***NAME: M SAMEER ZAHID***

***REG ID: 2480***

***DATE: 12-12-2013***

***ASSINGMENT: GENERATING 3D MODELS FROM 2D GEODATA.***

***PROGRAM: ELECTRONICS***

***SESSION: 8:30 TO 11:30 (THURSDAY)***

***GENERATING 3D MODELS FROM 2D GEODATA***

**ABSTRACT:**

We present a system for 3D modeling of free-form surfaces from 2D sketches. Our system frees users to create 2D sketches from arbitrary angles using their preferred tool, which may include pencil and paper. A 3D model is created by placing primitives and annotations on the 2D image. Our primitives are based on commonly used sketching conventions and allow users to maintain a single view of the model. This eliminates the frequent view changes inherent to existing 3D modeling tools, both traditional and sketch-based, and enables users to match input to the 2D guide image. Our annotations—same-lengths and angles, alignment, mirror symmetry, and connection curves—allow the user to communicate higher-level semantic information; through them our system builds a consistent model even in cases where the original image is in consistent. We present the results of a user study comparing our approach to a conventional “sketch-rotate-sketch” workflow.

**INTRODUCTION:**

Traditional 3D modeling tools (e.g. [Autodesk 2009]) require users to learn an interface wholly different from drawing or sculpting in the real world. 2D drawing remains much easier than 3D modeling, for professionals and amateurs alike. Professionals continue to create 2D drawings before 3D modeling and desire to use them to facilitate the modeling process ([Thor mahlen and Seidel 2008;Tsang et al. 2004; Eggli et al. 1997; Kallio 2005; Dorsey et al. 2007;Bae et al. 2008]). Sketch-based modeling systems, such as Teddy [Igarashi et al. 1999] and its descendants, approach the 3D modeling problem by asking users to sketch from many views, leveraging users’ 2D drawing skills. In these systems, choosing 3D viewpoints remains an essential part of the workflow: most shapes can only be created by sketching from a large number of different views. The workflow of these systems can be summarized as “sketch-rotate- sketch.” Because of the view changes, users cannot match their input strokes to a guide image. Moreover, finding a good view for a stroke is often difficult and time consuming: In [Schmidt et al. 2008], a 3D manipulation experiment involving users with a range of 3D modeling experience found that novice users were unable to complete their task and became frustrated. These novice users “positioned the chair parts as if they were 2D objects.” The change of views is a major bottleneck.

Sketching is also used in the context of traditional modeling systems: a workflow often employed by professional 3D modelers is placing axis-aligned sketches or photographs in the 3D scene for reference. This workflow could potentially allow amateurs who cannot draw well in 2D to create 3D models from sketches produced by others. Yet, paradoxically, this approach requires a higher level of skill despite relying on easier-to-produce 2D sketches as a modeling aid. This is because of the difficulty of using conventional tools, which require constant changes to the camera position, whereas a single view is needed to match an existing image. The goal of our work is to design a user interface that simplifies

Modeling from 2D drawings and is accessible to casual users. Ideally, an algorithm could automatically convert a 2D drawing into a 3D model, allowing a conventional sketch (or several sketches) to serve as the sole input to the system. This would eliminate the need for view point selection and specialized 3D UI tools. However, many (if not most) drawings are ambiguous and contain inconsistencies, and cannot be interpreted as precise depictions of any 3D model (Section 3). This limits the applicability of techniques such as Shape-from-Shading ([Prado’s 2004]) and reconstruction from line drawings ([Varley and Company 2007]). Humans apparently resolve many of the ambiguities and inconsistencies of drawings with semantic knowledge. Our work provides an interface for users to convert their interpretation of a drawing into a 3D shape. Instead of asking the user to provide many sketches or sketch strokes from multiple points-of-view, we ask the user to provide all information in 2D from a single view, where she can match her input to the underlying sketch. In our tool, user input takes the form of (1) primitives (generalized cylinders and ellipsoids) with dynamic handles, designed to provide complete flexibility in shape, placement, and orientation, while requiring a single view only, and (2)

Annotations marking similar angles, equal-length lines, connections between primitives, and symmetries, to provide additional semantic information. Our system generates 3D models entirely from this user input and does not use the 2D image, which may be inaccurate, inconsistent, sparse, or noisy. We do not expect that users have a consistent 3D mental model of the shape and are specifying primitives precisely; we aim to create plausible, reasonable quality 3D models even if a user’s input is inconsistent.

