



A Procedural Approach to Computer-Aided Modelling in Nautical Archaeology

8

Matthew Suarez, Frederic Parke, and Filipe Castro

Abstract

This chapter analyses the functionality and applicability of procedural computer-based modelling techniques in the field of nautical archaeology. To demonstrate this approach, an interactive procedural model of the lower hull timbers of a sixteenth-century European merchant ship was developed through a process of prototype implementation, and an evaluation of the usefulness and effectiveness of the prototypes developed was carried out using the timbers from the hull remains of the Belinho 1 shipwreck, found in Portugal in 2014. The 3D model was created using *Houdini*, a procedural node-based 3D software package. A basic collection of the main timber components of a ship's lower hull was defined, and functional rules were created for each timber, based on real-world ship design and construction processes. Then the rules were incorporated into the logic of the procedural modelling algorithm, and the resulting model was changed by using the Belinho 1 shipwreck scantlings. The results, which will be discussed, were satisfactory, except for the planking, which is a very complex part of the shipbuilding process and deserves future attention.

Keywords

Procedural modelling · Ship and boat archaeology · Shipwreck · Research models

8.1 Introduction

The collaboration between the J. Richard Steffy Ship Reconstruction Laboratory (ShipLAB) and the Department of Visualization (VizLab) started a decade ago and since then the two institutions have discussed and developed several projects pertaining to the modelling of archaeological ship remains. Alexander Hazlett (2007) sought to synthesize the data from various sources on the subject of the Portuguese *nau* to create a timber-by-timber model, following a manuscript which enumerated the timbers necessary to build a merchantman, or *nau da India*. From the model, developed in *Rhinoceros 3D*, Hazlett created, annotated, and illustrated construction diagrams of this ship, and hypothesized a construction sequence (Fig. 8.1). Audrey Wells (2008) created a detailed reconstruction of the Portuguese vessel *Nossa Senhora dos Mártires*, lost near Lisbon in 1606, following a hypothetical reconstruction (Castro 2003, 2005, 2009; Castro et al. 2010). This reconstruction could be viewed using a real-time immersive visualization system (Fig. 8.2). Justus Cook (2012) created a parametric computer model of an Iberian *nau* or *nao* hull, and varied the main dimensions according to a set of measurements from historical manuscripts, in order to better understand the changes in the hull shapes (Fig. 8.3).

The goal of this project was to reduce the large investment of time and expertise that is currently required to create 3D reconstruction models for nautical archaeological research using typical modelling methods. The strategy is an approach that leverages computer-based modelling, both parametric and rule-based. To demonstrate this, a procedural model of the lower hull of a sixteenth-century European merchant ship was developed through an iterative process of prototype implementation. The resulting model was flexible and versatile, and could be iterated through parametric controls, greatly reducing the traditional change and revision time.

The results of this project provided evidence of the time-saving effectiveness of a procedural approach to create 3D

M. Suarez (✉) · F. Parke
Department of Visualization, Texas A&M University,
College Station, TX, USA
e-mail: jlcasaban@tamu.edu; frederic-i-parke@tamu.edu

F. Castro
Department of Anthropology, Texas A&M University,
College Station, TX, USA
e-mail: fvcastro@tamu.edu

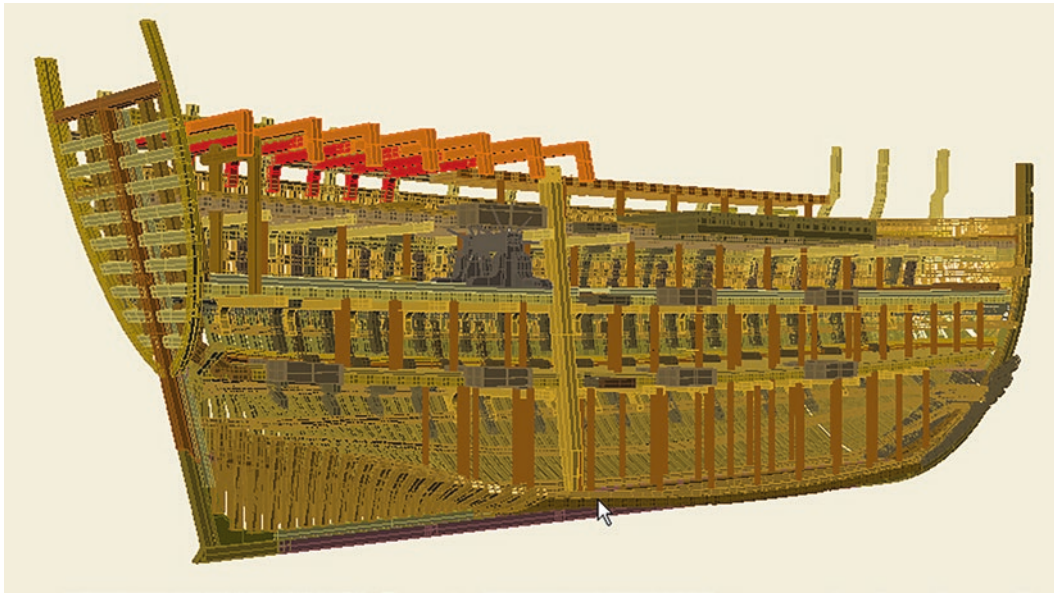


Fig. 8.1 Alexander Hazlett's model

models as research tools. Once procedural models are created, they provide both accessible and powerful means for researchers to create and test multiple interpretations of the archaeological data. Although the development of a procedural model requires skilled computer operators and a significant investment in design, construction, and trouble-shooting, these problems are outweighed by the flexibility offered by the models.

8.2 Computer-Aided Modelling in Archaeology

Computer-aided modelling offers archaeological researchers quick 3D images of their reconstructions of the past, and a perspective through which they can analyse archaeological data. Such models allow visualizations of the data collected from archaeological sites, including shipwrecks. Models offer researchers an opportunity to view what is left of the remains without the constraints of low visibility, poor lighting, and the partial observations inherent to the excavation process: archaeologists see their sites through layers and trenches. As it is highly destructive to expose an entire site all at the same time, marine archaeologists tend to dig and record partial areas, which are quickly covered—or recovered—as soon as they have been cleaned, tagged, and recorded.

These partial views—of partially preserved shipwreck remains—make it difficult to identify or understand any particular shipwreck. Cargo, construction features, and materials employed are important clues, but a 3D view of a shipwreck site, which can be scaled—zoomed in and out—provides a good tool for the interpretation and reconstruction of a shipwreck.

In any given time period, ship shapes and hull structures varied from region to region. This variation means that a ship's hull shape can give an indication to its provenience. Once an archaeologist has a basic idea of what kind of ship she is excavating, the mental reconstruction process begins and influences the recording process. Ship reconstruction is an iterative and interpretive process, continually evolving as evidence is uncovered. To assist in visualizing the data collected, researchers will often create models; hand-drawn, physically built, or computer based. The benefit of creating models is that they will often unveil or expose patterns in the data that were previously unclear or even unseen. Models can also expose oversights and misinterpretations of the data. Iterating upon these models can also be used to test hypothesis and explore alternatives. There are a number of benefits to creating computer-based 3D models. For instance, model precision is maintained by the computer and therefore models are subject to fewer opportunities for human error and fatigue. The scalability of 3D computer models allows them to be constructed at full scale. Moreover, 3D modelling also allows for the automation of redundant tasks. Another advantage is that computer processing power allows larger quantities of data to be included in a model.

