch based deformations* incorporating strokes as primary input data

Within their description, also applicability and usefulness towards the guidelines given in section 3.1.1 is rated.

3.2.1 Interaction Techniques

Besides the existing 3D interaction techniques in SESAME incorporating further tools can be useful to facilitate the modeling and design exploration process.

A **Automatic Adjacency**
This concept addresses the problem, that in SESAME no objects can be constructed floating in free space due to the gravity interaction paradigm described in section 3.1.2. Therefore effective later alignment methods are important. If the user wants to construct a larger object on top of a smaller one, he has to construct it else where, move it to the designated position and rotate it into its optimal position. Often this

position corresponds with the alignment of the object towards the corresponding surface normal of the smaller object as shown in figure 3.3.

The same action can be performed by defining one face on the objects surface and one on the target surface and perform an automatic alignment of the object. This concept could also be integrated directly in the 3D interaction mode, by aligning the back facing sides of the object towards the normal orientation of the surface occluded by the object. The structural information given during this operation can also be used in further interaction by constructing movement constraints out of the given information. Similar concepts (so called constraint based modeling) can also be found in CAD/CAM Modeling software, but often gets lost in further export steps with other programs, due to the lack of standardized file formats, supporting this kind of extra information.

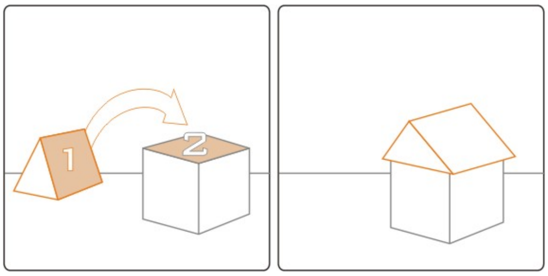

Figure 3.3: Defining surface 1 activates an object, while selecting surface 2 afterwards automatically rotates and translates the selected object to fit the adjacency condition for both surfaces.

B **Virtual Plane drawing**

Another more general problem while interacting with 2D devices in 3D environments can be seen in the ambiguities, which are created due to the projection. For example drawing a line on a 2D screen produces an infinite number of possible corresponding lines in 3D space. Especially drawing lines and 2D figures like in SESAME requires the definition of constraints to reduce the number of possible results. One way to accomplish this, would be the introduction of a reference planes for the direct drawing of strokes within the 3D scene. In its current implementation the drawing range of lines is limited towards the size of the underlying surface. So one extension possibility is to enlarge this area and indicate the additional drawing space by displaying a semitransparent plane within the 3D scene as can be seen in figure 3.4. The orientation of this plane can be determined by the surface normal of the reference object surface.

C **Cloning Patterns**

On physical object surfaces in general various repetitive patterns can be found.

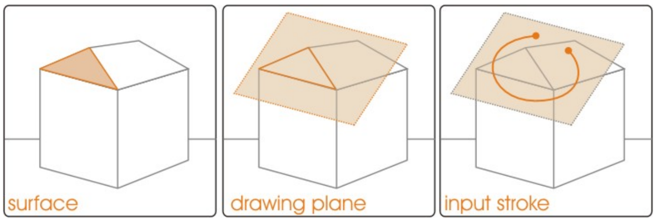

Figure 3.4: By activating an object surface a semitransparent 3D drawing plane is visualized within the scene to aid the sketching process on the defined plane.

Usually every single pattern has to be modeled separately, than copied and included in the existing geometry, which is depending on the structural complexity a time consuming task. In most technical CAD/CAM System the repetitive patterns are already incorporated in the development of technical shapes and there are approaches([ONI06]) applying copy and paste patterns for fast combination of already existing geometry. On the other hand effective cloning and repetition patterns offer essential advantages in modeling speed an compact representation as provided by procedural modeling techniques [Wat07]. By reusing and parameterizing single operations during manual object deformation and creation additional details as well as shape variations can be introduced. Cloning patterns can also be utilized in earlier modeling stages by defining a modeling area and additional cloning region with an corresponding parametrization function. The modeling operation itself is performed once in the original area, can be directly transformed into the new parameter values and applied on the copy regions. Additional parametrization of the copy areas can also influence scale and spatial parameters of the copied results. This concept is shown in principle in figure 3.5.

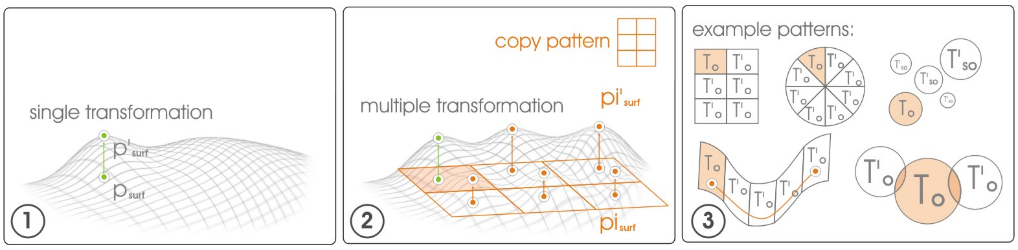

Figure 3.5: Illustration of the concept of cloning patterns. Image 1) shows a single deformation performed on the surface, image 2) the repeated deformation with the help of a simple rectangular cloning pattern. Image 3) shows a set of possible cloning patterns.

Simple patterns are basically segmented and parameterized 2D geometry, which is already available in SESAME. As the system also offers the drawing operations, so there is only a limited extra functionality that has to be included into the interface. Combined with strokes for more control and freedom in size and shape of the cloning pattern sketch input techniques can be used for parametrization. This approach could also be extended to the 3D domain parameterizing 3D objects and deforming them, resulting in more complex and irregular cloning patterns. A simple example for 3D domain cloning would be symmetry along the main axis, which is already incorporated in most of the current brush modeling software modules like MudBox, ZBrush and Blender.

D **Abstracted Interaction with Primitives**
Especially in more complex scenes and combination tasks the 3D interactions mode of SESAME incorporating the contact graph facilitate the interaction in 3D space. One current shortcoming of the technique can be seen in the fact, that modifications on lower levels of the contact graph (e.g., transforming a table with objects on it) does usually lead to artifacts in form of floating or intersecting objects. Using the structure of the contact graph, modifications in lower levels can also be propagated into higher levels an tested if they have any effect. To make this behavior more obvious, this idea can also be extended towards an direct, abstract visualization of the contact graph relations in 3D space. Single objects or groups of objects could be represented by simple primitives for selection or already impose certain transformations in form of 3D widgets (e.g., for scaling). As depicted in figure 3.6 this could facilitate recombination and restructuring with larger sets of objects and simple transformations for many objects could be more effective and visually tangible.

To effectively use the abstract structures, another focus and context view mode emphasizing the abstract shapes and lowering the visual contrast of the objects in the scene and lightning information, might be useful although it increases the complexity of the user interaction interface.

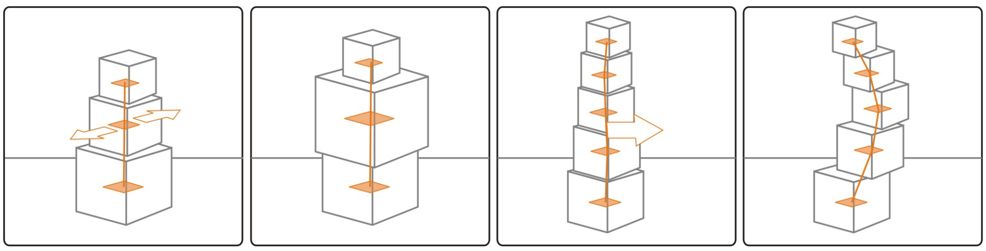

Figure 3.6: Concept of interacting with scene geometry represented by simple graphic primitives, in this case rectangular polygons. The hierarchical relation is indicated by a connecting line and the performed modification operations by arrows.

3.2.2 Modeling using Object Interaction

Intuitively the human mind often tries to relate digital modeling operations to processes, that can be observed in its environment as stated in section 2.3.1. Studies within the SESAME concept have shown that some of these processes could be reduced to some simple concepts, which can help to significantly improve interaction as for example the concept of gravity.