**HISTORY:**

These days, it is difficult to find a television documentary detailing an archaeological site that does not feature a representation in the form of a 3D model. Computer models make good teaching tools. A class of students may not have the opportunity to travel to Rome to view the Colosseum first-hand, and even if they did, they would have great difficulty visualizing what the mostly-ruined structure looked like 1,900 years ago. A model based on the most recent archaeological research, however, can help fill in the gaps left by time and the elements.

One of the more important aspects of a computer model is that it is dynamic. Using software, a model can be adjusted to reflect newer theories of the site’s architectural reconstruction. This is certainly a stark contrast to artists’ sketches and paintings, which, over time, tend to become outdated. Importantly, like other visualization methods used in the humanities (such as GIS), 3D models can help scholars get a fuller picture of a site and formulate research questions that never would have been considered otherwise.  This is the case in my most recent research.

Having never truly given up on the video game design aspirations of my high school days (I specifically remember my father turning the breaker off to the upstairs when I was up until 4 AM designing a Quake map), I have found a niche within my field of academic interest—Roman archaeology and architectural history. While many of my Pompeian’s classmates take a more traditional approach to graduate research projects, I chose to develop a 3D model of the House of the Faun, one of the largest and most famous houses in the city. The model was constructed as accurately as possible based on the archaeological plan, a number of artists’ reconstructions, and photographs of the house (many gathered from Flickr).

The intent of the model was to test art historians’ philosophical assertions about Roman atrium houses.  With accurate lighting simulation (i. e., calibrating a simulated sunlight to the latitude and longitude of the house and to any point in time back to antiquity), high resolution images of the model rendered by Mental ray software gave me a glimpse of what the House of the Faun looked like at noon on January 1st, 100 B.C., which is something no artist can replicate.

Coincidentally, lighting simulation may have an impact on how we consider the artwork within the house. For example, when many art historians point to the colors of a mosaic as being proof of its Greek influence, can that assertion bear the burden of the fact that the mosaic was rarely in sunlight?

Many of us have seen Roman floor mosaics hanging on the walls of American and European museums, but they have been removed from their original context. Even in Pompeii, one of the best-preserved sites of the ancient world, the roofs collapsed long ago, making it difficult to visualize the natural lighting scenario within the House of the Faun and other structures within the city. 3D models allow us to put artworks back in their original context and consider how the ancients viewed them, which is quite different from how we view them now. In this case, the computer model is more than just a teaching tool; it is a scholarly research tool.

**INFORMATION:**

Our approach is most similar in spirit to [Zhang et al. 2001] and [Wu et al. 2007], in which users annotate a single photograph or drawing with silhouette lines and normal and positional constraints; the systems solve for height fields that match these constraints. In our system, the primitives and annotations added by a user are structured and semantic, and we are able to generate 3D models from globally inconsistent drawings. Our system rectifies the shape primitives placed by a user in order to satisfy the user’s annotations (symmetries and congruencies).[Andre et al. 2007] presented a single-view modeling system in-tended to mimic the process of sketching on a blank canvas. We are similarly motivated; our system allows users to preserve intact the process of sketching on a blank canvas.

Interactive, multiple-view modeling. In [De-bevec et al. 1996] and [Sinha et al. 2008] and [van den Hengel et al. 2007], users mark edges or polygons in multiple photographs (or frames of video).

The systems extract 3D positions for the annotations, and, in fact, textured 3D models, by aligning the multiple photographs. (In [De-bevec et al. 1996], users align them to edges of a 3D model created in a traditional way). In our system, users have only a single, potentially inconsistent drawing; these computer vision-based techniques assume accurate, consistent input and hence cannot be applied to our problem.

Automatic, single-view sketch recognition.

Sketch recognition techniques convert a 2D line drawing into a 3D solid model. These

Approaches also typically assume a simple projection into the image plane. Furthermore, a variety of restrictions are placed on the line drawings, such as the maximum number of lines meeting at single

Point and the implied 3D models are assumed to be, for example, polyhedral surfaces. For a recent survey of line-drawing interpretation algorithms, see [Varley and Company 2007]. One notable recent work in this direction is [Chen et al. 2008], which allows forum precise, sketched input by matching input to a domain-specificdatabase of architectural geometry. More relevant to free-form

modeling, the recent works of [Karpenko and Hughes 2006] and[Cordier and Seo 2007] generate 3D models from a single view’sfree-form visible silhouette contours. These works are primarily

concerned with generating surfaces ([Karpenko and Hughes 2006]) or skeletons ([Cordier and Seo 2007]) correctly embedded in R3 with visible contours matching the user’s input. They do not represent modeling systems per se, but rather a necessary componentfor any system taking silhouette contours as input. Our approach can be viewed as a form of user-assisted 2D-to-3D interpretation.

