Modelling on Interactive Surfaces for 3D Fabrication

Miguel Ferreira Nixo

***Abstract*—The availability and use of digital 3D objects increased over the recent years, together with low-cost 3D printing. Such scenario raised the necessity of tools to create and modify these objects efficiently and with natural approaches, so that they can be used by both trained and non-trained users. With this work, we propose to develop a walk-up-and-use 3D modelling approach aimed at 3D fabrication, that takes advantage of methods for 3D object retrieval to provide interactive feedback, and the use of interactive surfaces, as the latter are becoming increasingly more popular in devices such as smartphones, tablets and even tabletops. In order to provide a natural interaction experience, we developed and discussed approaches that are solely based on touch or pen interaction, and others that combine both. After conducting a user evaluation process, results suggest that pen-based approaches are satisfying although not as efficient and entertaining to the average user as the touch-based and combined pen and touch approaches, in modelling scenario for 3D fabrication.**

***Keywords*—Multi-touch Interaction, Pen Interaction, 3D Modelling, 3D Object Retrieval, 3D Fabrication.**

I. Introduction

The 3D printing technology and community has undergone significant growth during the recent years, empowering the common user to fabricate 3D printed models at home at relatively low costs. Also, with the generalization of interactive surfaces in every-day devices such as smartphones and tablets, Multi-touch enabled interactive tabletops that existed for over forty years are becoming more desired and popular.

The emergence of new technologies and devices naturally leads to the need to develop new interaction techniques. These new have approaches will have to be more familiar to the generic user, while offering full control of the application. To do so, natural analogies can be used, such as the bimanual manipulation or the use of a pen for sketching. Although the use of touch and calligraphic interaction has been studied in different works, several questions remained unanswered, namely regarding the advantages and disadvantages over each other and over the combined use of both. intuitive.

With the proliferation of the 3D printing technology as well as the high availability of 3D resources online in the recent years, it becomes important to devise 3D modelling methods that allow the common user to be able to create and modify models with ease. Also, as interactive surfaces are becoming widely used in smartphones, tablets, laptops, and intelligent tables and boards, new possibilities arise for the development of 3D modelling solutions that are more natural.

Although much work has already been done to develop professional tools, these require advanced knowledge and training in order to be used effectively, even if the goal is to create an arbitrarily simple shape. Additionally, in recent years we witnessed a significant growth in the field of 3D printing, as the average price of the standard 3D printer has decreased substantially during that period. Taking this into account, it is predictable that 3D printing will gradually be more available to the average user, resulting in the need to develop new 3D modelling methods that allow non-trained users to create and modify 3D models in natural ways. Also, interactive surfaces are currently common in most mobile devices, representing a viable alternative to the standard WIMP (windows, icons, menus and pointer) interaction methodology, allowing fast and yet simple interaction using gestures and multi-touch. For devices that also support it, pen interaction can also provide advantages when combined with multi-touch as it is naturally more familiar for drawing.

Although much work has already been done to develop professional tools, these require advanced knowledge and training in order to be used effectively, even if the goal is to create an arbitrarily simple shape. Likewise, we aim to develop a walk-up-and-use 3D modelling approach focused on 3D fabrication, that takes advantage of methods for 3D object retrieval and the use of interactive surfaces. Also, developing a user interface as a NUI (natural user interface) allows the average user to be able to use the solution in a more proper and natural way. Additionally, this interface can also be complemented by interactive and non-intrusive tools, able to suggest appropriate outcomes given the current state of the application.

The overall idea of this study is to develop a system that enables any user to create and print an object idealized and modelled by him or herself, with an approach that is preferred over the increasingly banal and exclusively tactile interfaces, or the traditional interaction with a single point of contact, such as mouse or digital pens. Therefore, we take as hypothesis that **the combined use of touch and pen allows 3D modelling to be more appealing and accessible to novice users**, when compared to other approaches that use a single contact point for input.

II. Related Work

As the use of 3D models in computer graphics applications grew over the years, so did the need to develop tools to create and modify these models. Although these tools are presented to achieve a common generic goal, naturally, some of them are considered to be more appropriate to be used by experienced users over the others, having a high level of control over a volume's properties. On the other hand, other tools focus on presenting simpler interfaces, targeting a broader user group in which untrained casual users can be included.

A. 3D Modelling Tools

A commonly used tool for modelling is Autodesk's 3ds Max[1]. It features a well built user interface and multiple

[1]Autodesk 3ds Max, http://www.autodesk.com/products/3ds-max/

capabilities for manipulation and animation of 3D models and images.

Much like 3ds Max, Maya[2] is also a modelling tool published by Autodesk, although it is considered to be best fit to create visual effects for animated films. Compared to the previous tool, it presents a complex yet consistent user interface, with easy to understand shortcuts and more options for visualization and appearance modifiers, causing it to be considered a better tool for artists. An important user interface feature is the Hotbox, a customizable and floating menu that is quickly toggled and contains every action from Maya's interface, divided in subsections referred to as marking menus.

As a free and open-source modelling tool alternative to Autodesk's 3ds Max and Maya, there is Blender[3]. Blender provides not only functionality for 3D modelling but also for texturing, rigging, skinning, video editing, tracking and fluid and particle simulation.

Previously owned by Google, SketchUp[4] is also a 3D modelling software, focused on design and modelling for architecture, civil and mechanical engineering. SketchUp is directly connected with 3D Warehouse, an online repository that allows users to search, store and share free 3D models. Although Sketchup's user interface might seem limited for being minimal, it provides fast and precise modelling, aided by dynamic snapping, input of exact values for distance, angle and radius, while hiding the technical mechanics below.

B. *Sketch-based Modelling*

A classic example of sketch-based modelling is provided by Zeleznik et al. [10] through the SKETCH application. In SKETCH, the user creates new geometry through sequences of strokes that define the shape to be created and the details of its form. By representing three perpendicular lines, SKETCH acknowledges the input as a box with sides defined by the sizes of the lines. This method allows the user to create basic shapes very quickly, while being able to change them later. Although the application is simple, the original system provided only a small set of basic shapes, allowing the set of available gestures to remain fairly small, and the gestures themselves iconic of the shapes they represent.