A **Movement Extrusion**
One strength of the SESAME system is the fast availability of 3D shape information out of 2D sketches. This approach could be extended towards already existing 3D shapes and applied movement operations. The surface together with its temporal translation over time defines the hull of a more complex shape, sharing some properties with the initial surface, similar to the concept of sweep objects in 2D. Therefore the direction of the movement of the shape at a certain point of time can be used to obtain the directional 2D contour of this object, which is swept along the path of motion like shown in image 3.7. By defining control points intermediate steps can be made accessible to modifications, for example to change the size of the object at a certain point of time. Afterwards, the propagation of the resulting changes to neighboring steps on the motion path allows flexibility towards the resulting shapes. Conceptually shapes produced with this technique can also be achieved with the original approach but with a significant higher number of intermediate steps. In order to use this technique effectively, the translation of objects in the scene has to be solved convincingly, otherwise this will cause problems with occlusions or unintended deformations.

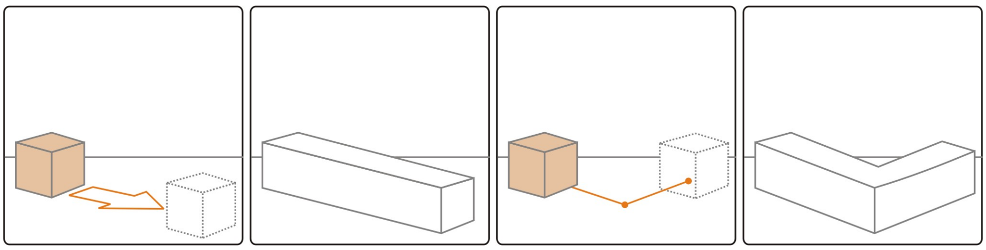

Figure 3.7: Object extrusion for a simple one dimensional translation and the movement along an extrusion path with defined control points.

B **Context Update**

Currently the contact graph in the SESAME system is only used to realize gravity like effects for the selection and translation of object groups in 3D space. This approach can be extended into more directions also maintaining structural information along other axes. Currently the objects are only treated in a static way, but similar approaches could also be used for the geometrical transformation of objects depicted in figure 3.8. The incorporation of information out of the construction process (construction history) could be used to produce the required information about the structure of certain objects.

Although offering opportunities for improving and increasing the interactions with multiple objects these internal structural relations have to be designed carefully. Otherwise especially for complex scenes they will result in ambiguous cases or over-determined descriptions of the relations. Additionally conditions like the intersection of other objects has to be checked for all transformed objects resulting in complex interaction scenarios.

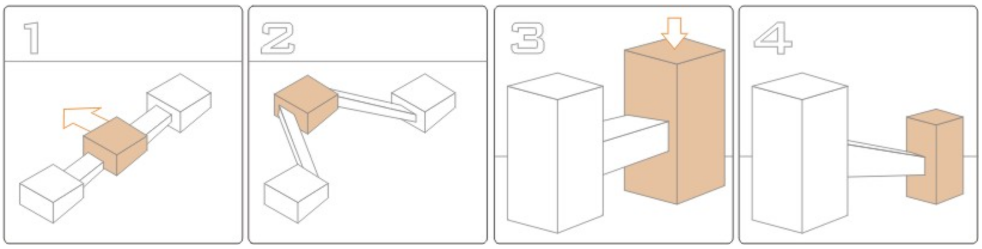

Figure 3.8: Automatic deformation by updating geometrical context information. Image 1 and 2 illustrate the effect of a simple translation of the activated object, while 3 and 4 demonstrate the possible impact of an scaling operation.

C **Object Surface Interaction**

This approach closely relates to observations that could be made in real world physics. Under the influence of interacting forces the surface of an object is deformed which can be used as an interaction paradigm for surfaces deformations. Similar approaches already have been done in earlier studies like [HHK92].

This description can be directly included within the interface of the current modeling system, because the only input parameters required to perform the operation are the spatial surface parameters of the interacting objects and a direction vector, that can be retrieved from the movement of the objects in the scene as depicted in figure 3.9.

On the other hand, the deformation itself might not be visible due to occlusions occurring in the scene, especially in complex scenes. The deformations are furthermore equivalent to a series of CSG operations and can also be achieved with the current system, but especially for more complex shapes with an significant higher amount of operations. To extend the number of possible shapes, parametrization of the movement path could be introduced resulting in curved path operation and rounded surfaces. The computational effort of this method will also raise with the number of interacting objects and an efficient collision detection method will be required.

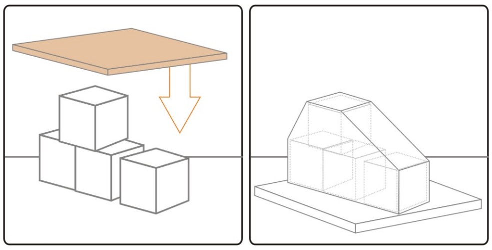

Figure 3.9: Interaction of an activated surface plane with a number of static scene objects.

D **Shape Interaction**

Similar to the previous described technique this approach relates to observations that can be made in real world physical interaction. Extending the deformation from the interacting surface towards a volumetric deformation, several new aspects come into play as it can be seen in figure 3.10. First there can be defined a set of different object behaviors, similar to physical parameters of different materials during interaction of real objects and secondly they can be assigned to different objects like done with textures for visual purposes.

A major drawback of this approach is, that exact local control of the deformation is not given and might be not suitable for the interaction with multiple complex objects with a detailed surface structure.

An apparent advantage can be seen in the fact, that the deformation only depends on the interacting shapes and the direction of the movement. So there are no additional interface elements or parameters required in the basic version and the introduction of the material metaphor offers flexibility towards the shape design. Once an object is created it can also be reused to create similar shapes out of other objects in form of imprints and stored in object libraries. This can be included as an additional interface element, allowing fast access to available shapes, similar to concepts implemented in ZBrush [2].

3.2.3 Sketch based Deformation Techniques

Sketching techniques are flexible, versatile and familiar to most users. They provide a decent input method for conceptual design purposes as already shown in section 2.3.3. A single stroke drawn on a 2D Interface already offers a number of parameters which can be mapped to interaction commands: start- and end point, general direction, position, tangency in every point, length and curvature. All of these features can be parameterized and represented with the help of spline curves and therefore be edited in later stages

[2]Pixologic, www.pixologic.com

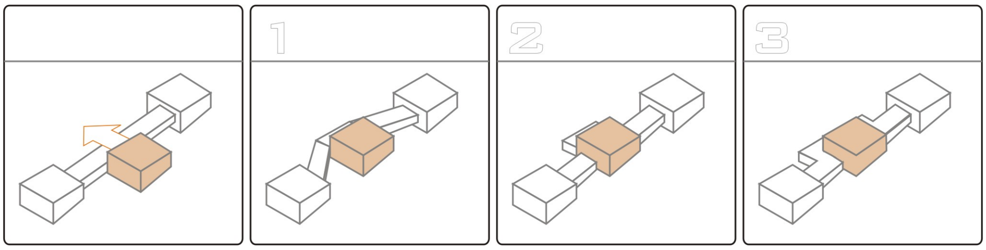

Figure 3.10: Interaction of an activated object with shapes in the scene. Image 1, 2 and 3 show possible deformation behaviors, which could depend on different material properties (e.g., high stiffness in figure 2 or changing volume in figure 3).

or used as input method for so called gesture commands as done in tabletop oriented systems like ShapeShop [SWSJ05].

A Line Parametrization

Depending on its position, length and curvature a single stroke can also be used to modify several parameters in one step. Representing a linear surface deformation by means of a vector $\overrightarrow{v_d}$ in 2D space, this vector can be deformed by an input stroke S_i and matched with the original extrusion. This could be done by projecting S_i and $\overrightarrow{v_d}$ into screenspace and finding the intersection points, which give the new parameter values for $\overrightarrow{v_d}$. A similar approach basing on a different object representations has been proposed in [GHQ04].

As shown in figure 3.11 a series of linear deformations represented by its extrusion vector can also be parameterized with one stroke. This step requires an appropriate parametrization of the surface deformations in means of an vector $\overrightarrow{v_d}$, which is easier to do for linear one directional extrusions like in SESAME, but will get more complex for already deformed or non linear deformations.

One advantage of this approach is the possibility to apply these parametrization transformations on any level of abstraction, which means the given multiple parametrization lines of previous steps can be parameterized again by another one. This gives theoretically the possibility to perform many deformation steps in a structured way with one input command.