The main drawback of typical 3D modelling techniques is the large overhead of expertise and knowledge in 3D modelling and 3D modelling software required to create a good 3D model. A second problem is that once a model is created, revisions and iterations can be time consuming. These aspects make one-off production computer models less than ideal for the ship reconstruction iterative process, and that is a deterrent for researchers wanting to implement 3D research models.

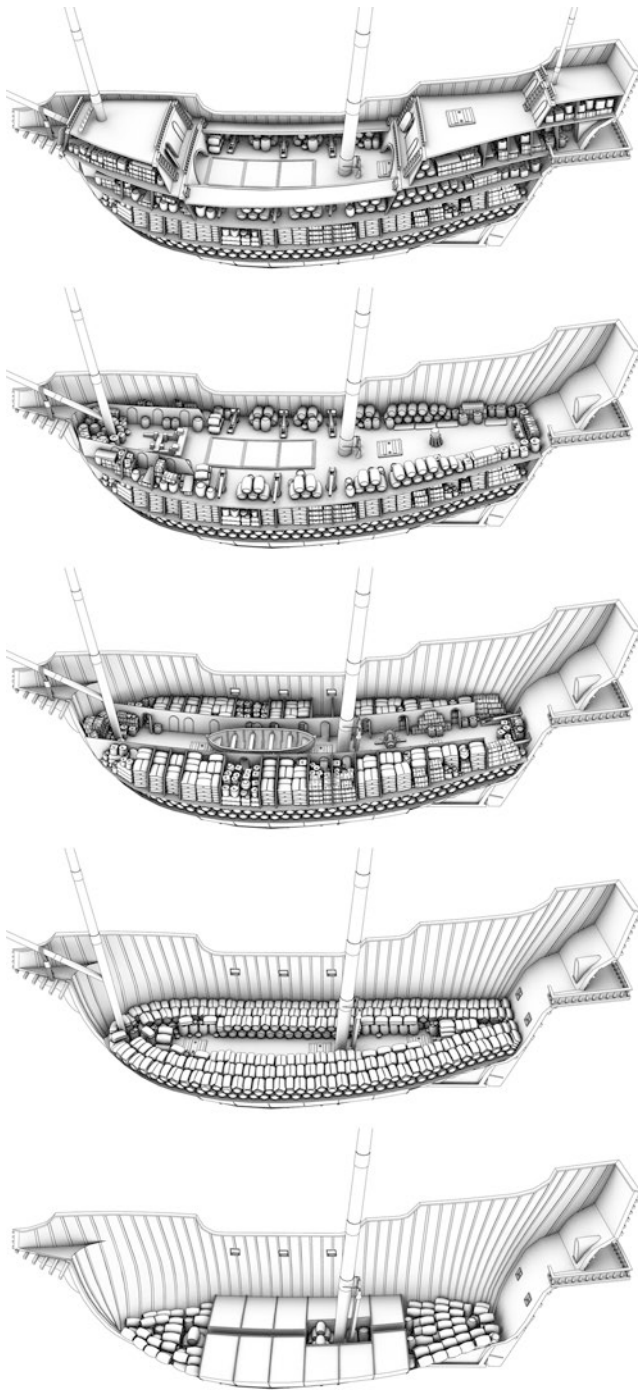


Fig. 8.2 Audrey Wells' model

8.3 Computer-Based Modelling in Archaeology

During the 1960s and 1970s computers were mainly used in archaeological research for statistical applications such as classification and seriation, archaeological techniques which predate computers. One of the first textbooks published on the topic of computers in archaeology, *Mathematics and*

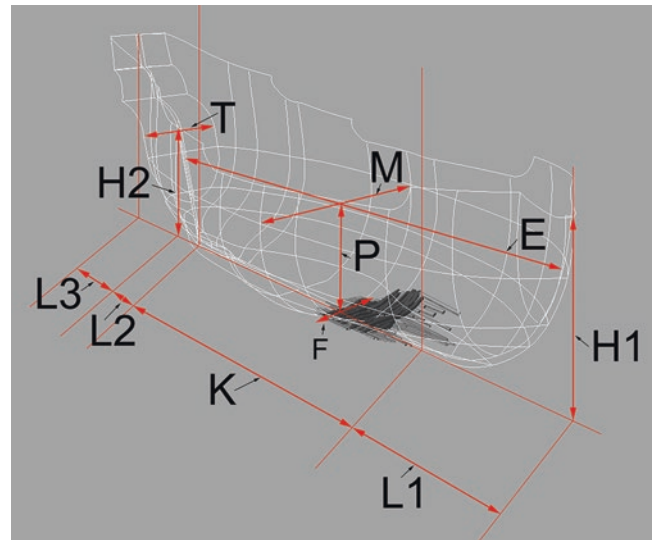


Fig. 8.3 Aspect of Justus Cook's model

Computers in Archaeology (Doran and Hodson 1975), focuses on the application of data classification and quantification in archaeological research. The use of computers at the time was limited, due to their cost and low accessibility. Most computers were only available at universities or other large institutions.

The advent of the microprocessor in 1971 and the invention of the first microcomputer in 1975 reduced costs and facilitated access, and by the late 1970s computers were integrated into most areas of archaeological work. According to a 1986 survey of computer usage in British archaeology, computers at the time were mainly focused on theoretical tasks (Richards 1986, 2). These were tasks, not based on theory, but rather on using computers to automate tasks like managing and processing large amounts of data (Lock 2003, 1). The rapid development of graphics, computer-based visualization, and computer software in the mid-1980s and through the 1990s brought about the modern integration of computers into archaeology.

Computer applications within archaeology at this point were now multimedia; integrating the use of text, images, models, animation, and sound. Computers also allowed opportunities to cross-link all this information into different contextual situations. The cross-linking of information encouraged a new data-driven exploratory method of archaeology which we see today (Lock 2003, 211).

8.4 Computer Models

A model is defined as 'a simplification of something more complex to enable understanding' (Lock 2003, 6). In *Models in Archaeology* Andrew Fleming and David Clarke (1973) described models as 'ideal representations of observations

which are heuristic, visualizing, comparative, organizational, and explanatory devices.’ With this definition, the authors support multiple interpretations and open the possibility of more than one model for any one situation, because they are ‘not “true” but a part of the hypothesis generation and testing procedure’ (Fleming and Clarke 1973, 316–317).