[2]Autodesk Maya, http://www.autodesk.com/products/maya/

[3]Blender, http://www.blender.org/

[4]SketchUp, http://www.sketchup.com/

Fig. 1. Examples of characters created using Teddy.

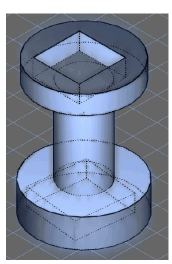

Fig. 2. A model created in GIDeS.

As a different solution for processing the user input, Takeo Igarashi et al. [5] use in Teddy an approach that relies solely on the user's stroke rather than trying to understand what previously stored shape was the user referring to. This approach is named blobby inflation and uses the 2D silhouette drawn by the user to create a 3D inflated version of it. The process starts by finding the central spine of the silhouette and displaces the vertices of the spine plane to form a tessellated mesh dome that is mirrored, creating a symmetric and watertight model that is topologically equivalent to a sphere.

Compared to the previously stated method, blobby inflation is not limited by the size of stored dataset, since it generates new geometry procedurally, not needing to map a sketch into an already known shape. On the other hand, the blobby inflation method provides a lower degree of control, and is more appropriate for shapes with rounder faces (as seen in Figure 1) and with no sharp edges (since those are partially lost due to the inflation).

Fonseca et al. [2] present the GIDeS (Figure 2) system and study how to improve the usability of a CAD system in the early stages of product design, with a simple and learnable approach. To do so, the system provides paper-like interaction through a calligraphic metaphor, dynamic menus to display the state of the application without interfering with the task and the ability to construct precise drawings from sketches through simple constraint satisfaction. In order to keep the user engaged in the modelling experience, the system suggests shapes according to the current progress, supporting the user and providing a more responsive experience.

While using hierarchical implicit volume models (BlobTrees) as an underlying shape representation, Schmidt et al. [9] propose ShapeShop, a sketch-based solid modelling tool capable of reproducing solid models (with arbitrary topology), with complexity ranging from cartoon-like characters to detailed mechanical parts, using multi-touch input. As in Teddy, 2D contours are inflated into rounded three-dimensional implicit volumes. The underlying volume hierarchy is used as a construction history, allowing individual sketched components to be non-linearly edited and removed. Each BlobTree procedurally defines an implicit volume using a tree of primitive volumes and composition operators, such as CSG and blending. ShapeShop includes sketch-based operations for hole-cutting, oversketched blending and adding of surface detail.

Besides inflation, surfaces can also be created using linear sweeps, in which a drawn profile is interactively adjusted with a slider in order to create a controlled extrusion, or using surfaces of revolution (with spherical or toroidal topology), that revolves a sketch around an axis and thereby creating a model. Regarding the sketches, raw polylines drawn by the user are replaced by smooth 2D variational implicit curves to create discrete samples. For refinement, the user can perform gestures to remove or smooth groups of point samples, or to discard whole top layers of a sketch. In order to resolve viewing issues and depth-determination ambiguities, ShapeShop uses a dynamic cutting plane to edit regions obscured by other parts of the current volume.

Later, Lopes et al. [6] also added pen functionality to the ShapeShop solution, aiming to achieve a more fluid modelling experience, with less interruptions. In this new approach, the authors delegate the sketching operations to the dominant hand using the pen device, while manipulation operations are performed with the non-dominant hand via multi-touch input.

This division of labor in human bimanual interaction was studied in early works such as Guiard's [3], in which guidelines for human-computer interaction techniques are proposed, taking advantage of bimanual coordination. The author describes the dominant hand as the more suitable hand for fine movements and manipulation of tools, while the non-dominant can be used for coarse movements and setting the spatial frame of reference.

Newer works like Mockup Builder [1] take advantage of the distinction between dominant and non-dominant hand operations, to create and manipulate 3D shapes with familiar gestures. As in the pen enabled version of ShapeShop, Mockup Builder allows the drawing of shapes using the dominant hand, while handling non-dominant hand gestures as camera and world operations. Using on-and-above-the-surface interaction techniques on a tabletop interactive surface, 3D shapes can be created by using push and pull gestures, creating extrusions of previously drawn profiles. Although the authors were able to achieve satisfactory results, the application required a complex hardware apparatus in order to function.

Aiming at a 3D modelling system for LEGO models, Santos et al. [8] developed LSketchIt, a sketch-based modelling tool that combines calligraphic interaction, a constraint solver and a retrieval mechanism. The user is able to add an object to the scene by drawing it. Then, the 2D sketch is transformed in a 3D sketch using the unprojection of points technique, that converts 2D into 3D points by computing the intersection of a ray from the user point of view with a plane. This 3D sketch is then used as a query, enabling the system to provide a list of results in which the user can select the object to be used.

The system's constraint module is responsible for the creation of connections between parts, propagation of transformations (translations and rotations) and maintenance of the gravity property (when a part is deleted or moved), while the retrieval mechanism relies mainly on the information about dimensions and categories of the parts in the database. The calligraphic interface allows over-sketching of gestures on parts, which enables the refinement of the search. In the following section we will describe other works that focus on 3D object retrieval using sketch-based queries.

Also for LEGO models, Mendes et al. [7] developed an interaction technique for interactive surfaces based on the building block metaphor using multi-touch. With this simplified modelling approach, users are able to elaborate complex LEGO constructions, manipulating the blocks in the 3D space via multi-touch gestures. Authors justified having atomic block transformations instead of combined as it allows a higher degree of control over the operation.

C. Expectation Lists

As described by Igarashi and Hughes [4], it is difficult to precisely model 3D objects that have exact characteristics like symmetries or repeated substructures, using only gestural interfaces. To cope with this problem, Igarashi and Hughes developed a suggestive interface for 3D drawing as part of Chateau, a 3D modeling system. As a shape is drawn in the system (pictured in Figure 3), currently active edges are highlighted (as they are being created or modified), allowing the system to infer possible following operations and present them in a list of thumbnails. If a presented result is desirable, clicking the respective thumbnail completes the drawing as expected. This method of predicting and showing different possible outcomes to the user is often referred to as expectation lists. Chateau, in particular, generates suggestions whenever the user adds, erases, highlights or unhighlights edges, and uses different suggestion engines with mutually exclusive input patterns, preventing the display of an overwhelming number of suggestions.