On the other hand these deformations, especially on high levels of abstraction cannot be considered as being intuitive anymore so the plausibility of approaches like this, needs to be tested in practice and evaluated within user studies.

B Edge dragging

Another conceptual extension in SESAME would be the extrusion of single border lines of objects as shown for a simple case in figure 3.12. This concept can

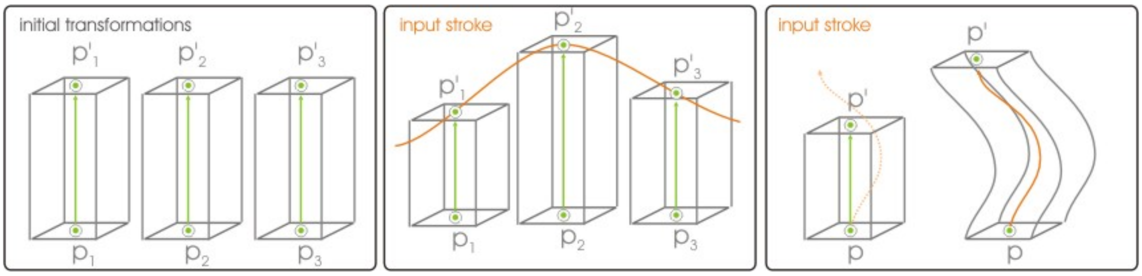

Figure 3.11: Parametrization of multiple linear deformations with one input stroke.

be incorporated into the existing paradigm, the main problem is to solve directional ambiguities of the extrusion, due to two adjacent faces with different normal directions.

On the other hand this will not directly extend the range of possible shapes available to SESAME, because same operations can also be performed with the existing system, but with a higher number of intermediate steps. Additionally the number of object borders will be significantly higher for more complex models and editing every single border will not be feasible anymore. This can lead to the extension towards grouping by certain edge features like the angle between adjacent faces.

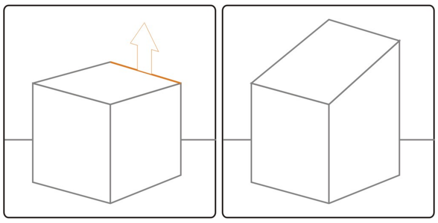

Figure 3.12: Simple extrusion of object boarders of a polygonal shape.

C **Line Surface Extrusion**
In its current implementation the extrusion paradigm can only be applied to closed polygonal surfaces and extrusions in the normal direction of this surface. As seen in figure 3.13 this approach could be extended towards freeform sketches and single lines on the surface of existing objects similar to the approach implemented in CB Model Pro [3].

Therefore several new aspects have to be considered: First there has to be defined a region of interest, which limits the area of deformation. This can be done by considering the surface limiting boundary edges. For curved surfaces this region

[3]Dassault Systems, www.cbmodelpro.com

of interest either has to be specified directly or automatically by certain boundary criteria like the direction of the surface normal or limiting the deformation by silhouette lines from the viewers perspective.

For closed section in the surface, the deformation can be considered constant within the closed section and especially the shape of the transition region will be important for the final result of the transformation.

Introducing this technique requires a restructuring of the current implementation of SESAME because it offers the possibility to create curved and non planar surfaces.

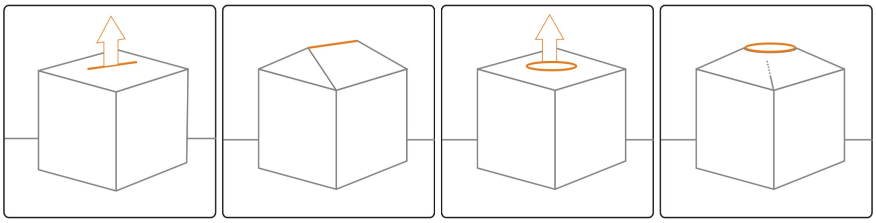

Figure 3.13: Deformation via line dragging for a simple linear stroke on the one hand and a defined closed region on the object surface on the other hand.

3.2.4 Concept Evaluation

Due to limited time resources a decision has be made, which concepts are selected for implementation. For this purpose a list of evaluation criteria has been defined to decide, whether the concepts are adequate for being implemented under the given circumstances. The result of this evaluation can bee seen in figure 3.14.

It is important to note, that the evaluation of the given concepts is supposed to aid the decision-making process during the project and does not depend on objective assumptions.

	criteria	I. Interaction techniques A: Automatic adjacency	B: Virtual planes	C: Cloning patterns	D: Primitive interaction
1	Interface complexity (G1)	+	+	-	-
2	Easy application (G2)	++	+	+	-
3	Incompleteness tolerance (G4)	-	-	+	-
4	Independent of representation (G6)	+	+	+	+
5	Estimated learning time	+	+	-	-
6	Improves process speed	+	-	++	++
7	Extends range of shapes	0	0	0	0
	Mean value:	0.71	0.29	0.43	-0.14

	criteria	II. Object interaction A: Movement extrusion	B: Context update	C: Surface interaction	D: Shape interaction
1	Interface complexity (G1)	+	-	+	-
2	Easy application (G2)	+	-	+	-
3	Incompleteness tolerance (G4)	+	-	-	-
4	Independent of representation (G6)	-	+	+	+
5	Estimated learning time	+	-	+	-
6	Improves process speed	+	++	+	+
7	Extends range of shapes	-	-	-	++
	Mean value:	0.43	-0.29	0.43	0.00

	criteria	III. Sketch based deformation A: Line parametrization	B: Edge dragging	C: Line extrusion
1	Interface complexity (G1)	+	+	+
2	Easy application (G2)	+	+	+
3	Incompleteness tolerance (G4)	0	-	++
4	Independent of representation (G6)	+	-	+
5	Estimated learning time	-	+	-
6	Improves process speed	++	0	0
7	Extends range of shapes	-	-	+
	Mean value:	0.43	0.00	0.71

Figure 3.14: Evaluation of the proposed concepts, whereas the first 4 criteria are derived from the guidelines(G) given in section 3.1.1.

4 Results and Implementation

4.1 Tools for Implementation

For implementation two different concepts have been chosen: One surface interaction approach and one sketch deformation technique.

In the beginning of the project it was also decided to implement the concepts independently from the original system. The main reason for this decision was the requirement for a different object representation structure, than the originally chosen one and to use the time spend on integration of the concept for their development. In addition to this the integration of the concepts was hindered by the sparse documentation of the original system.

So in order to realize the proposed concepts a decision has to be made which tools are used for implementation. For graphical applications the C++ and OpenGL was chosen because of various available libraries allowing high performance and flexibility towards application development. The implementation was done under the Operating System Microsoft Windows XP with Visual Studio 2005.

The Coin3D Library

For implementing the interaction framework the high-level 3D API Coin3D [1] was chosen. Coin3D is a independent development of the Open Inventor API with the main focus on fast access to import, rendering and interaction capabilities, while maintaining access to the basic OpenGL functionality.

The main structure of the Coin3D library is represented by scene graph data structures and is oriented towards interactive interface-based applications. The user interface handling for the applications is supported by the integration of the Windows standard GUI system. Additionally output functions for several 2D and 3D file formats are supported.

4.2 Object Interaction

The first technique selected for implementation is the object surface interaction technique.

Therefore a simple mesh structure was developed to handle the representation of objects described by a polygonal surface.

The aim of this construction is to provide a basic structure to render and access the objects on polygonal level. As shown in picture 4.1 the mesh structure consists of an set of Vertices $\mathbb{V}$, Edges $\mathbb{E}$ and Faces $\mathbb{F}$. The structures are linked to each other, so every

[1] Systems in Motion AS (SIM), http://www.coin3d.org

edge has two pointers on two vertices and two adjacent Faces. The faces have three pointers into the set of vertices and three for the adjacent edges as shown in figure 4.1. Additionally every face and edge has a flag, which is set to one if the edge or face in the current step has been visited, otherwise it is reset to zero after finishing one processing step. Furthermore all faces include an vector which contains the precomputed surface normal.

The global translation of an object is interpreted as the movement applied to all vertices of this object. For later interaction and deformation methods, this allows to parameterize the global movement to create local differences resulting in changes of the object shape.

The focus in this stage was mainly on functionality and accessibility, which results in some drawbacks in terms of performance for larger objects and more complex deformation operations.