Prior to the mid-1970s, an archaeological model was either a 2D orthographic set of drawings or a physical reconstruction. These were the traditional methods of exploring and recording ship dimensions. Up until this point, digital developments had been essentially methodological. Computers and computer models provided tools that were considered a theoretical, meaning that they were not intended to explore or inspire interpretations but rather to measure and document existing data (Evans and Daly 2006, 11). In this context, computers and computer models were not yet used to inspire or explore alternative ideas. Once this capacity was developed, computers became tools capable of influencing the creation of theory, adding to the traditional use of 2D drawings or physical models (Haegler et al. 2009, 1). One of the opposing arguments to this view was summarized by Miller and Richards (1995), who claimed that each project was the result of the collaboration between computer scientists and archaeologists, rather than being archaeologically controlled. In most cases the visualization software itself was not accessible to the archaeologists and therefore the computer scientists were perceived as a filter between archaeologists and their data. In other words, archaeologists did not have direct control of the modelling themselves (Miller and Richards 1995, 20).

It is only recently that this 3D software gap of accessibility has been narrowed. Computer-based 3D modelling software is now widely available and widely utilized in most universities and institutions of research. Our research proposes an approach which could further narrow the accessibility gap by leveraging parametric user-interaction and procedural modelling to facilitate the use of computer-based modelling as an exploratory theoretical tool.

The questions raised by Roger Hill (1994; Yamafune et al. 2016), remain more relevant than ever, pertaining to the necessity to understand the relation between the archaeological remains, created by a range of dynamic processes, which he called ‘a database in which an imperfect memory of those processes is retained,’ and the simplified computer models we use to try to understand and reconstruct the past they represent.

8.5 Procedural Modelling

‘Procedural modelling’ is a general designation for techniques in computer graphics which create 3D models or textures from a set of rules. L-Systems, fractals, and generative

modelling are all included in this family of techniques. For our work, the term procedural modelling refers to the creation of 3D models through rules which are configurable by parameters. In a paper titled ‘Procedural Modeling for Digital Cultural Heritage’ (Haegler et al. 2009) the authors examine the application of procedural modelling in archaeology. They argue that ‘the efficiency and compactness of procedural modelling make it a tool to produce multiple models, which together sample the space of possibilities.’ The core of their argument is what they refer to as ‘The Problem of Reconstruction Uncertainty’. This is the notion that detailed or realistic visualizations of archaeological research have the potential to falsely lead the viewer to take the ‘correctness of every detail for granted.’ This can be misleading, they argue, because a reconstruction is an educated guess among several other hypotheses. They further argue that procedural modelling addresses this concern because the variation between different models expresses levels of uncertainty implicitly and they go on to discuss examples of procedural modelling in archaeology, some of which are mentioned in this section, and instances where the notion of ‘reconstruction uncertainty’ is addressed successfully using procedural modelling.

Procedural modelling has been used to efficiently create a 3D reconstruction of an archaeological site in Mexico (Mueller et al. 2006). This implementation was based on the Computer Generated Architecture shape grammar, or CGA, which is a programming language specified to generate architectural 3D content used in the software *Esri CitiEngine*. Using this shape grammar, a rule set was created which could be used to create 3D models of Puuc-style architecture with a minimal effort. The authors summarize their approach in contrast to traditional 3D modelling: ‘Traditional 3D modeling tools often require too much manual work and their application is therefore overly expensive for archaeological projects. In contrast, our procedural modelling approach allows for the testing of several hypotheses by adjusting some of the parameters’ (Mueller et al. 2006, 1). The procedural model presented in their paper was created in 3 days. According to the authors, each of the buildings could be created within minutes, using this procedural model (Mueller et al. 2006, 6).

In another seminal study, Chun-Yen Huang and Wen-Kai Tai (2012), present a procedural approach for modelling a detailed Chinese ting, or pavilion. Huang and Tai propose that the use of procedural modelling and a user-friendly Graphic User Interface, or GUI, provide non-professionals with an intuitive means of constructing variants of complex Chinese tings within minutes. They provide evidence for this by way of a user study. They invited twelve users, two 3D artists and ten novice users, to use their modelling tool to accomplish three tasks: (1) model an existing ting from a reference photo; (2) model a ting to match a reference model;

and (3) create a ting with innovation. The researchers documented the time each user spent on each task and how many polygons made up the resulting 3D model. The results of this user study supported their hypothesis that non-professionals could effectively create 3D models using their procedural tools. The average time spent on the three tasks by the twelve users was 9.6 min (576 s), 10.3 min (618 s), and 7.2 min (432 s) respectively. On average, the users rated their experience using the procedural modelling tool to complete the three tasks as a 7.5 out of 10. Huang and Tai's paper provides evidence that a well-designed procedural model can provide researchers with limited 3D modelling experience an efficient means of constructing detailed 3D models (Huang and Tai 2012, 1303).

Another interesting approach, by Marie Saldaña (2015), demonstrates the use of procedural modelling to construct an entire city from GIS data and procedural rules. The project utilizes geographic data and maps to create the terrain. *CityEngine* software was used to describe and generate the different Roman building types and city rules. Once the scenes were generated by *CityEngine*, they were made viewable within the *Unity* game engine. By implementing a procedural approach, the researchers were able to build a comprehensive model of the city of Augustan Rome. The limitations of this approach discussed by Saldaña suggest that the CGA shape grammar does not have vocabulary to describe curved or radial geometry: 'My rules for a theater or stadium, for example, would seem to have been a simple exercise in symmetrical, radial geometry. The procedural grammar, however, was not well-equipped to describe such geometry, which made the writing of this rule a rather tortuous and long-winded process' (Saldaña 2015, 6).

8.6 Methodology

The methodology of this study was to develop a procedural model of the lower hull timbers of a sixteenth-century European merchant ship to demonstrate the effectiveness of a procedural modelling approach to ship reconstruction in nautical archaeology. Our approach was to construct each timber parametrically and maintain a procedural relationship between any given timber and the rest of the ship. Each component can be updated, in real-time, as revisions are made to interdependent components. The intent was to develop an approach which could construct a model with each timber component adjusted automatically; dramatically reducing the iteration time currently required using traditional modelling techniques. The development process for this model was a cycle of prototyping using *Houdini* modelling software.

The goal of this project was to test whether a procedural approach could be used to create a computer-based 3D model of the lower hull of any sixteenth-century European mer-

chant ship, based on a known recipe. Fernando Oliveira's instructions were selected to build an Indiaman, dating to circa 1580. To accomplish this project, we attempted to define: (1) a taxonomy describing each of the ship components considered, and its relationship to the other components; (2) a procedural model for each ship component based on a taxonomy developed from a series of sixteenth- and seventeenth-century technical documents related to shipbuilding; (3) a set of relations connecting the component parts, which generated a procedural model of the main components of a ship's lower hull; and (4) a set of rules to allow an assessment of the usefulness and effectiveness of the models created.