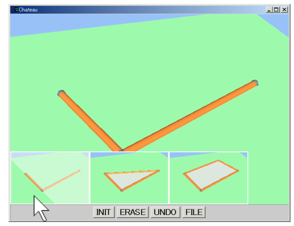

Fig. 3. Expectation list in the Chateau system.

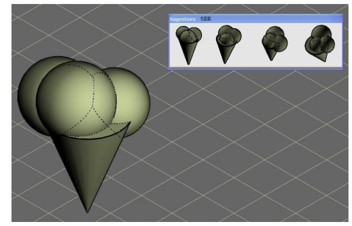

Fig. 4. Expectation list of the GIDeS system.

While dealing with a more complex set of shapes in a 3D object retrieval scenario, it is likely that these shapes will be less different from each other, leading to a bigger number of scenarios of ambiguity while analysing the user's input. To address this problem, Fonseca et al. [2] and Santos et al. [8] also explored the concept of expectation lists, as a non-intrusive context-based dynamic menu that is triggered when the user's strokes are ambiguously recognized, displaying a menu that can be seen in Figure 4. Given that scenario, the user is able to choose from two or more possible interpretations what's the one that better suits his previous stroke stroke.

Schmidt et al. [9] also used expectation lists on ShapeShop to avoid the ambiguity while determining if a sketch should be interpreted as a blobby inflation or as a linear sweep. Instead of small rendered images of what the updated surface would look like for each expectation list icon (heavy on performance), the system uses color-coded representations that also distinguish the creation of new volumes from modifications of the current volume.

III. 3D Modelling on Interactive Surfaces

To create a walk-up-and-use 3D modelling solution, we developed three different interaction techniques that take advantage of touch, pen, and both. In the next subsections we describe the overall modelling, camera and object manipulation functionalities, respectively. Interaction-specific details are described afterwards.

A. Modelling

Given that the aim of this study is to develop an solution that can take advantage of both multi-touch and pen interaction,

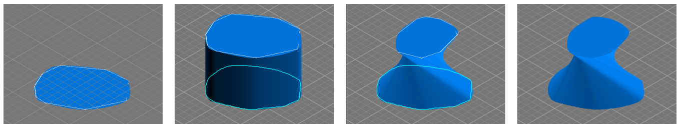

Fig. 5. An example model created using Swivel Extrusion. Starting the extrusion on a closed profile, the user is then able to extrude the shape and manipulate the profile. Tapping finishes the extrusion

a sketch-based approach was deemed appropriate for the 3D modelling component. Similarly to other works presented in section II-B such as Teddy [5], Chateau [4], GIDeS [2] and LSketchIT [8], the modelling process relies on the capture of the user's free-hand draw, that is then analysed in order to determine the desired outcome. In our approach, once a user finishes a stroke and the possible outcomes are computed, the user can be asked to choose a possible transformation through a non-intrusive interface (expectation list).

1) Primitive Shapes: By drawing one or several 2D strokes, the user is able to compose a 2D shape that can later be used for the creation of a 3D model with similar characteristics. To do so, once the user is satisfied with a suggested outcome, the 2D stroke is projected onto the scene's ground plane, locating the stroke in the 3D space to create the new model. The whole purpose of the scene's 2D strokes is to define geometrical boundaries for the creation of 3D objects, therefore serve no purpose by themselves.

2) Swivel Extrusion Tool: As an alternative to the drawing functionality, the Swivel Extrusion method was created. Using Swivel Extrusion, the user is able to create complex extruded shapes with high level of control (an example is shown in Figure 5) based on a closed 2D profile that can be represent anything from an exact straight shape to an arbitrarily shaken free-hand stroke. The user can control the extrusion using two distinct areas of the screen. On both left and right side areas of the screen (about 20% of the screen-size on each side) pointer input is used to control the height level (z-axis) of the closed 2D contour, defining the current height of the extrusion. On the remaining center screen area, pointer input controls transformations to the closed 2D profile. These transformations include rotation, scale and displacement (on the x and y-axis) of the profile, allowing the creation of complex models. All three profile transformations can be individually toggled on or off, since controlling all three transformations at a single given time can be excessively complex. Other control details for the Swivel Extrusion method are dependent of the interaction type and are described ahead in this section.

B. Camera Manipulation

In order to allow a satisfying and fluid experience, it is crucial to provide natural controls for camera manipulation to the user in a 3D modelling context. To allow the user to freely navigate the scene we provide three different camera manipulation options: panning, zooming and orbiting.

Panning actuates by translating the camera on the x and y-axis, while maintaining a constant distance from the scene floor (z-axis is fixed) and the camera rotation. Zooming is achieved by narrowing or widening the camera's FOV (field-of-view), displaying a larger or smaller portion of the scene, therefore creating the illusion that the camera is moving forward or backward. Finally, camera orbiting is done by using both panning and zooming to simulate spatial orbiting around a given point in 3D space. This orbit manipulation only allows the camera direction to be in a vertical anlge between 15 and 90 degrees in reference to the scene floor. Vertical angles below 15 degrees cause problems when trying to cast rays from camera to the scene floor (camera direction becomes almost parallel with the ground plane) while vertical angles above 90 degrees unnecessarily tilt the camera upside down. Although the camera can not be manipulated to be positioned under an object or the scene floor, users can fully inspect an object by manipulating it.

Adjusting the camera's FOV is an unusual method for the manipulation of both camera zooming and orbiting. This approach is used due to the special architecture of the Editor module. As a generic component, the Editor module is based on a generic 2D scene that can contain other 3D scenes, and subsequently 3D scene nodes (objects). Using this architecture, the Editor is not necessarily bound to be a 3D element, becoming more flexible component. Since the camera used in the described solution is fixed directly on the 2D scene as a 2D object, it is not positioned in the z-axis, requiring adjustments on the camera's FOV to simulate zooming or orbiting.