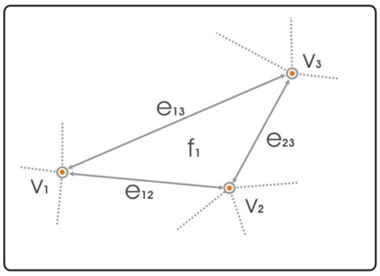

Figure 4.1: Simplified mesh representation concept

4.2.1 Mesh Smoothing

In order to evaluate results and to improve the structure of the mesh a simple local smoothing paradigm was included. This smoothing is based on the concept of a simple spring mass model for the edges between two vertices. For every edge in the mesh structure a value holding its original length is stored. If the edge $e_i \epsilon \mathbb{E}$ becomes longer than its initial value, the vertices are moved into the direction of each other. The stronger an edge is deformed the further its vertices will be moved.

This step is repeated iteratively over the whole mesh, resulting in an easy but for this purpose effective smoothing operation, which can be controlled over the number of iterations that is performed. If the initial mesh is well constructed in terms of the shape of the triangles, this local method also yields acceptable results and the progress can bee seen immediately, offering direct visual feedback about the smoothing results.

After every successful modeling step, the current length of the edited edges is reassigned to its initial value to store implicitly spatial relations in the shape.

4.2.2 Mesh Reconstruction

For larger deformations and displacements of the vertices the mesh usually contains irregularly shaped triangles which are sought to be avoided, as they produce visual artifacts and influence the result of further modeling steps. For this reason two splitting mesh methods are incorporated in the mesh structure: Edge splitting and Face split.

Both steps require the insertion of new faces, three new edges and one new vertex and result in additional sample points for representing the geometry. The original algorithm for this procedure is describe in [KCVS98]. The concepts of both techniques can be seen in figure 4.2.

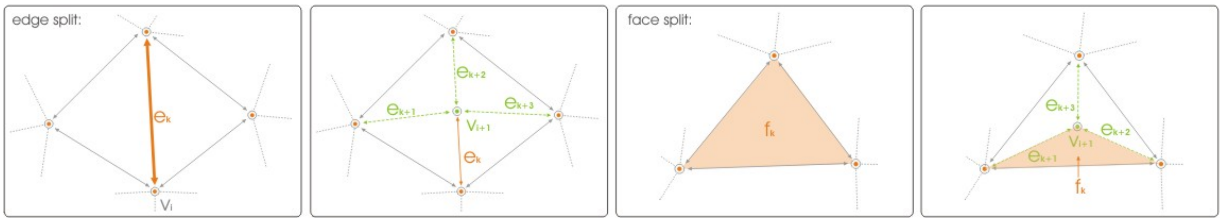

Figure 4.2: Concept of both splitting methods

The splitting can be done for a certain global edge length threshold or surface area. if the value of one of these parameters is exceeded the split operation is executed for the relevant faces and edges and the new normal vectors of the inserted faces are computed by comparing to the neighboring faces.

4.2.3 Interaction Concept

To realize the interaction of the surfaces there are two factors important for the deformation: The Current position p_i of the vertex v_i in the mesh and its translated position $\hat{p}_i$ resulting from the movement of the whole object. The deformation is done by testing, the ray r_i between p_i and $\hat{p}_i$ for intersections with the scene geometry. If there is any interesection, the meshpoint is moved to the position of the intersection and than stopped.

For numerical reasons, the point is not moved directly on the surface intersection point, but stopped before a threshold distance to avoid the vertex moving through the geometry of another object in further movement steps. In this concept the object, which is currently moved is activated and deformed, avoiding unwanted deformations in other parts of the scene.

The result of this interaction can be seen in picture 4.3.

From the basic description there are already some important properties that can be derived of this concept: First of all the result of the operation and shape representation of the interacting objects depends on resolution of the initial mesh because the collision detection is performed on the mesh vertices.

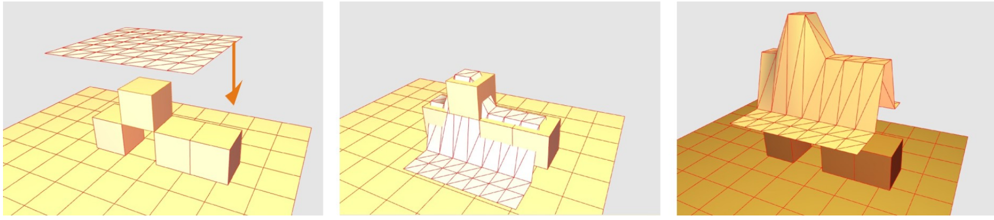

Figure 4.3: Results of the surface interaction concept

On the other hand the deformation method can be performed from any direction and from different sides of the interacting objects.

As an extension to this basic description it would be possible to define the collision points on the mesh more faithfully by projecting the borders of the static object on the moved mesh. Another possibility would be to take a closer look at the edges between points which are stopped and and points, which can perform the full object translation, without being blocked by an obstacle. By dividing the edge into a fixed ratio and testing the split position for the collision property, the are of the object border imprint can be determined with higher accuracy.

Another extension might include libraries of already deformed objects which can be reused for interaction or a special included library which offers a set of tools which are known to produce a wide variety of shapes.

One drawback of the current implementation is the speed the deformation can be performed with. For a low number of surface intersection tests the method is quite fast due to basic ray intersection test optimization provided in Coin3D. For a larger number of vertices (1000) the framerate drops below interactive threshold. Therefore a more efficient implementation and the use of GPU shaders will offer acceleration possibilities.

4.3 Sketch Extrusion

As already described in section 2.6.3 sketches provide a powerful technique to represent conceptual ideas and have been successfully used in conceptual 3D applications. Besides the fact that the concept of sketches is well known to a larger group of users, they also offer a variety of parameters that can be used for interacting in a virtual environment.

As known from their real counterpart strokes can be drawn with usual input devices like mouse or a digital pen directly into the scene like already done in the original SESAME System, where linear strokes can be drawn directly on the object surface and closed areas can be extruded. This approach is extended towards the extrusion of single freeform strokes, which might not define a closed area as shown in picture 3.13.

In general this concept can be described by the following steps: First the user defines a set of sketches on a given geometry which are processed automatically to provide necessary information for further steps. After processing the user can perform an extrusion on an activated stroke in direction of the surface normal and finish editing if the intended result is achieved. The underlying algorithmic concept is shown in figure 4.4.

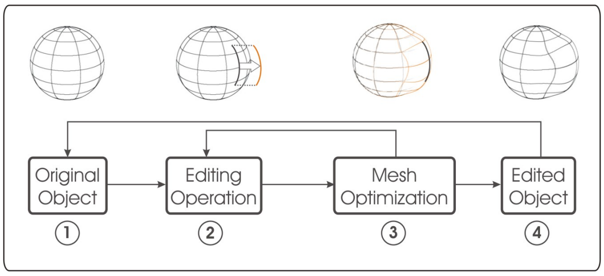

Figure 4.4: The Process of deforming an model with the help strokes

4.3.1 Stroke Processing

Mathematically strokes can be represented as spline curves over a set of given control points $\mathbb{P} \in \mathbb{R}^3$. Given this set of points all positions on the stroke can be reconstructed by interpolation. In the current implementation this is done by Catmull-Rom splines as described in [Far88].

When defining the input points on a surface it has to be ensured, that they are positioned on the surface of the same object and the distance between them is not too large. In the ideal case neighboring control points p_i and $p_{i+1} \in \mathbb{P}$ on the stroke are either positioned on the same surface polygon or positioned on adjacent polygons, otherwise the results of the following deformation operations in such areas might be unpredictable or result in visual artifacts. Additionally it is useful to smoothen the given input samples to certain degree to improve the visual result.

The implemented algorithm for the definition of strokes performs the following steps:

First the user starts clicking in the scene selecting the first object hit by the ray between the current camera position and the corresponding point on the viewing plane. By dragging the mouse over the object the user defines a set of input points, which are sampled over the surface of the selected object as shown in figure 4.5 b). Additionally a data structure is created holding all the points belonging to the stroke, the corresponding surfaces and their mean normal direction. Currently the sampling rate mainly depends on the speed of the executing system, which might have to be changed according to the distance or a fixed sampling time to provide more constant sampling during the stroke drawing process.