This project used *Side Effects Houdini* Software, a node-based procedural 3D package. The parameter interface for each component was constructed by leveraging Houdini's 'digital asset' file format. This facilitated the design and construction of parametric graphical user interfaces (GUIs). *Houdini* organizes the parts of a model into networks of nodes. Each node defines a part of the parametric data flow, which in turn defines each component. A digital asset is a way of encapsulating a network of nodes, which can then be interacted with, at a high level. Once the network is created and encapsulated within a digital asset, the user interface can be constructed by referencing node parameters inside the asset. And once created, the digital asset file can be loaded into any *Houdini* scene file. The asset can be placed inside of the scene using the software TAB menu or tool palette. We have created a collection of *Houdini* digital assets (HDAs), which can be installed into a scene file and then be used to efficiently create 3D models of a hull.

An important step in the implementation of this methodology was to create Graphical User Interfaces (GUIs), for each of the procedural timbers of the *nau* model. This is basically a library of timbers, which can be changed according to the taste of the user, varying its length, sided and moulded dimensions, scarves, curvature, etc.

The power of this methodology is that the GUI of each component, or HDA, allows the user to set parameters that affect the modelling procedure of each component. Each HDA has its own unique GUI, which is designed to present the user with parametric control of each of the procedural variables in a compartmentalized fashion. For instance, when an HDA encompasses multiple timber components, such as the keel, deck, and transom, each component's respective parameters are organized into labelled tabs. Some HDAs, such as the keel HDA, have context sensitive parameters, which are only activated and displayed when other parameters have certain values. In the case of the keel HDA, depending on the value of the *Rabet Type* parameter, the HDA interface will update with parameters specific to the selected rabbit type (Fig. 8.4).

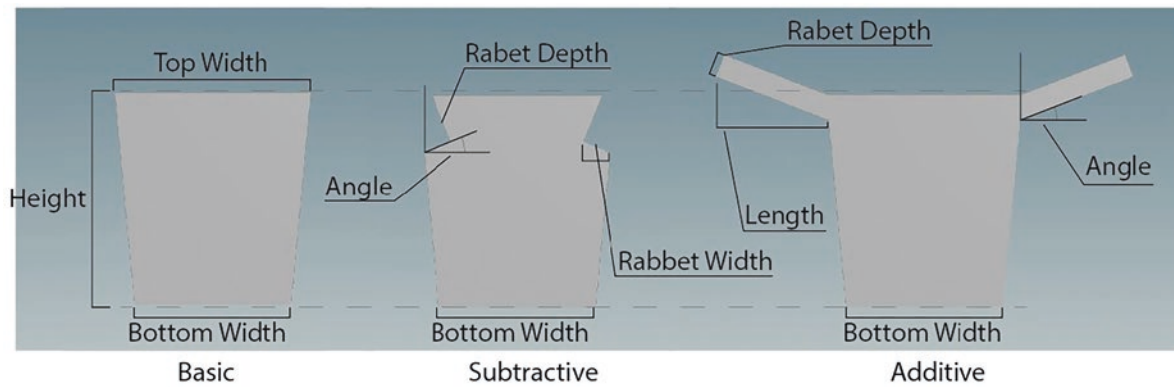


Fig. 8.4 Possible rabbet configurations built in the program

Dynamically populated parameters is another interface mechanism which we implemented in the planking GUI. This GUI proved to be too complex and diverse to be tackled in the preliminary study presented here, but its development is possible and relatively easy, only requiring extensive time to define and test the required parameters and the relations between them. Dynamically populated refers to the fact that the parameters are created and linked to their corresponding variables via a script, which is run to initialize the HDA. For example, upon initialization the planking HDA will automatically create sets of parameters for each frame used. The planking HDA interface also has clear and populate buttons to force updates when the number of input frames is changed.

The GUI for each HDA was designed with the intention of ease of use. The intention behind compartmentalizing parameters is to reduce visual clutter and avoid overwhelming the user with large quantities of parameters on screen at once. Context sensitive parameters also provide a way to limit the number of parameters displayed, only displaying parameters which are relevant to the current state of the model. Dynamically populated parameters provided the ability to design open ended interfaces, which adapt to user needs as the model changes and more parameters are needed.

8.7 Approach

The first step was to set the scope for the project by deciding on a basic collection of main timber components that go into the construction of a ship's hull. This scope defines the breadth of the timber taxonomy. The components considered were divided into three groups: longitudinal timbers, transversal timbers, and planking. Within the longitudinal timbers we have established the following sub-groups:

- Main ship spine—keel, stem, sternpost, and keelson;
- Longitudinal reinforcements—wales, stringers and breast hooks;

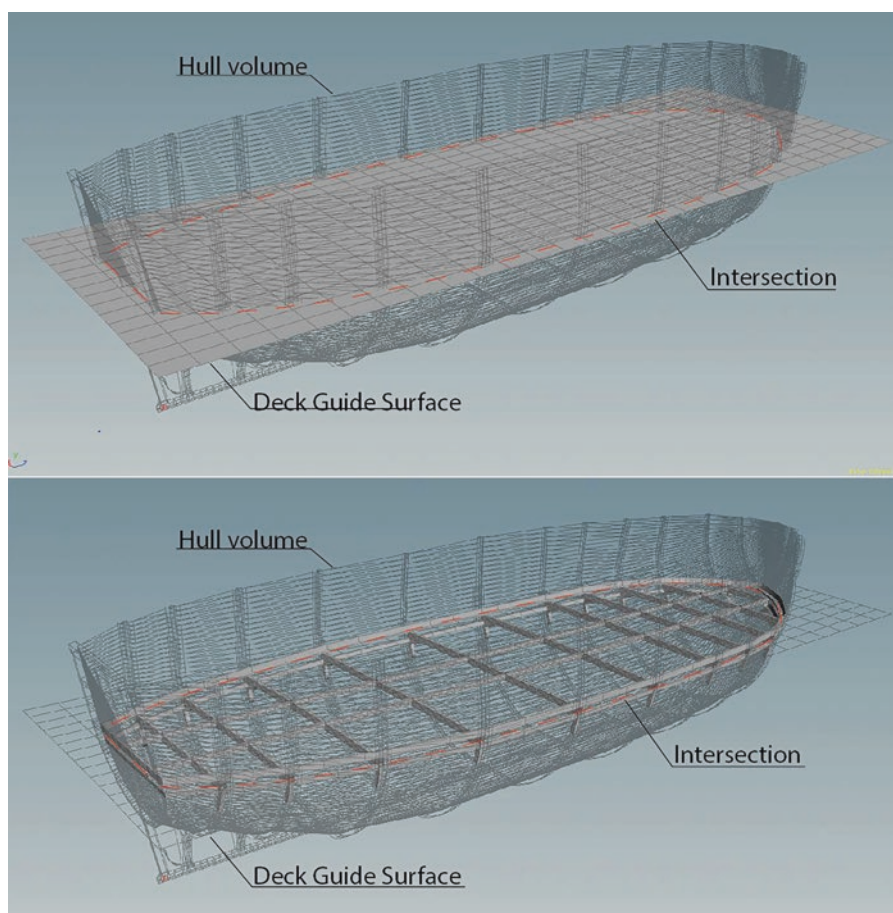
- Within the transversal timbers I have established the following sub-groups:
 - Frames—floor timbers, futtocks;
 - Stern panel timbers—fashion pieces, transoms; and
 - Deck timbers—deck beams, knees, clamps, waterways, and carlings.