Camera manipulation occurs when a specific gesture is applied directly to the scene, and not on any existing selected object.

C. Object Manipulation

In order to compose complex models, the presented solution allows the basic manipulation of individual models. In this section, we describe the generic information for object manipulation, while in the interaction-specific sections below we describe specific details in regard to each interaction type. An object can be transformed through translation in the 3D space, simple scaling (proportional in all three axes) and individual axis rotation. For consistency purposes, these object transformations are analogous to the camera transformations: pan, zoom and orbit, respectively.

To maintain an unambiguous environment, the manipulation of a given object can only be done if a gesture is applied to it and the object itself is selected. Once an object is selected a transparent sphere is displayed around the object, meaning that the object is ready to receive transformation commands. Naturally, a selected object can also be deselected using the same action (a simple tap gesture). One or more objects can be currently selected at a given moment although transformation gestures only affect the object targeted by them.

As pointer input information is based on 2D screen positions, it can not contain enough information to describe transformation information for all three axes of the 3D space. Therefore, for this solution we chose to define translation and rotation transformations as camera orientation dependent (similar to the camera manipulation in [7]). By doing so, at any given moment, an object can only be translated or

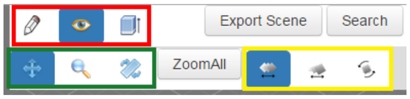

Fig. 6. Set of menus of the Pen-based approach. On the top (marked as red) are located the draw, manipulation and Swivel Extrusion modes. Inside the manipulation mode, the camera manipulations (pan, zoom and orbit) are marked as green and the object manipulations (translate, scale and rotate) are marked as yellow. Camera pan and object translation are the currently selected manipulations.

rotated in a maximum of two axes. The axes to be used for these transformations are decided given the current camera orientation. For example, if the camera direction is more closely aligned with the x-axis (x-axis the most perpendicular axis to the view plane), then the other axis to be considered for the translation and rotation transformations are the y and z-axis. This approach not only avoids overwhelming the user with extra menus or complicated input combinations, as it also allows precise positioning and rotation, in an familiar manner.

D. Pen-based Interaction

Although the focus of this work is the development of new techniques for a sketch-based 3D modelling environment taking advantage of combined pen and touch interaction, it is important to devise and evaluate other techniques to measure and compare its advantages and disadvantages. To do so, we propose a simple pen-based interaction, as used in other previously presented works such as Chateau [4] and ShapeShop [9].

In this pen-based approach for our solution, we focused on defining clearly the different steps of the modelling process, presenting the user with the draw, manipulation and Swivel Extrusion main menus. In the draw section the user is able to specify strokes freely, while accessing the expectation list in order to transform drawn shapes into 3D objects. As described in the generic Section III-A, these transformations occur by transforming 2D drawn shapes into primitive, complex (boolean operations or Teddy-style blobs) or Swivel Extrusion shapes.

While on the manipulation main menu (presented in Figure 6), the user is able to select and manipulate existing 3D objects in the scene, as well as manipulate the camera. By default, both camera and object manipulation settings are set to pan and translation, respectively, allowing the user to freely move a selected object on the designated plane (defined by the camera orientation, as described in Section III-C) by gesturing a stroke starting from the object location to the object's desired final position. On the other hand, performing the same gesture starting over the scene or over an unselected object performs the camera pan manipulation. The selection or deselection of an object is performed by tapping an unselected or selected object respectively.

Also on the manipulation main menu, the user is able to change the default settings for both camera and object manipulation. For camera manipulation, a sub-menu is shown to allow the user to choose between camera panning, zooming and orbiting. Similarly, for the object manipulation the user is able to choose from another sub-menu between object translation, scale and rotation.

Regarding the Swivel Extrusion technique, on both left and right area sections of the screen the user is able to perform vertical pen strokes to raise or decrease the extrusion level. On the remaining center area of the screen, vertical motion of the pen stroke affects the scale of the extrusion profile while horizontal motion rotates the profile in the z-axis (parallel to the ground plane of the scene). Tapping with the pen during the swivel extrusion terminates the method and finalizes the shape, becoming ready for manipulation.

E. Touch-based Interaction

As in the pen-based approach, it is also important to measure how users perform when using an approach solely based on touch interaction. Therefore, a touch-based approach was devised aided by the same main menus used in the pen-based approach (Section III-D), to isolate the free-hand drawing, the camera and object manipulation, and the Swivel Extrusion functionalities.

While on the drawing main menu of the multi-touch interaction, the free-hand sketch modelling can be done by using one or more fingers simultaneously. Naturally, drawing using multiple touch points will generate multiple individual strokes. Although these strokes are initially independent from each other, it is possible to recognize patterns from the combination of the multiple ocurring strokes. For instance, if the user chooses to draw two sides of a square with one finger and the other two sides with a second finger, the recognizers will still be able to acknowledge the both strokes as an intent to create a single square, the same way it would if the user chose to draw the same square with a single continuous stroke. The same principle applies to other recognizable shapes or when closing open sections.

Regarding the manipulation main menu, it allows the user to freely navigate the scene and manipulate objects while using specific touch gestures. All object manipulation gestures are described in Figure 7. Before an object can be manipulated, it has to be selected by performing a simple touch tap gesture

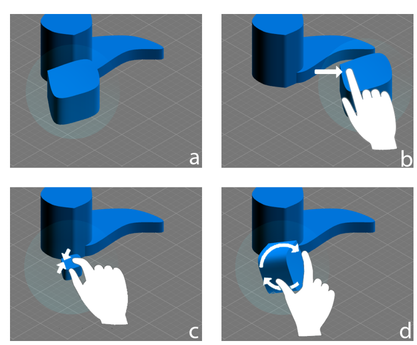

Fig. 7. Object manipulation touch gestures for both the Touch-based and Combined Pen and Touch approaches. a) initial state of the object; b) translation gesture; c) scaling gesture; d) rotation gesture.