For every point the corresponding polygon on the surface is stored and it is tested whether the polygons of two neighboring points p_i and p_{i+1} for i $\in$ [0,n-1] on the line are either the same or at least adjacent to each other.

If this is not the case an automatic gap closing function is called to ensure this property. This done by taking the corresponding surface of p_i and checking if p_{i+1} either is located on this surface or on one of the neighboring faces. If this test fails, the algorithm automatically selects out of the set of adjacent polygons the one, which has its center of gravity closest to the position of p_{i+1} according to the euclidian distance measure.

This last step is repeated until the corresponding polygon is found or a threshold value for the number of examined polygons is exceeded. In the last case the line is split up into two segments and the processing is continued on the second segment, although for an appropriate sampling rate this case should be rare.

As soon as the user finished drawing, a smoothing operation is performed using the Catmull-Rom description. Therefore all points of the stroke are iterated and their position on the stroke is computed by taking the 4 neighboring points on the spline into account (two in each direction) and computing the position on the constructed spline at t=0.5 moving the current point to this position while first and the last two points of the stroke are fixed. This operation can be performed several times resulting in a stepwise smoothing of the stroke. By observation it has turned out that 5 to 10 repetitions produce an appropriate smoothing of the stroke without loosing too much detail information or moving the stroke too far away from its original position.

After this smoothing operation a closed edge path on the polygonal surface is computed, which is closest to the input stroke. This is done as the introduced stroke might represent corners or sharp features after the deformation, which cannot be faithfully displayed with edges intersecting this feature as they produce visual artifacts. This procedure has been introduced in [NSACO05] and modified for this concept. For the implemented concept a modified version of the A* algorithm has been used (first described in [HNR68]). In a first step the edges of all corresponding polygons of the stroke are selected and their euclidian distance of the bounding edges to the corresponding stroke is computed representing their weight. This distance computation is approximated by computing and accumulate the distances of both vertices defining the edge. On this set of weighted edges the A* Algorithm is carried out, finding a set of closed edges including the desired closest edge path and a number of marked edges, which are close to the stroke, but all their following adjacent edges are to far away.

These unused edges are removed in a second step, starting from the last edge found from the algorithm and selecting the longest connected edge path. One drawback of this implementation is that self-intersecting strokes cannot be handled correctly, as the algorithm always finds the shortest path from p_0 to p_n on the marked polygons, ignoring loops and intersecting parts of the stroke. This case is shown in figure 4.6 b).

After these steps the vertices on the closest edge path are moved directly to their corresponding perpendicular position on the input stroke as they might represent sharp features or creases in later modeling steps. The program visualizes the success of these operations by changing the color of the input stroke and selecting the stroke as currently activated for further processing as shown in 4.5 c).

Due to an error in the current implementation, the movement of the nodes on the activated stroke partially also effects nodes on previously defined strokes. This results in an unexpected local translations of the vertices, which is shown in figure 4.6 a).

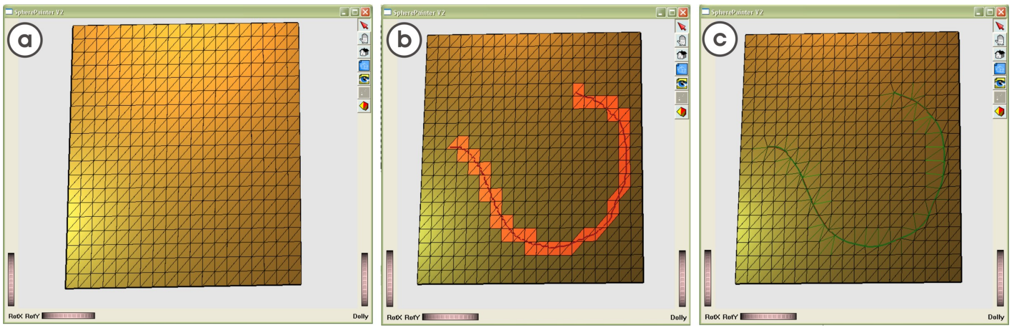

Figure 4.5: The processing of an input stroke shown on a planar mesh with a) showing the initial mesh, b) the visual feedback while drawing on the mesh with coloring the corresponding polygons and representing the currently drawn points as red line. In c) the smoothed stroke and the adjusted mesh is shown.

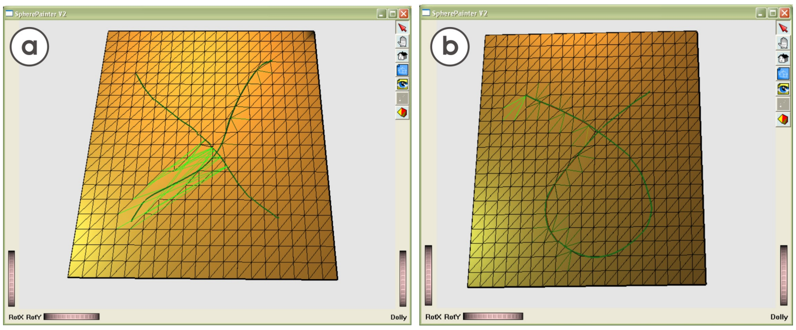

Figure 4.6: Results of two known implementation issues. Figure a) shows unexpected local vertex translations for overlapping strokes and figure b) illustrates the inadequate computation of the closest edge path for self-intersecting strokes.

4.3.2 Extrusion Concept

After these automatic stroke processing steps the user can extrude the currently activated sketch in direction of the mean surface normal $\vec{v_e}$ of the intersecting object polygons.

This can easily be done by dragging the mouse cursor while the second mouse button is pressed, as in the original SESAME System. The screen space distance in y direction to the original dragging location thereby defines the amount of extrusion.

The amount of extrusion is described with help of a height- or blending function $\mathbf{h}(\mathbf{p})$ which is in principle shown in figure 4.7 and corresponds to the transformation propagation concept in section 2.5.1.

In general a deformation $\mathbf{d} : \mathcal{S} \rightarrow \mathbb{R}^3$ function describes the change of a mesh structure

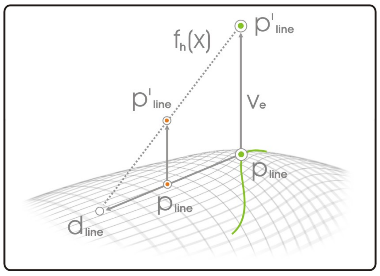

Figure 4.7: Concept of the extrusion with a height function

$\mathcal{M}$ into its modified version $\mathcal{M}'$:

$$\mathcal{M}' := \{\mathbf{p_i} + \mathbf{d}(\mathbf{p_i}) \mid \mathbf{p_i} \in \mathcal{M}\} \tag{4.1}$$

This description is extended by the definition of the height function:

$$\mathcal{M}' := \{\mathbf{p_i} + \mathbf{d}(\mathbf{p_i}) * \mathbf{h}(\mathbf{p_i}) \mid \mathbf{p_i} \in \mathcal{M},\ \mathbf{h} \in [\mathbf{0}, \mathbf{1}]\} \tag{4.2}$$

While $\mathbf{d}(\mathbf{p_i})$ equals the current extrusion of the activated stroke the height function $\mathbf{h}$ equals 1 for all vertices on that stroke and 0 for all vertices which are outside a certain region of interest.

This region can be defined automatically or directly by the user, which offers higher control over the deformation result. One possibility to limit the polygon extrusion automatically is to define a distance threshold to the input stroke and limit the deformation to faces with a normal pointing in a similar direction as the extrusion vector. Another way is to let the user interactively define additional strokes to limit the amount of the extrusion to a certain region.

The concept of the extrusion function is quite flexible and can easily be adapted towards different behaviors which can be seen in figure 4.8. To define a continuous scalar field around a stroke, the distance field can be computed denoted by $\mathbf{dist}(\mathcal{S}, \mathbf{p}),\ p \in \mathbb{R}^3$ which defines a scalar value for every point according to an given stroke.

With this definition $\mathbf{h}(\mathbf{p_i})$ can be defined as following:

$$\mathbf{h}(\mathbf{p_i}) := \begin{cases} 1 - \frac{dist(\mathcal{S}, p_i)}{c_s * |d(p_i)|} & , \quad dist(\mathcal{S}, p_i) \ < \ c_s * |d(p_i)| \\ 0 & , \quad else \end{cases} \tag{4.3}$$

This results in a linear extrusion around the activated stroke with the slope c_s. The further the stroke is extruded, the larger gets the influence of the extrusion function in this case. If the boundary condition is set to a constant value the slope of the height function in the blending region changes with the amount of the extrusion.