The deck timbers group contains both longitudinal and transversal timbers, and these were grouped together by their common purpose, the construction of a diaphragm in the shape of a deck (Fig. 8.5).

We established a grammar of spatial relations between these timbers. Functional rules were also created for each timber, based on real-world ship design and construction processes. These rules were incorporated into the logic of the procedural algorithm. We built the ability of varying specific dimensions of each component into this model, while adhering to the rules of the grammar. It often took several attempts to determine a way to implement procedures which mimicked traditional ship construction processes. For every component there was careful attention to achieving a balance between customizability and automation. Automation was reserved for enforcing rules within the grammar and parametric control, and was provided for variables which were traditionally 'eye-balled' by the ship designer.

To define the basic shape of the main components of our model, we used a recipe proposed by Fernando Oliveira for the construction of a merchantman in 1580. Oliveira was a Portuguese priest and intellectual who wrote a shipbuilding treatise describing the main dimensions and shape of a Portuguese *nau* (Oliveira 1991). His text and drawings provided us with a departing model of a generic hull, which served as a base upon which to reason, test, develop new hypothesis, and try to understand the Belinho 1 timbers. The first step was to adopt a system of coordinates that defined the space where our base model was developed: Z-forward, Y-up, right handed coordinate system. This means the keel is created along the positive Z axis. In this coordinate system,

Fig. 8.5 A deck structure, encompassing all the timbers that have a natural symbiotic relation, such as clamps, beams, waterways, deck knees, and carlings



longitudinal reinforcements run along the Z axis and transversal timbers lie in defined XY planes.

8.8 Hull Components Description

The ship's longitudinal axis is composed of three main parts: the sternpost, the keel, and the stem. We divided the keel into five sub-groups of parameters; length, cross-section, sternpost, stern knee, and skeg. The most important parameter, the keel's length, is the length in metres of the horizontal portion of the keel, from sternpost to stem. The cross-section is a 2D shape which is swept along the keel and stem. For this model we provided three cross-section options: basic—a trapezoidal shape with variable top and bottom widths; subtractive—the same as the basic trapezoidal shape but with the subtraction of triangular or rectangular grooves known as *rabbets* (with a variable length, depth, and angles); and additive—with protruding rabbets with variable length, thickness, and angle (Fig. 8.4).

The sternpost is described using three measurements; length, depth, and angle. The stern knee is a curved support timber at the base of the sternpost. It is described by three measurements: height, length, and thickness. The skeg, the

protruding timber at the base of the sternpost whose function was to protect the ship's rudder in the event of beaching or hitting a reef, is described using three measurements; back edge Y, skeg height, and length.

The first step in the construction of the keel was defining the shape of its cross-section by creating a trapezoid based on the *Top Width*, *Bottom Width*, and *Height* parameters. The *Rabbit Type* parameter allowed the creation of several geometries for this feature. Next, the main horizontal portion of the keel, extending from the base of the sternpost to the base of the stem, was created by sweeping the cross-section shape along a line of N length, defined by the *Keel Length* parameter. The stem was created according to Oliveira's treatise, using a simple arc. In this case the keel's cross-section was swept along this curve to create the stem.

The stern knee was created using a rectangular polygon placed where the sternpost and keel meet. This polygon has the same width as the sternpost and its bottom edge was placed at the crease where the sternpost meets the keel. The height of the polygon was determined by a *Height* parameter. The polygon can rotate its base edge to the same angle as the sternpost. The bottom edge of the polygon was therefore extruded in the positive Z direction according to the *Length* parameter. The 'L' shaped geometry created was then

extruded inward according to the *thickness* parameter. The inside edge was bevelled to create a smooth curve. The skeg was created by extruding the keel's cross-section shape, without any rabbets, in the negative Z direction by its *Length* parameter. The *skeg height* parameter defined how much taller than the keel cross-section height the skeg was supposed to be. The *Back Edge Y* parameter controls the height of the top corner of the skeg. This can be used to create a tapering skeg as seen on some ships.

All the subcomponents described above were assembled together to create the final ship axial structure (Fig. 8.6). *Houdini* allows assignment of arbitrary data as attributes to any geometry and we utilized this feature to pass data from one HDA, or timber, to another. Keel length, cross-section height, cross-section width, and stern angle measurement are assigned as attributes to the keel geometry so that they can be read by other HDAs.

The resulting model of the keel, or HDA, can be created inside of a *Houdini* geometry node. This placed a keel with default parameters into the *Houdini* scene file, displaying in the screen viewport. Adjustments to parameters of the keel HDA will have an immediate effect on the model shown in the screen viewport. By adjusting parameters, a user can quickly create a completely different custom keel model, to be used to create an entire ship hull.

The frames were created based on the keel, a feature that will be automatically updated by any changes made to the keel HDA. To describe Oliveira's frames we used three basic shapes; a horizontal line for the flat bottom, a semi-circular futtock arc, and a straight line tangent to the end of the arc to define the top timbers. By describing these shapes, their relationship to each other, and their relationship to the keel, we developed rules which guided the procedural model.

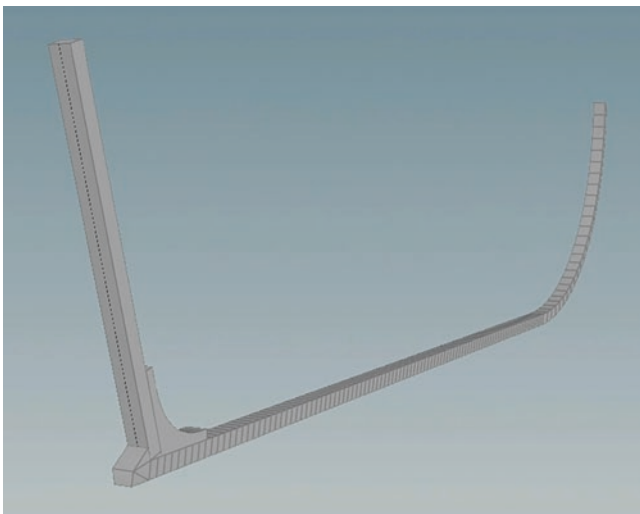


Fig. 8.6 Screen capture showing the model's keel and posts

To create the base curve which will act as a guide for the frame, a horizontal line is first created to represent the flat. The flat of the master frame is the widest of all the frames. For the rest of the frames, the length of the flat can be determined by the amount of its narrowing, and its height from the amount of the rising (Castro 2007; Suarez 2016). The centre of the futtock arcs was calculated geometrically and built into the model. Raising the flat of the frame further above the keel created the Y-frames, which were also defined with parameters and rules. To complete the frame's parameterization, we needed to describe each frame's position along the keel. Oliveira gives us clues to position all frames (Cook 2012, 50). Oliveira's treatise also says that the mast step is placed at half the keel's length. We have calculated the position of the master frame and placed the rest of the frames at similar distances.