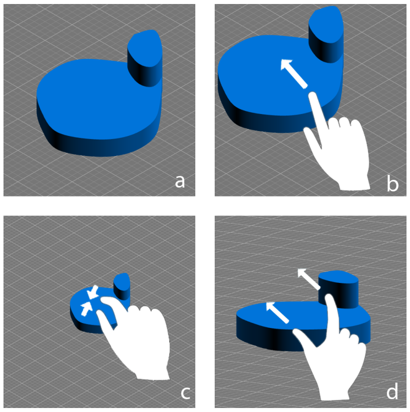

Fig. 8. Camera manipulation touch gestures for both the Touch-based and Combined Pen and Touch approaches. a) initial state of the camera; b) panning gesture; c) zooming gesture; d) orbiting gesture.

over it. A similar touch tap gesture over an already selected 3D object deselects it.

To apply a translation transformation to a selected object, a single-touch drag gesture (starting from the object) allows it to be moved in space and, when the touch is released, the object is fixed in its current location. For the rotation and scaling of a given selected object, once two touch control points are initialized in a selected object, changes in both distance and angle between the two control points determine transformation in scaling and rotation. The object is enlarged if the distance between the two control points increases and is minimized if the same distance decreases. In terms of rotation, the object is rotated in the closest parallel axis to the camera orientation direction.

As described in Section III-C, both the translation and the rotation transformations are dependent of the camera orientation. Also, to maintain both scale and rotation proportions in relation to the two touch control points, these are projected onto the closest perpendicular canonical plane, as differences in distances and angles for screen points are not appropriate to be used as source for the rotation and scaling of three-dimensional points. It is important to note that it is allowed to have multiple selected objects at the same time, and perform simultaneous manipulation operations over a single object (scale and rotation).

Once the manipulation gestures are applied to the scene or non-selected objects, the camera is manipulated. All camera manipulation gestures are described in Figure 8. For camera panning, a single-touch drag gesture pans the camera over the XY plane (ground), maintaining it's distance from the scene ground plane.

Using a touch gesture it is possible to manipulate both camera zooming and orbiting. For both transformations a calculated middle point is used to project a point onto the XY plane, providing a central point for the camera operation manipulations. While zooming, variations in the distance of the dual touch points determines whether the camera is moved forwards or backwards, in relation to the projected center point. On the other hand, for the orbiting, this projected central point acts as the center of the orbit, meaning that by the end of the orbit operation the camera will still be facing it, and at the same distance from it.

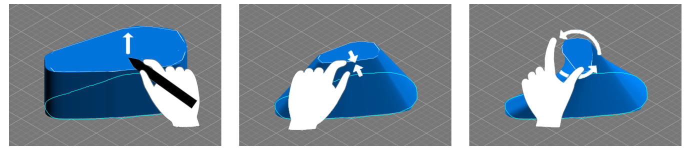

Fig. 9. Using the Swivel Extrusion tool in the Combined Pen and Touch interaction. Level adjustments are done using the pen device and profile manipulation is achieved via multi-touch gestures.

Having a closed stroke and changing to the Swivel Extrusion main menu initializes the method. On this state, the control of the extrusion level can be done with a single touch along either the left or right side of the application window. On the remaining central area, the user is able to use dual touch to apply the same modifications to the extrusion profile. Changes in the distance and finger angle of the dual touch interaction alter the scale and rotation (on the z-axis) of the profile, respectively. Tapping an ongoing extrusion finalizes its progress.

F. *Combined Pen and Touch Interaction*

Tthe combined pen and touch interaction aims to take advantage of the most positive features of both methods. In this approach, pen interaction is delegated to the creation of 2D strokes and extrusion level control, while manipulation transformations and expectation list interaction are controlled by multi-touch input. Since pen and touch interaction can be clearly distinguished, more gesture options arise from the combination of the two. Therefore, in the combined approach, no menus are required and all actions are performed directly in the scene view.

Stroke drawing in the combined interaction is done by using the pen device directly in the scene. As in the previous two approaches, expectation results are shown and allow the user to pick a desired outcome with a touch tap gesture.

Regarding camera and object manipulation, these are directly inherited from the touch-based interaction approach (specifically in the manipulation main menu), resorting to simple tap for selecting/deselecting shapes, single-touch drag and drop for both camera panning and object translation, and multi-touch for camera zooming and orbiting, as well as object scaling and rotation. Likewise, camera and object manipulation gestures are described in Figure 8 and 7, respectively.

Without a menu option for the Swivel Extrusion, the method is initiated once a closed stroke exists in the scene and an upwards pen stroke is performed on either left or right side of the application window. On entering the Swivel Extrusion method, vertical pen strokes adjust the extrusion level (just as in the pen-based approach) while multi-touch interaction controls transformations to the extrusion profile (just as in the touch-based approach). Figure 9 illustrates the extrusion process using this approach.

G. Expectation Lists

As proven in works such as Chateau [4], ShapeShop [9] or GIDeS [2], during a user's modelling process it is important keep the user productively engaged at all times by providing the right tools as they are needed. To do so, expectation lists play a key role in the interactive performance of our solution. As described in the architecture of our solution, recognizers are components that actively search for possible transformations given the current state of the application. These new possible modifications are then used to populate the expectation list that will be presented to the user. In the presented solution, a simple list of animated thumbnails located at the bottom left of the application window. Each element of the list contains an animated three dimensional preview of the respective outcome. The list is also non-intrusive, meaning that the user can simply ignore suggestions as he/she pleases, without any type of repercussions. It is also important to note that the expectation lists should never overwhelm the user with large sets of outcomes, as most of these will probably not be useful in the current modelling context.

Results of retrieval queries also result in possible outcomes for the expectation list. As the user composes an arbitrarily complex model using other simpler shapes, these can be grouped as a single object to be submitted for retrieval. Once the retrieval server replies with the respective results, the expectation is populated with a small set of the ones that most closely resemble the queried example. In the following section we will describe the retrieval process in detail.

H. Retrieval

As it is difficult to produce exact shapes using sketch-based 3D modelling, the integration with a 3D object retrieval system allows a user to specify an object that closely relates to a desired shape and query a database using his/her sketch as an example. With this practice, it is possible to acquire arbitrarily complex shapes by modelling an approximate sketch. Not only is this process faster, but it also simpler. On the downside, it is limited by the set of shapes stored in the retrieval system, as well as by the descriptors used to compute the model indexes.