The behavior of the extrusion function can be modified by various other functions, which might also be defined by the user. This leads to the following general description:

$$\mathbf{h}(\mathbf{p_i}) := \begin{cases} f_h(dist(\mathcal{S}, p_i), |d(p_i)|) & , \quad b_h(dist(\mathcal{S}, p_i), |d(p_i)|) \\ 0 & , \quad else \end{cases} \tag{4.4}$$

Which denotes the results and the appearance of the deformation by $\mathbf{f_h}(\mathbf{x})$ and the boundary condition for this deformation by $\mathbf{b_h}(\mathbf{x})$ which are independent from each other and $\mathbf{b_h}(\mathbf{x})$ returns a boolean value. For most interactive deformations both functions might be depending on the position and distance to the currently activated stroke and the amount of the extrusion.

This representation can be defined independently form the underlying surface description and also for points which are not directly located on the surface.

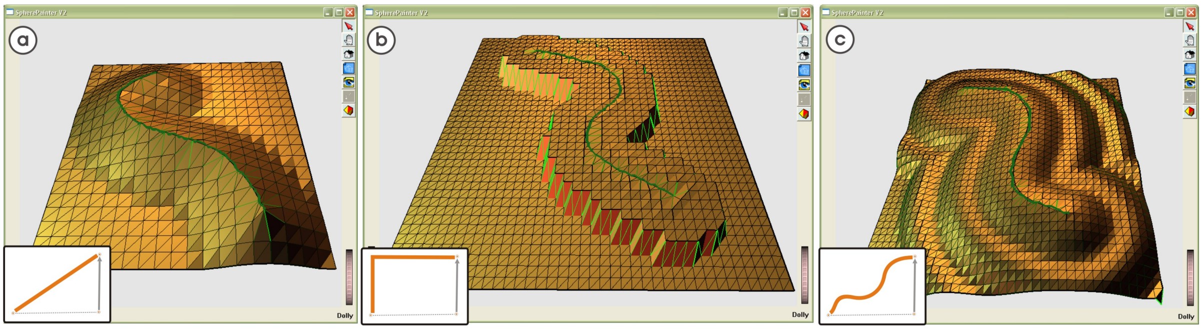

Figure 4.8: The effect of different extrusion behaviors on a planar surface. While a) shows the extrusion with a linear slope of the height function as defined in equation 4.3, b) represents the extrusion with a distance threshold and c) the combination with a simple sinus function depending on the distance to the stroke

Therefore definition of the distance field and the depending height function also allows various extensions.

A stroke on a polygonal object surface together with an corresponding distance field and its local tangent direction defines a 2D subset of the surface. On this area deformations of the original surface in normal direction can be interpreted as 2D height field. This height field can be extracted and separated from the original geometry which is shown in figure 4.9 b) by interpreting the height values as grey levels with 0 as maximal extrusion.

If the placement of the 2D height is aligned along the input stroke, complex convex geometric extrusions can be created on the object surface just by defining multiple strokes on the surface as seen in 4.9 c).

This concept can also be extended towards the capturing of existing surface features and together with the surface normal also 3D subsets can be defined which would allow more complex non-convex geometry extrusion of shape features requiring structural changes of the underlying polygonal mesh by means of a additional mesh refinement.

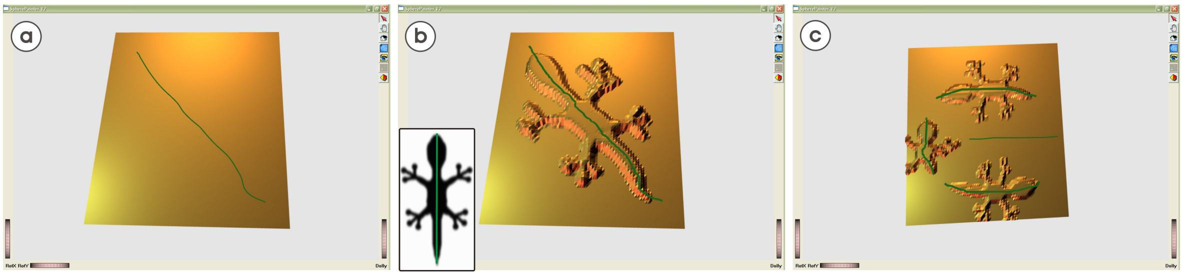

Figure 4.9: Application of a scalar height function along the original stroke shown in a) the result of the extrusion in b) and multiple strokes applied in c)

4.3.3 Concept Extension

Up to now the definition of extrusions and the area of influence solely depends on automatic definitions limiting the control over the final extrusion result. Therefore the concept is extended towards the definition of multiple interacting strokes to limit and control the shape deformation.

In order to get an reasonable result the position of every vertex that has to be translated is evaluated in terms of its position relative to the strokes. Strokes surrounding the currently activated one are interpreted as being static and vertices which are on or beyond them are not supposed to move.

The first concept is to move all vertices between two strokes in a linear fashion so the slope of the deformation area is controlled over the height of the extrusion. Additionally the extrusion should also produce reasonable results in areas, where no strokes are defined.

To construct these conditions, the boundary condition $\mathbf{b_h}(\mathbf{x})$ has to be adapted towards finding the position on the next relevant stroke. This is done by constructing a plane through the current position of the mesh vertex, its corresponding perpendicular position on the activated stroke in its initial and on its extruded version. This plane is intersected with the relevant strokes around this position and for the case of multiple intersections, the closest position is selected as boundary point.

In case, that there are no intersections, the closest perpendicular position on any other stroke is taken into account. The results of this approach can be seen in figure 4.10. What also becomes visible is, that it is possible, that with the extrusion new sharp features and edges are produced on the surface and in further extension it might be useful to automatically recognize critical areas with high curvature and automatically introduce lines to handle these features.

The advantage of this approach is its flexibility towards missing input. Even for very few boundary conditions given by the input strokes the result is still predictable. In

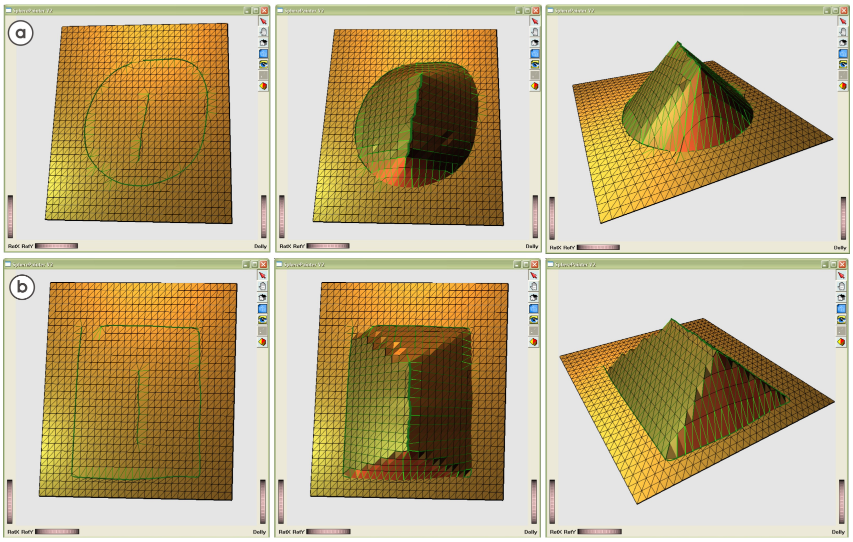

Figure 4.10: Two examples for simple extrusions of a stroke limited by another bounding stroke.

the current implementation the method is purely local as properties of the surrounding surface are not considered during the deformation. Therefore local discontinuities are likely to occur in areas with fewer boundary conditions. Some examples for cases with very few input lines are shown in figure 4.11.