Using the parameter values and the rules from the taxonomy developed for the project, we then developed spline curves to control the hull longitudinal shape and act as construction guides for the frame geometry. In the definition of the frames we have considered several ways to create futtocks. This option allowed models to be double-framed, as most ships were after the late seventeenth century, or to have frames composed of floor timbers and futtocks, as they were built before that. Therefore, if the *Futtock* parameter is unchecked, the frame geometry is created by sweeping a rectangular cross section along the guide curve. The dimensions of the rectangular cross section are defined by the parameters. If the futtocks option is turned on, four additional parameters are needed; *First Futtock Height*, *Second Futtock Height*, *First Futtock Overlap*, and *Second Futtock Overlap*. The resulting frame model, or HDA, can be created inside of a *Houdini* geometry node. To create a frame, a keel HDA must be used as input. By connecting the keel's output to the frame HDA's input, the keel passes along its attributes and the frame will be created according to its relationship to the keel as described above. A frame can be set to be *Master Frame* or *Custom Frame*. Setting the frame to master frame will automatically create a master frame based on the Oliveira recipe. Frames set to custom will have a procedural relationship with both the keel and the master frame. Adjustments to the parameters of the frame HDA will have an immediate effect on the model shown in the viewport. By adjusting parameters, a user can quickly create any frame in the hull.

The stern panel was defined as a sub-group of timbers within the transversal group, consisting of fashion pieces and transoms. Again, we used Oliveira's recipe and modelled the stern panel as a unit (Suarez 2016). The moulded dimension of the top transom and lower transoms can be defined by two parameters, *Top Transom Sided* and *Lower Transoms Sided*. Again, the resulting model of the stern panel, or HDA, can be created inside of a *Houdini* geometry node.

'Longitudinal reinforcements' was a group of timbers encompassing the whales, stringers and breast hooks. The timbers in this group were determined by the shape of the frames. The whales follow the exterior of the frames and the stringers the interior, and the breast hook follows the interior of the frames at the bow of the ship. To construct the stringers and the whales, there must first be a series of frames which define the hull shape. By using the frames as a guide, two hull shapes can be determined; one by the inside surface and the second by the outside surface of the frames.

'Deck structures' comprised the group of timbers containing both transverse and longitudinal timbers: beams, knees, carlings, clamps and waterways. For this model a deck is imagined as the intersection of a horizontal surface, which curves in three dimensions, and the volume of the hull defined by the frames. The surface has variable longitudinal and transversal curvature to create the cambered shape of a deck surface. The level defined by the surface is the top of the beams, above which the deck planks would be laid. By referencing the intersection surface which represents the deck surface, we could determine the location of all of the timbers in the deck structures group based on their dimensions, relationship to the deck surface, and their relationships to one another (Fig. 8.7).

The placement of the beams in this model is determined by two factors; the height of the top of the beam and its position along the length of the hull. The height of the top of a beam is defined by the deck intersection surface. A beam is connected at either end to a frame. The longitudinal position of a beam is determined by the location of the frames along the keel. The width of each beam can be determined by the distance, at the height of the deck, between each arm of the frame. The bottom surface of a beam can be determined by

offsetting from the deck intersection surface by the amount of the beam's moulded dimension. In this model, beneath each beam, connecting the beam to the frame, there is a deck knee. In other models these knees could be placed every other frame or even every three or four frames.

The resulting model of the deck, or HDA, can be created inside of a *Houdini* geometry node. Creating a deck needs three inputs; a keel, stern panel, and a group of frames. The output of the keel, stern panel, and merged frames is input into the deck HDA. This provides the deck HDA with all the necessary information to create a deck. The *Deck Height* parameter defines the deck location in the Y dimension. The *Lateral Bend* and *Longitudinal Bend* parameters control the camber for the deck. Each of the six timbers which make up the deck structure; beams, knees, carlings, clamps, and waterways, which in this model are composed of two timbers, in the Portuguese way, can be turned on or off independently. Each timber's width and sided dimension can be adjusted and all other timbers will update accordingly. Any changes to the keel, frames, or stern structure will update the deck timbers.

The stanchions span from the bottom-most stringer to the bottom deck beams, and then between the beams of each sequential deck. In this model, the stanchions are positioned longitudinally at each frame. The decks HDA also stores the locations at which the carlings meet the deck beams. Given these locations as input into the stanchions HDA, a spline through each set of corresponding points is created. To create the stanchion models, a rectangular cross section is swept along their splines. And once again, the resulting model of the stanchions, or HDA, can be created inside of a *Houdini* geometry node.

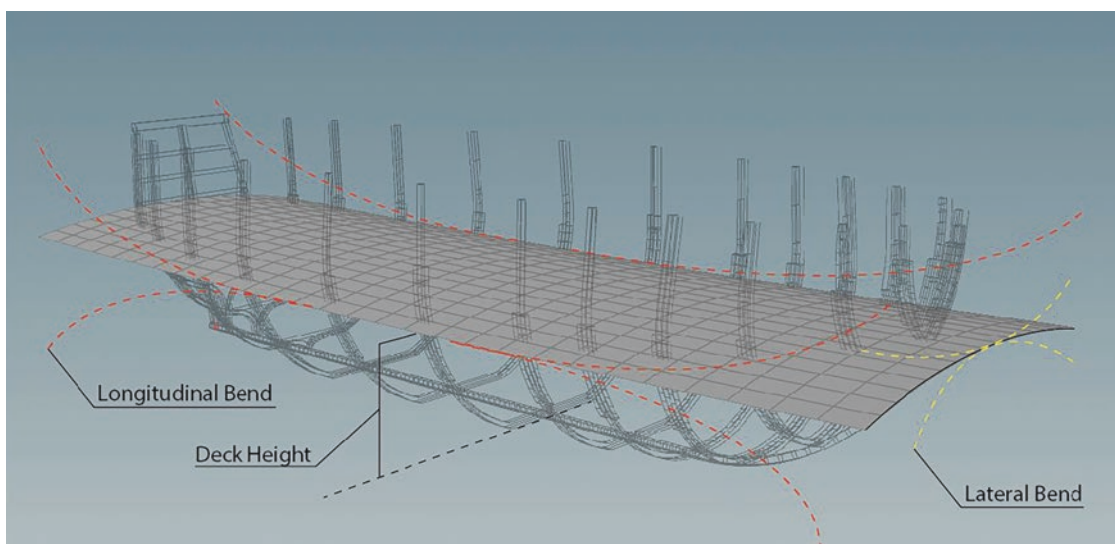


Fig. 8.7 Deck design

A keelson was also considered in this model, with a maststep as an enlarged portion of its section. The keelson sits atop the keel, sandwiching the frames along the flat. It typically starts at the point where the frames change from ‘Y-shaped’ to ‘V-shaped’ at the stern, ends forward at the apron, and is notched at each frame so that it locks in over them. The keelson has a rectangular cross section and follows along the lowest point of the frames. The keelson model was also created inside of a *Houdini* geometry node. The keelson is automatically placed along the top of the frames. The start and end locations, relative to the keel length are adjustable via the *Start Distance* and *End Distance* parameters. The *Notch Depth* parameter will adjust the depth of the notches in the keelson, lowering it over the frames by the notched amount. The *Maststep Start*, *Maststep End*, *Width*, and *Mortise Dimensions* allow the user full control over the placement and shape of the maststep. Any changes made to the keel or frames will automatically update the keelson HDA.