Although the retrieval component of the application is considered to be an important feature of the overall solution, due to it not being precise in regard to the measurement of similarity values between 3D models, even using the best performing algorithm (D1), we decided to remove the automatic submission for retrieval from the developed solution. Instead, the user is able to select objects to perform an on-demand query whenever he/she deems as appropriate by clicking a menu button outside of the scene view.

IV. Evaluation

We had particular interest in understanding whether users could use object and camera manipulation tools both correctly and naturally. Likewise, we evaluated the solution for all three different types of interaction.

Each user performed the same task using all three approaches, in alternate order. For each user testing the prototype the expected duration for the evaluation session was between 40 and 60 minutes. Before any interaction with the application, useres were presented with a short video explaining how to perform the basic functionalities regarding the interaction type. Then, users were invited to freely use the application in a training session for 2 minutes. After each task execution, users were asked to fill a survey inquiring about the levels of entertainment and difficulty felt during the completion of the task. The task itself required the modelling of a simple desk lamp.

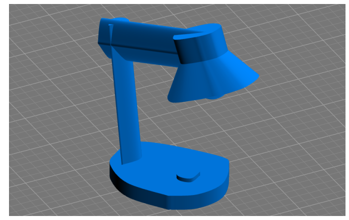

Fig. 10. Example desk lamp model.

A. Modelling task

The proposed task consisted of the modelling of a simple desk lamp (Figure 10 represents an example), comprised of five separate solids: base, column, arm, shade and bulb. Each user received an instructions sheet with five advised ordered steps to build the desk lamp, although they were not forced to follow them, and encouraged to try different orders if considered appropriate. Likewise, users were also not required to perform retrieval searches, although were encouraged to try.

In order to correctly model the desk lamp, users had to model simple cylinders for most of the components (column, base and arm), which could be easily done with simple extrusions. As of the bulb, it could be simply modelled using a basic primitive sphere. The most complex element of the lamp is the shade, modelled by an extrusion with at least three steps, requiring the user to also manipulate the profile of the extrusion. Both extrusions and primitive shape modelling require the user to use the expectation list for stroke closing and primitive shape transformation (from a free-hand 2D stroke).

Regarding camera and object manipulation, as it is needed to compose some objects on top of others, it is required that the user manipulates the camera in order to assure the correct positioning of the parts. Parts such as the arm and the shade also require the user to rotate the objects. Also, since all parts result of free-hand sketch-based modelling, it is also required to perform scale transformation in different objects in order to maintain the correct proportion of the components.

B. Participants and Setup

The evaluation session was attended by 20 users, mostly male, with ages ranging from 21 to 57 years old, although the large majority was below 30 years old. Regarding the education level, most of the users had at least a bachelor's degree. The large majority of the users had vast experience with multi-touch devices while the remainder did not.

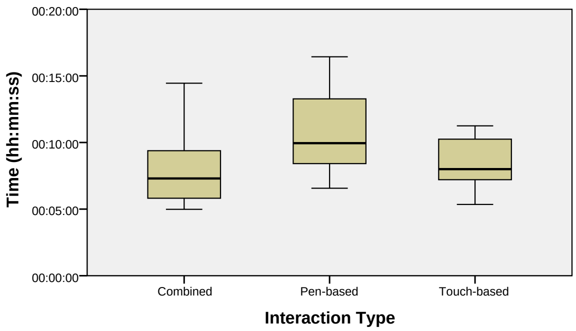

Fig. 11. Overall view of the time to complete the task using the three interaction types. The graphic presents the median, first and third interquartile ranges (boxes) and 95% confidence interval (whiskers).

	Pen-based	Touch-based	Combined
Mean	0:10:48	0:08:50	0:07:49
N	20	19	19
Std. Deviation	0:02:54	0:02:31	0:02:43

Fig. 12. Average task completion times for Pen-based, Touch-based and Combined Pen and Touch interactions.

C. Objective Analysis

For objective analysis we compare the three interaction types separately. Having measured the time taken for each user to perform the task using each interaction type, we averaged those values to obtain a mean value for each interaction.

Applying the Shapiro-Wilk test to the evaluated time samples revealed that only the Pen-based interaction samples are considered normally distributed (significance=.062) while time samples for both the Touch-based and Combined interaction types are not (with significance=.030 and significance=.019, respectively). In addition, the non-parametric Friedman test suggests that there was a statistically significant difference in the time taken by the users to complete the task using the different interaction approaches (χ^2=17.360 p=.001).

Post hoc analysis with Wilcoxon Signed-Rank tests was conducted with a Bonferroni correction applied, resulting in a significance level set at p<.017. These tests suggested that statistically significant differences existed only when comparing the Pen-based approach to the Touch-based (Z=-2.495 p=.013) and Combined (Z=-3.461 p=.001) approaches, and not when comparing the Touch-based approach to the Combined approach (Z=-2.200 p=.028).

In Figure 12 it is visible that the users performed faster while using the Combined Pen and Touch interaction (mean: 7 minutes and 49 seconds), then using the Touch-based approach (mean: 8 minutes and 50 seconds), and finally slower using the Pen-based interaction (mean: 10 minutes and 48 seconds). Figure 11 depicts an overall view of times using the three interaction types.

D. Subjective Analysis

In the task-specific surveys, we asked users to classify the perceived difficulty for the task of modelling the desk lamp as well as for various specific camera and object operations, using a 4 point scale (1-Very Difficult, 2-Difficult, 3-Easy, 4-Very Easy). Likewise, we also measured the perceived fun factor (1-Not Fun, 2-Slightly Fun, 3-Fun, 4-Very Fun) and method fluidity (1-Not Fluid, 2-Slightly Fluid, 3-Fluid, 4-Very Fluid). The Wilcoxon Signed Rank test was used once again to determine whether the differences were statistically significant.