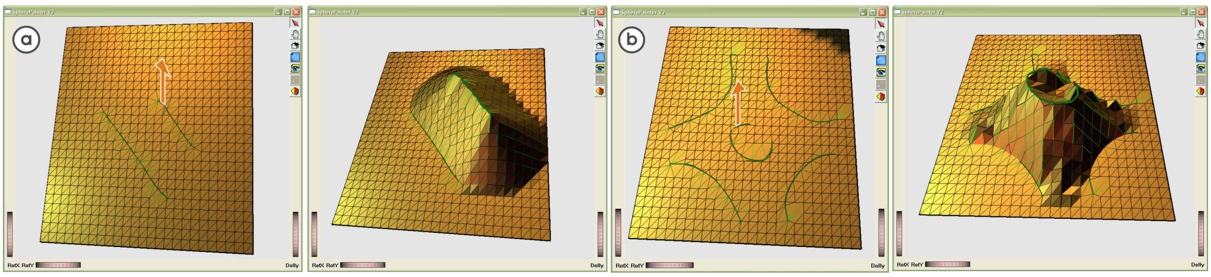

Figure 4.11: Extrusions with only a few boundary conditions (the extruded line and the extrusion direction is indicated by an arrow)

In order to get more control over the final shape and to interact with the extrusion result the height function itself can be interpreted as a spline. The control points of this spline are given with the boundary conditions and by the definition of two vectors in each control point as shown in figure 4.12 the surface can be reconstructed. The results of this approach can be seen in figure 4.13.

This concept also allows to decide whether the extrusion should be interpreted as sharp

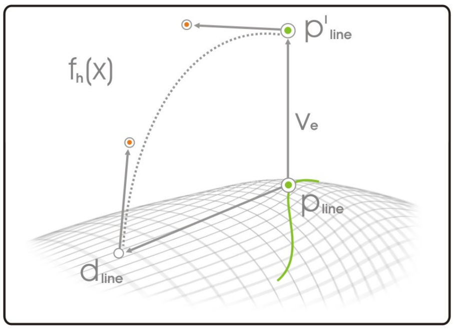

Figure 4.12: Interpretation of the height function as a spline.

feature on the surface or if it has a smooth appearance by aligning the control vectors along the tangent vectors of the surrounding surface. On the other hand the shape of the extrusion can get rather complex for many input strokes and therefore for the further optimization of the mesh structure and remeshing during the extrusion operations gets necessary in order to get adequate results. Also the interaction with these elements is getting more complex. The vectors to determine the final control points could be determined by evaluating the directions of intersecting strokes.

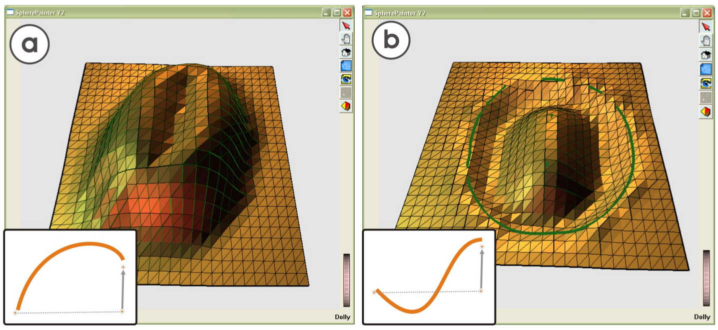

Figure 4.13: Two examples for surfaces created with the controlled height function for a similar stroke setting like in figure 4.10 a). Both result are smoothed with 5 iterations.

The main drawback of the current implementation is the low framerate resulting from the simple mesh structure. Especially for testing the efficiency of this approach and its applicability in practical systems interactivity one main aspect for further improvements.

5 Conclusion and Future Work

5.1 Summary

The task of this thesis was to develop interaction techniques for the existing SESAME modeling system.

These techniques should support especially the conceptual design phase and therefore intuitive interaction in means of a minimal cognitive effort performing these operations, as well as the exploration of different design solutions. Furthermore they are supposed to be used with a minimum of required previous knowledge and match general shape expectations although the give input might be incomplete.

Additionally the concepts should extend the variety of shapes which can be created or improve the interaction with them. The proposed deformation and interaction methods are supposed to be independent from the actual object representation and support the modification of already existing objects as well as the creation of new shapes.

In order to develop appropriate concepts an overview about the application areas together with an introduction to the topic of shape modeling has been given to retrace the basic ideas which have lead to the proposed concepts. This overview is followed by a brief discussion and classification over currently existing object representation and deformation techniques in correspondence to the issues during the conceptual design phase. In addition to this existing modeling systems have been compared to identify shortcomings and advantages under the aspects of the given task and their applicability towards the later concepts.

Out of these considerations a number of conceptual ideas have been proposed and after an evaluation given in figure 3.14, two of them have been selected for implementation and detailed discussion. Within this discussion strategies for the realization have been developed, as well as pointing out the problems connected to these attempts and further improvement proposals.

5.2 Final Conclusions and Evaluation

The first selected technique is based on an object surface interaction concept. Out of the given movement direction of an object in the scene and the collision with other objects, the surface is deformed and new shapes can be created out of the existing objects in the scene.

The advantages of this concept towards the initial task are the predictable and intuitive interaction and no requirement for additional interface elements. Therefore the conceptual integration into the existing SESAME framework can be assured and the principles behind this deformation technique could be learned within a few examples. Furthermore more complex shapes can be created out of simpler existing shapes in the scene, supporting a fast context related exploration of possible shapes.

The drawback of the concept is that it does not extend the variety of shapes towards the existing extrusion paradigm and most of the shapes could also be created with CSG operators although a higher number of modeling steps and additional interface elements would be required. Another criteria which is not fulfilled by the current implementation is the independency from the object representation. The final result of the shape interaction basing on an polygonal description strongly depends on the density of the given mesh, which should be avoided by an alternative representation of the deformation. In addition this concept demands an exact control of the object movement and deformation results might be hidden due to occlusion in the scene.

On the other hand the extension of this concept towards the idea of different interaction behavior depending on a material metaphor could result in a valuable extension of the shape variety.

The second implemented concept offers a variety of interaction possibilities and new shapes. It complements the original operations without complicating the existing user interface in a significant amount. To realize an additional interaction mode freeform strokes are required and the interaction with this strokes can be realized directly using their representation within the scene.

The effort in learning the deformation concept basing on freeform-strokes can be considered higher than in the first proposed method. Based on the assumption, that most users will be familiar with the concept of strokes on object surfaces, it should not extend the learning phase in a significant way. By offering immediate feedback of the deformation on the surface the user can also evaluated the result of the operations and make further design decisions basing on these modifications supporting an iterative creation process.

Shape deformation in this concept is also supported for few input strokes and produces predictive extrusions. Together with the concept of interacting with the extrusion result basing on a spline concept described in section 4.3.3 the exploration of design results is also supported.

Furthermore the conceptual definition of the deformation with the help of the height function, as described in section 4.3.2 is independent from the final object representation. Although the visual result of these modifications for polygonal representations depends on the adequate sampling of the surface. This has to be ensured by additional mesh optimization procedures.

One downside of this approach can be seen the missing possibility to create 3D content from the scratch. Up to now this deformation concept is defined on existing objects which can be included in the scene in form of simple shape primitives. Additionally the current simple mesh representation structure limits the interactivity of this approach to objects with a limited number of vertices and needs further optimization in order to deal with larger geometric data sets.

For both concepts their early development state did not allow effective user studies in order to prove the given assumptions about their applicability in practical use and might represent an important aspect in the further development.

5.3 Future Work

Within the previous chapter the advantages and disadvantages of the proposed approaches already have been outlined and hint on different improvement aspects.

Besides the already referenced improvement opportunities these aspects include the enhancement towards higher interactive display rates of the deformation operations on complex objects and further support of methods to enhance the final surface quality and creation of more complex structures.

Another important aspect is the extension of these concepts towards volumetric object representations. One problem of polygonal models is their property to imply a certain visual shape completeness to the user which actually might not be present in early conceptual stages as mentioned in chapter 2.3.3.

Therefore further concept extension towards alternative visual representation of incomplete shape models and the incorporation of different NPR rendering techniques can be considered as a valuable extension of these concepts.

Furthermore the applicability and usability properties of the concepts have to be proven in application in a practical environment and evaluating user studies.