The planking of a hull is the application of the exterior layer of timbers which form the skin of the ship. The long strips of wood, called strakes, are each custom formed and cut to create the correct flow along the length of the hull. The designing of their shape is largely left up to the eye of the shipwright (Antscherl 2016, 3). The orientation of each strake along its length is determined by the curvature of the frame at that particular point along the hull’s length. The plank varies in its moulded dimension along its length to compensate for the hull’s 3D curvature. Each strake of the planking can be described by two variables for each frame;

the location of the bottom of the strake along the curve of each frame and the moulded dimension of the strake at each frame. The location of the bottom of the strake along the frame, in most cases, can be derived from the strake beneath it since each strake is stacked on the one lying below it (Fig. 8.8).

The planking was the most complex part of this project and missing from this model are the inevitable drop strakes, fillers, and repairs because it proved to be too time consuming and was not essential to the purpose of this experiment. The question of the bevels was similarly not addressed from a theoretical viewpoint. The resulting planking was considered adequate to the objectives of this experiment and is comprised of multiple strake HDAs, which can be created inside of a *Houdini* geometry node. A planking strake needs three inputs; a keel HDA, a stern panel, and a merged group of frames.

The fourth input is optional; it can be used to automatically set the position of the bottom of a strake directly on top of the strake beneath it. This will provide the strake HDA with all the necessary information to create one strake around the volume of the hull. For each frame, the strake HDA will automatically create the *Bottom of Strake #* and *Moulded #* parameters, where the ‘#’ represents the frame number. The *Bottom of Strake #* parameter will slide the current strake along the outside surface of the frame at each respective frame. The *Moulded #* parameter will adjust the strake’s moulded dimension at each particular frame. By adjusting these two parameters for each frame, each strake can be designed with a custom shape, to create the desired flow of

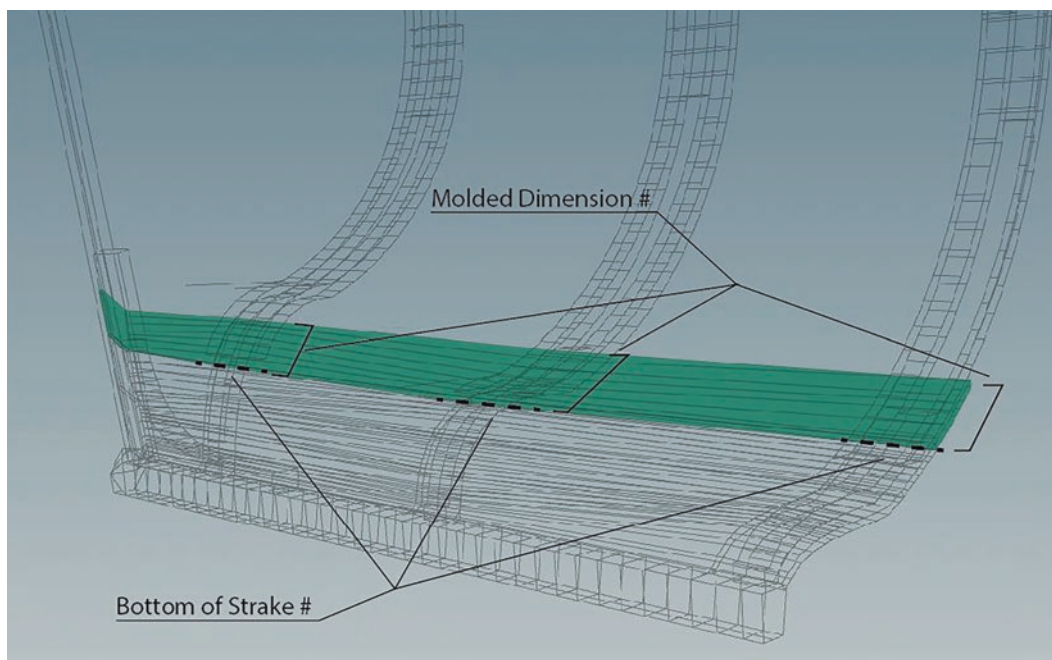


Fig. 8.8 Planking parameters

the hull planking, trying to emulate the methods employed by a shipwright.

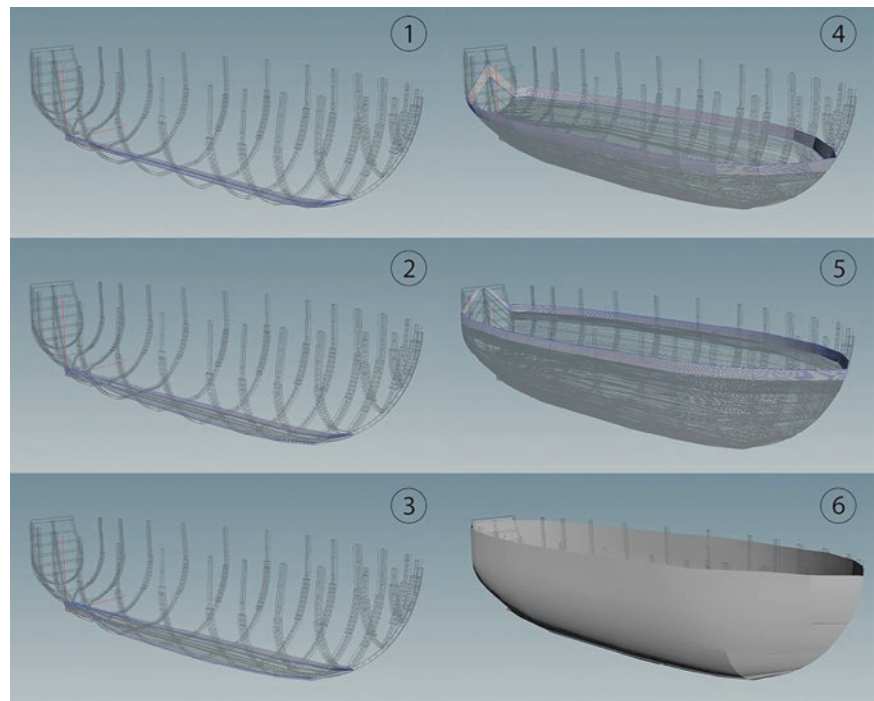
The drop strake concept was emulated in the following way: if the *Moulded #* parameter is set to zero for any frame, the strake will not be created at this frame. This feature allows the user to terminate strakes anywhere along the length of the hull, as is often necessary when ships are planked. By using a strake HDA as input the fourth input of another strake HDA, the *Bottom of Strake #* parameter will automatically update to the location of the top of the input strake for each frame.