	Fun Factor	Overall Difficulty	Method Fluidity
Pen-based	2.90(1)	2.65(1)	2.65(1)
Touch-based	3.25(1)	2.65(1)	3.00(0)
Combined	3.45(1)	2.75(1)	3.50(1)

TABLE I. User preference for each interaction technique in regard to fun, overall difficulty and method fluidity (Mean, Inter-quartile range). All ranges are measured between 1 (worse) and 4 (better).

	Draw Difficulty	Panning Difficulty	Orbiting Difficulty	Zooming Difficulty
Pen-based	3.75(1)	3.45(1)	3.10(1)	3.40(1)
Touch-based	3.70(1)	3.15(1)	2.60(1)	2.80(1)
Combined	3.65(1)	3.15(1)	2.45(1)	2.65(1)

TABLE II. User preference for each interaction technique in regard to difficulty of camera-specific tasks (Mean, Inter-quartile range). All ranges are measured between 1 (worse) and 4 (better).

	Selection Difficulty	Translation Difficulty	Rotation Difficulty	Scaling Difficulty
Pen-based	3.60(1)	3.10(1)	2.70(1)	3.30(1)
Touch-based	3.50(1)	3.00(2)	2.35(1)	2.80(1)
Combined	3.60(1)	3.10(1)	2.30(2)	2.55(1)

TABLE III. User preference for each interaction technique in regard to difficulty of object-specific tasks (Mean, Inter-quartile range). All ranges are measured between 1 (worse) and 4 (better).

Considering the Table I we can assume that in general, users considered the solution (including all three interactions) to be at least fun and fluid. As for the overall difficulty of the modelling task, participants considered all three interaction types to be relatively similar, averaging a rank between easy and difficult.

In the Tables II and III, the mean values for the difficulties of the specific modelling and manipulation operations are presented. Regarding the mean difficulty of drawing 2D shapes, object selection and translation, the differences are negligible between interaction types. Overall, users considered the both the drawing of 2D shapes and the object selection to be trivial operations, while object translation was considered an easy task.

For orbiting and zooming, users strongly agreed that using the Pen-based interaction camera manipulation operations are easier. Orbiting using the pen is considerably simple when

Pen-based	3.80(0)
Touch-based	2.10(1)
Combined	3.70(1)

TABLE IV. USER PREFERENCE FOR EACH TOOL IN REGARD TO OVERALL UTILITY (MEAN, INTER-QUARTILE RANGE). ALL RANGES ARE MEASURED BETWEEN 1 (WORSE) AND 4 (BETTER).

compared to orbiting using just touch (Z=-2.887 p=.004) or using the combined (Z=-2.982 p=.003) approach. Similarly, zooming is also considered to be simpler using the pen when compared to both Touch-based (Z=-3.207 p=.001) and Combined (Z=-3.441 p=.001) approaches.

When rotating an object, users considered all approaches to be slightly difficult. On the other hand, scaling was considered to be an easier operation overall. Scaling obtained significantly better results in the Pen-based approach when compared with the Touch-based one (Z=-2.324 p=.020).

As described in the previous chapter, Pen-based interaction divides manipulation operations in different sub-menus. By doing so, only one manipulation operation (at most) is occurring at any given time. As of the other two interaction methods, multiple manipulation operations can be active at a given time, allowing the user to simultaneously rotate and scale an object, or zoom and orbit the scene. Likewise, users naturally considered the manipulation with the pen to be easier, as it is atomic and simpler, although the approach itself requires the user to change the manipulation modes several times for a simple transformation.

Regarding the different tools in the solution (described in Table IV), utility was ranked with a 4 point scale (1-Not Useful, 2-Slightly Useful, 3-Useful, 4-Very Useful). Users strongly agreed that both the Swivel Extrusion tool and the expectation list are very useful in a 3D modelling context. Specifically for the Swivel Extrusion tool, it allowed the modelling of most of the parts required for the desk lamp task with minimal interaction and yet accurate precision. As for the expectation list, not only was it considered crucial to transform 2D strokes into primitive 3D shapes, as it also provided useful suggestions for the closing of 2D paths and presentation of retrieval results.

Regarding the Retrieval tool, users considered it to be slightly useful, given the fact that the returned results were most of the times incorrect due to the miscalculation of similarity values in the current implementation of the EnContRA server. However, most of the users agreed that having a functional 3D object retrieval system as part of a 3D modelling application could be substantially useful.

E. Observations

In regards to object selection, some users revealed some difficulty while picking an object, as this object has to occupy a considerable amount of screen space to be pickable. To counter this, a user suggested the creation of a box selection tool, capable of selecting surely one or multiple objects inside a user defined area at the same time. Figures 13 and 14 depict users performing the task during the evaluation process.

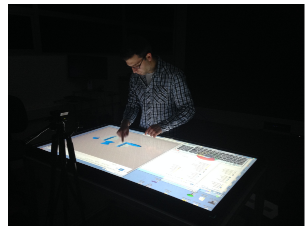

Fig. 13. User manipulating a shape using multi-touch.

Fig. 14. User drawing with the pen device using the combined pen and touch approach.

While using a Combined Pen and Touch approach on the interactive tabletop, some users described the object and camera manipulation as a difficult task when performed using multi-touch gestures with a single hand. Therefore, these users felt more comfortable putting the pen device aside while making complex object and camera manipulations and using both hands (as in the Touch-based interaction, although some hadn't yet performed the task using this interaction type), and then picking the pen up again when needed. Regarding the Touch-based and Pen-based interactions, the menus were considered to be less intuitive, as they often required the users to learn what button was associated to the desired transformations. Also, the method is purely based on gestures and interaction type, with the lack of visual references some users struggled initially, having trouble reminding how to perform a certain manipulation or action.

When using the Swivel Extrusion Tool, tapping to end the extrusion process can be a problem as unwanted taps can occur. Some users suggested the introduction of a confirmation menu to ensure that the user is about to finish the extrusion, while others suggested the use of a more complex gesture instead of a simple tap.

In general, users who chose to design all the pieces first and then assemble them later were able to perform the task faster. Performing similar tasks sequentially required fewer changes between menus and between two-hand and dominant hand interaction.