Bibliography

[AFM$^+$00] David Anderson, James L. Frankel, Joe Marks, Aseem Agarwala, Paul Beardsley, Jessica Hodgins, Darren Leigh, Kathy Ryall, Eddie Sullivan, and Jonathan S. Yedidia. Tangible interaction + graphical interpretation: a new approach to 3d modeling. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 393–402, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[AMPC$^+$04] Alexis Angelidis, Marie-Paule, Wyvill Cani, Geoff Wyvill, and Scott King. Swirling-sweepers: Constant volume modeling. In *Pacific Graphics*, Korea, october 2004. Best paper award.

[BC02] Peter J. Bentley and David W. Corne. *Creative Evolutionary Systems.* Morgan Kaufmann, 2002.

[BM03] Ioana M. Boier-Martin. Domain decomposition for multiresolution analysis. In *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 31–40, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[BP07] Mario Botsch and Mark Pauly. Course 23: Geometric modeling based on polygonal meshes. ACM SIGGRAPH 2007 courses, 2007.

[CA06] Marie-Paule Cani and Alexis Angelidis. Towards virtual clay. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, pages 67–83, New York, NY, USA, 2006. ACM.

[CCH$^+$02] Bruce Campbell, Paul Collins, Hunter Hadaway, Nick Hedley, and Mark Stoermer. Web3d in ocean science learning environments: virtual big beef creek. In *Web3D '02: Proceeding of the seventh international conference on 3D Web technology*, pages 85–91, New York, NY, USA, 2002. ACM.

[Far88] Gerald Farin. *Curves and surfaces for computer aided geometric design: a practical guide.* Academic Press Professional, Inc., San Diego, CA, USA, 1988.

[FCAD03] N. Faria, T. Cardoso, B. Araújo, and F. Dias. Gides++ - 1st usability test. TECHNICAL REPORT 1, CENTIFME, 2003.

[GHQ04] Xiaohu Guo, Jing Hua, and Hong Qin. Point set surface editing techniques based on level-sets. In *Computer Graphics International Proceedings*, pages 52–59, June 2004.

[Hav05] Sven Havemann. *Generative Mesh Modeling.* PhD thesis, TU Braunschweig, 2005.

[HHK92] William M. Hsu, John F. Hughes, and Henry Kaufman. Direct manipulation of free-form deformations. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 177–184, New York, NY, USA, 1992. ACM.

[HNR68] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, SSC-4(2):100–107, 1968.

[HPSP01] Horst K. Hahn, Bernhard Preim, Dirk Selle, and Heinz Otto Peitgen. Visualization and interaction techniques for the exploration of vascular structures. In *VIS '01: Proceedings of the conference on Visualization '01*, pages 395–402, Washington, DC, USA, 2001. IEEE Computer Society.

[HQ02] Jing Hua and Hong Qin. Haptics-based volumetric modeling using dynamic spline-based implicit functions. In *VVS '02: Proceedings of the 2002 IEEE symposium on Volume visualization and graphics*, pages 55–64, Piscataway, NJ, USA, 2002. IEEE Press.

[HS97] Donald D. Hoffman and Manish Singh. Salience of visual parts. *Cognition*, 63(1):29–78, 1997.

[IH01] Takeo Igarashi and John F. Hughes. A suggestive interface for 3d drawing. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 173–181, New York, NY, USA, 2001. ACM.

[IMT99] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy: a sketching interface for 3d freeform design. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 409–416, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[KB04] Leif Kobbelt and Mario Botsch. A survey of pointbased techniques in computer graphics. *Computers and Graphics*, 28(6):801–814, 2004.

[KCVS98] Leif Kobbelt, Swen Campagna, Jens Vorsatz, and Hans-Peter Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 105–114, New York, NY, USA, 1998. ACM.

[LPRS05] Ignacio Llamas, Alexander Powell, Jarek Rossignac, and Chris D. Shaw. Bender: a virtual ribbon for deforming 3d shapes in biomedical and styling applications. In *SPM '05: Proceedings of the 2005 ACM symposium on Solid and physical modeling*, pages 89–99, New York, NY, USA, 2005. ACM.

[LS04] Chia-Hsun Lee and Ted Selker. isphere: a proximity-based 3d input device. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Posters*, page 72, New York, NY, USA, 2004. ACM.

[LVKY02] Henning Biermann Luiz Velho Ken, Ken Perlin and Lexing Ying. Algorithmic shape modeling with subdivision surfaces. *Computers and Graphics 2006*, 26(6):865–875, December 2002.

[LXJY05] Guo Liping, Liu Xiaoqing, Lei Jiaheng, and Cheng Yongxi. Description for structure of substances by computer 3d animation in chemical education. 2005.

[MTG+04] Mikhail Morozov, Andrey Tanakov, Alexey Gerasimov, Dmitry Bystrov, and Eduard Cvirco. Virtual chemistry laboratory for school education. In *IEEE International Conference on Advanced Learning Technologies*, number ISBN 0-7695-2181-9, pages 605–608. Mari State Technical University, 2004.

[Nay95] Bruce F. Naylor. Interactive playing with large synthetic environments. In *SI3D '95: Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 107–ff., New York, NY, USA, 1995. ACM.

[NISA07] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. Fibermesh: designing freeform surfaces with 3d curves. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 41, New York, NY, USA, 2007. ACM.

[NSACO05] Andrew Nealen, Olga Sorkine, Marc Alexa, and Daniel Cohen-Or. A sketch-based interface for detail-preserving mesh editing. *ACM Trans. Graph.*, 24(3):1142–1147, 2005.

[OLG+05] John D. Owens, David Luebke, Naga Govindaraju, Mark Harris, Jens Krüger, Aaron E. Lefohn, and Timothy J. Purcell. A survey of general-purpose computation on graphics hardware. In *Eurographics 2005, State of the Art Reports*, pages 21–51, august 2005.

[ONI06] Shigeru Owada, Frank Nielsen1, and Takeo Igarashi. Copy-paste synthesis of 3d geometry with repetitive patterns. *Smart Graphics 2006*, 2006.

[RMT05] II Russell M. Taylor. Haptics for scientific visualization. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Courses*, page 174, New York, NY, USA, 2005. ACM.

[RS97] Ari Rappoport and Steven Spitz. Interactive boolean operations for conceptual design of 3-d solids. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 269–278, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[SBS06] Jia Sheng, Ravin Balakrishnan, and Karan Singh. An interface for virtual 3d sculpting via physical proxy. In *GRAPHITE '06: Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pages 213–220, New York, NY, USA, 2006. ACM.

[Sin06] Karan Singh. Industrial motivation for interactive shape modeling: a case study in conceptual automotive design. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, pages 3–9, New York, NY, USA, 2006. ACM.

[SOD05] W. Stuerzlinger, Ji-Young Oh, and John Danahy. Comparing sesame and sketching on paper for conceptual 3d design. In *Sketch Based Interfaces and Modeling*, number ISSN 1812-3503, pages 81–88. York University, 2005.

[SOD06] W. Stuerzlinger, Ji-Young Oh, and John Danahy. Sesame: Towards better 3d conceptual design systems. 2006.

[SWSJ05] Ryan Schmidt, B. Wyvill, M. C. Sousa, and J. A. Jorge. Shapeshop: Sketch-based solid modeling with blobtrees. In Joaquim Armando Pires Jorge and Takeo Igarashi, editors, *Eurographics Workshop on Sketch-Based Interfaces and Modeling*, pages 53–62, Dublin, Ireland, 2005. Eurographics Association.

[vFTS06] Wolfram von Funck, Holger Theisel, and Hans-Peter Seidel. Vector field based shape deformations. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 1118–1125, New York, NY, USA, 2006. ACM.

[VH98] I.M. Verstijnen and J.M. Hennessey. Sketching and creative discovery. *Design Studies*, (19):519–546, 1998.

[Wat07] Benjamin Watson. Real and virtual urban design. In *SIGGRAPH '07: ACM SIGGRAPH 2007 courses*, pages 167–228, New York, NY, USA, 2007. ACM.

[WS01] Gerold Wesche and Hans-Peter Seidel. Freedrawer: a free-form sketching system on the responsive workbench. In *VRST '01: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 167–174, New York, NY, USA, 2001. ACM.

[ZHH96] Robert C. Zeleznik, Kenneth P. Herndon, and John F. Hughes. Sketch: An interface for sketching 3d scenes. In *Proceedings of SIGGRAPH 96*, Computer Graphics Proceedings, Annual Conference Series, pages 163–170, august 1996.

[ZTCS99] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape from shading: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):690–706, 1999.