The ideal workflow for planking the hull is to first create the bottom-most strake (garboard), adjusting the parameters until the desired shape of the first strake is determined. Then, copy and paste the first strake node to create a duplicate of the first strake and use the first strake as input into the new strake. The second strake will maintain the same moulded value but will be automatically placed along top of the first strake. The moulded dimension of the second strake can be adjusted to create a custom shape for the new strake. The new strake can also be extended to frames which the previous strake did not span, by using a non-zero *Moulded #* parameter value at those frames. This process of duplicating and stacking strakes continues up the side of the hull until the entire hull is planked (Fig. 8.9). Any changes made to timbers which affect the hull's shape will automatically update the strake HDAs. If adjustments are made to lower strakes, the strakes above it will automatically adjust to the new shape due to their defined relationship with the strakes below.

8.9 Conclusions and Future Work

This project and its methods proved to be an excellent exploratory reflection on the use of procedural methods to model ship hulls. The results, which were tested by entering the scantlings of a set of timbers from the Belinho 1 shipwreck (Castro et al. 2015), were promising and helped understand the complexity and variability of possible construction solutions in the history of early modern European merchant shipbuilding. *Houdini* proved to be versatile in ways unthinkable with other software packages, such as Autodesk *Maya*, for instance (Yamafune et al. 2016). We have established several directions for future work. Firstly, we have developed a number of additional features and details of the timbers considered for this project that could be included in future procedural models, such as scarves, hatches, stairs, gun ports, or even small details such as fastening patterns. We have also looked at extending the model upwards, and try to model the ship's upper works. But the most interesting direction of research is the definition of more recipes for the construction of ships. We believe that the power of procedural modelling lies precisely on future reflections on the diversity and similarity of shipbuilding recipes. Secondly, this project did not include a user study, which would test the application and user experience of the procedural tools, similar to the research of Chun-Yen Huang and Wen-Kai Tai in *Ting Tools*. Thirdly, we want to look at the possibility of integrating the *Houdini* digital assets created for this project into a game engine, such as *Unity* or *Unreal*, using the *Houdini* engine

Fig. 8.9 Planking process



plugin. This could create a widely accessible shipbuilding utility tool that could be deployed using *Unity* or *Unreal* engine, which are both free to download and use. We believe that this study firmly established the potential of procedural models to explore research hypotheses and we intend to pursue this avenue of investigation.

Acknowledgements This study would not be possible without the support of the Andre Thomas Family Fellowship, and in fact stemmed from a conversation he initiated between the Texas A&M University Departments of Visualization and Anthropology.

References

- Antscherl D (2016) A primer on planking. http://www.admiraltymodels.com/Planking_primer.pdf. Accessed 15 Aug 2016
- Castro F (2003) The pepper wreck, an early 17th-century Portuguese Indiaman at the mouth of the Tagus River, Portugal. *Int J Naut Archaeol* 32(1):6–23. <https://doi.org/10.1111/j.1095-9270.2003.tb01428.x>
- Castro F (2005) Rigging the pepper wreck. Part I: masts and yards. *Int J Naut Archaeol* 34(1):112–124. <https://doi.org/10.1111/j.1095-9270.2005.00048.x>
- Castro F (2007) Rising and narrowing: 16th-century geometric algorithms used to design the bottom of ships in Portugal. *Int J Naut Archaeol* 36(1):148–154. <https://doi.org/10.1111/j.1095-9270.2006.00116.x>
- Castro F (2009) Rigging the pepper wreck. Part II: sails. *Int J Naut Archaeol* 38(1):105–115. <https://doi.org/10.1111/j.1095-9270.2005.00048.x>
- Castro F, Fonseca N, Wells A (2010) Outfitting the pepper wreck. *Hist Archaeol* 44(2):14–34
- Castro F, Almeida A, Bezant J, Biscaia F, Carmo A, Crespo A, Farias I, Gonçalves I, Groenendijk P, Magalhães I, Martins A, Monteiro A, Nayling N, Santos A, Trapaga K (2015) The Belinho 1 timber catalogue. https://www.academia.edu/18771630/Belinho_1_Shipwreck_Timber_Catalogue_2015. Accessed 23 Aug 2018
- Cook C (2012) A parametric model of the Portuguese nau. MA Thesis, Texas A&M University
- Doran J, Hodson F (1975) Mathematics and computers in archaeology. Harvard UP, Cambridge, MA
- Evans TL, Daly PW (2006) Digital archaeology: bridging method and theory. Routledge, London
- Fleming A, Clarke DL (1973) Models in archaeology. *Man* 8(2):315
- Haegler S, Mueller P, Gool L (2009) Procedural modeling for digital cultural heritage. *EURASIP J Image Video Process* 2009:852392. <https://doi.org/10.1155/2009/852392>
- Hazlett AD (2007) The nau of the livro nautico: reconstructing a sixteenth-century Indiaman from texts. PhD dissertation, Texas A&M University
- Hill RW (1994) A dynamic context recording and modeling system for archaeology. *Int J Naut Archaeol* 23(2):141–145. <https://doi.org/10.1111/j.1095-9270.1994.tb00453.x>
- Huang C-Y, Tai W-K (2012) Ting tools: interactive and procedural modeling of Chinese ting. *Vis Comput* 29(12):1303–1318. <https://doi.org/10.1007/s00371-012-0771-3>
- Lock G (2003) Using computers in archaeology: towards virtual pasts. Routledge, London
- Miller P, Richards J (1995) The good, the bad, and the downright misleading: archaeological adoption of computer visualization. In: Huggett J, Ryan N (eds) CAA94. Computer applications and quantitative methods in archaeology 1994, BAR International Series 600. Tempus Reparatum, Oxford, pp 19–22
- Mueller P, Vereenoghe T, Wonka P, Paap I, Gool L (2006) Procedural 3D reconstruction of Puuc buildings in Xkipché. In: Ioannides M, Arnold D, Niccolucci F, Mania K (eds) The 7th international symposium on virtual reality, archaeology and cultural heritage (VAST 2006)
- Oliveira F (1991) O Livro da Fabrica das Naus. Academia de Marinha, Lisboa
- Richards J (1986) Computer usage in British archaeology, IFA occasional paper no. 1. Institute of Field Archaeologists, Birmingham
- Saldaña M (2015) An integrated approach to the procedural modeling of ancient cities and buildings. *Digit Scholar Human* 30(1):148–163. <https://doi.org/10.1093/llc/fqv013>
- Suarez M (2016) A procedural approach to computer-aided modeling in nautical archaeology. MA thesis, Texas A&M University
- Wells A (2008) Virtual reconstruction of a seventeenth-century Portuguese nau. MA thesis, Texas A&M University
- Yamafune K, Torres R, Castro F (2016) Multi-Image photogrammetry to record and reconstruct underwater shipwreck sites. *J Archaeol Method Theory* 24(3):703–725. <https://doi.org/10.1007/s10816-016-9283-1>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