Because a human uses our tool to annotate the 2D image, we are able to receive user input that eliminates ambiguity and rectifies inconsistencies in the image.

Interactive 3D modeling.

There are a variety of sketch-based modeling tools based on the concept of sketching curves from various angles (sketch-rotate-sketch). The earliest of these is [Igarashi et al.1999], and this direction has been explored in a variety of later works. A good overview can be found in the recent survey of [Olsen

et al. 2008]. These works assume users are capable of sketching a model from multiple points-of-view, and that users can find good views for sketching and manipulating the model. As such, users

Cannot trace a guide image. We do not assume such skill, and believe that that rotation and sketching from novel views is the most difficult aspect of these systems.

The work of [Cherlin et al. 2005] deserves further mention. The goal of this work is minimizing the number of strokes a user must draw to create a surface. They introduce “rotational blending sur-

faces,” which are similar to our generalized cylinders. However,these surfaces’ “spines” are planar xy

curves unless over-sketched from a rotated view.

Generalized cylinders have been used extensively in the blobby modeling literature ([Bloom enthal 1997]). The Meta Reyes 3Dmodeling system ([Info graphic a 2009]) makes extensive use of a

Generalized cylinder primitive with non-circular cross sections and is based on implicit surfaces; our generalized cylinder primitive could be implemented similarly. Meta Reyes is not designed to be a

single-view modeler; as a result, users cannot easily match a guide image.

Two early sketch-based modeling systems relevant to our work are [Zeleznik et al. 1996] and [Eggli et al. 1997]. [Zeleznik et al.1996] introduced SKETCH, a gestural interface for 3D modeling.

In SKETCH, users are capable of performing most modeling operations from a single view. For this reason, SKETCH could almost be re-purposed as a tool for annotating existing 2D drawings. However, inconsistencies commonly present in 2D drawings preclude this application of SKETCH. In addition, SKETCH only supports a subset of CAD primitives and cannot be used for free-form modeling. [Eggli et al. 1997] introduced constraints for beautifying users’imprecise, sketched input. [Cabral et al. 2009] presented an interactive system for deforming and reshaping existing architectural models with length and angle constraints. We, too, use constraints,although our constraints are also designed to reconcile globally inconsistent user input.

The systems presented in [Kallio 2005; Dorsey et al. 2007; Bae et al. 2008] allow users to sketch freely with 2D input devices, projecting users’ strokes onto various planes in space. These sys-tems are designed to support the early conceptual design phase of aproject; as such, they do not create surfaces, but rather a collection of strokes in space. Our interface can be used to create a 3D surface from a sketch drawn during a traditional 2D conceptual design

CONCLUSION:

We have presented an interface for 3D modeling based on the idea of annotating an existing 2D sketch. Although many 2D drawingshave no consistent 3D representation (Section 3), with a small

degree of training, even novices are able to create 3D models from 2D drawings. Our interface eliminates the need for constant rotation inherent to many previous sketch-based modeling tools, which precludes users from matching their input to a guide image. Our primitives and annotations are structured and persistent, and provide semantic information about the output models useful in a variety of applications. An additional benefit of our approach is that the entire modeling process is visible in a single, static 2D image, which makes it easy to explain and learn how to create a given model.

While we are able to create a variety of models using our existing primitives and annotations, our approach is not without its limitations. First, our existing primitives are limited to free-form surfaces. We cannot model surfaces with edges or relatively flat surfaces. Second, our interface cannot be used when a drawing’s point-of-view aligns with primitives’ spines or long axes, projecting them

to (or nearly to) a single point. Third, our interface is not appropriate for adding fine-scale details. For this task, models created in our system can be refined with a displacement editor. Fourth, it is

not possible to do away with a 3D view altogether. The 3D view is necessary for verification; it is consulted interactively when manipulating out-of-image-plane tilt handles and connection curve annotations. Fifth, our interface has operations which feel like modeling rather than sketching, and it takes some training to learn the 2D-o-3D mapping. Sixth, we provide no way to color or texture 3D

models, even though drawings may have been colored. Seventh,we do not allow cycles of connection curves, which are useful insome cases.