Finally, users that were not familiar with touch-based applications suffered from major difficulties while trying to perform the task using the Touch-based and Combined interactions. In this special situation was the 57 year-old user who only had little experience with computers in general, and was only able to finish the task using the Pen-based approach (the one that most closely relates to the mouse interaction), having given up while performing the task using the other two interactions, mainly because of difficulties while trying to use the multi-touch to manipulate the camera and the objects.

V. CONCLUSIONS

Interfaces based on the WIMP paradigm (windows, icons, menus and pointer) have a long history in the field of user-centered design, being used for most applications' user interfaces, including 3D modelling tools. However, as tactile surfaces are increasingly more common in every-day devices, it becomes necessary to devise new and more natural interaction techniques that can take advantage of those surfaces. Additionally, 3D fabrication has become a popular subject in the recent years, propelled by the high availability of 3D resources online.

Over the recent years, several works have been presented in order to solve common interaction problems when modelling using interactive surfaces. Although most presented satisfying results, they could not be generalized to all modelling scenarios, particularly for 3D fabrication.

In this study, we proposed to identify the advantages and disadvantages of different natural interaction types in a 3D modelling context, that explore the use of interactive surfaces, multi-touch and/or calligraphic interaction. Regardless of the interaction type, it was intended to develop a solution that can be used by both experienced and novice users, requiring only a brief learning process.

To do so, three main interaction types were devised: Pen-based, Touch-based and Combined Pen and Touch interaction. For each interaction type, different methods are used in order to create and transform shapes, as well as manipulate both the objects and the view. These interactions were also evaluated with a set of users, in order to assess the differences between them, measuring the performance and the usability of the different sub-operations of the solution. Additionally, users evaluated generic tool components such as the Swivel Extrusion tool, the expectation list and the 3D object retrieval system.

Analysing the results of the evaluation with users revealed that although users were not as familiar with the Combined and Touch-based approaches, they were able to perform the task faster using the Combined, then the Touch-based and finally with the Pen-based approach. This relation is also consistent with the menu complexity level between approaches, as the Pen-based uses two levels of menus, while the Touch-based uses only one level, and finally the Combined approach uses no menus. These results are coherent with others present in related works, regarding the levels of naturalness and menu complexity between interaction types.

Considering all the above, we were able to develop a solution that supports three different interaction types, enhanced by non-intrusive interactive tools, such as the expectation lists, 3D object retrieval functionality and the Swivel Extrusion. The evaluation phase revealed that users, regardless of having experience with other modelling tools, are able to autonomously create 3D models (that are production ready and can be later exported, stored or printed) with satisfying results in regard to the perceived difficulty level and time taken to complete the task.

A. Future Work

After measuring the results of the user testing, we consider that some solution details are worthy of improvement. Some of them are:

a) 3D Object Retrieval:: Although it was not considered to be the main focus of the developed application, the 3D object retrieval component was an important part of the solution. To improve it, it is necessary to inspect and correct the current implementation of the EnContRA Server, specifically the computation of shape descriptors and evaluation of similarity values.

b) Multi-touch Gestures:: As some users stated, using a simple tap gesture to end an extrusion can lead to it being ended unwillingly early (due to unwanted interaction). Instead, alternatives must be found to substitute these overly simple gestures.

c) New tools for modelling:: After obtaining good results in the user evaluation phase, it became clear that users feel engaged while using familiar and simple tools to create complex shapes, such as the Swivel Extrusion tool. As a follow-up for the Swivel Extrusion, we considered the creation of a similar simple tool to model a surface revolution, taking advantage of the multi-touch interaction.

REFERENCES

[1] Bruno R. De Araùjo, Géry Casiez, and Joaquim A. Jorge, *Mockup builder: Direct 3d modeling on and above the surface in a continuous interaction space*, Proceedings of Graphics Interface 2012 (Toronto, Ont., Canada, Canada), GI '12, Canadian Information Processing Society, 2012, pp. 173–180.

[2] Manuel J. Fonseca, Alfredo Ferreira, and Joaquim A. Jorge, *Towards 3d modeling using sketches and retrieval*, Proceedings of the First Eurographics Conference on Sketch-Based Interfaces and Modeling (Aire-la-Ville, Switzerland, Switzerland), SBM'04, Eurographics Association, 2004, pp. 127–136.

[3] Yves Guiard, *Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model*, Journal of Motor Behavior 19, 1987, pp. 486–517.

[4] Takeo Igarashi and John F. Hughes, *A suggestive interface for 3d drawing*, Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology (New York, NY, USA), UIST '01, ACM, 2001, pp. 173–181.

[5] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka, *Teddy: A sketching interface for 3d freeform design*, Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (New York, NY, USA), SIGGRAPH '99, ACM Press/Addison-Wesley Publishing Co., 1999, pp. 409–416.

[6] Pedro Lopes, Daniel Mendes, Bruno Araújo, and Joaquim A. Jorge, *Combining bimanual manipulation and pen-based input for 3d modelling*, Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling (New York, NY, USA), SBIM '11, ACM, 2011, pp. 15–22.

[7] Daniel Mendes, Pedro Lopes, and Alfredo Ferreira, *Hands-on interactive tabletop lego application*, Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology (New York, NY, USA), ACE '11, ACM, 2011, pp. 19:1–19:8.

[8] Tiago Santos, Alfredo Ferreira, Filipe Dias, and Manuel J. Fonseca, *Using sketches and retrieval to create lego models*, Proceedings of the Fifth Eurographics Conference on Sketch-Based Interfaces and Modeling (Aire-la-Ville, Switzerland, Switzerland), SBM'08, Eurographics Association, 2008, pp. 89–96.

[9] Ryan Schmidt, Brian Wyvill, Mário Sousa, and Joaquim Jorge, *ShapeShop: Sketch-Based Solid Modelling with BlobTrees*, EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling (2005).

[10] Robert C. Zeleznik, Kenneth P. Herndon, and John F. Hughes, *Sketch: An interface for sketching 3d scenes*, Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (New York, NY, USA), SIGGRAPH '96, ACM, 1996, pp. 163–170.