A PROCEDURAL APPROACH TO COMPUTER-AIDED MODELING IN

NAUTICAL ARCHAEOLOGY

A Thesis

by

MATTHEW RAY SUAREZ

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Frederic Parke
Co-Chair of Committee,	Tim McLaughlin
Committee Members,	Felipe Castro
Head of Department,	Tim McLaughlin

December 2016

Major Subject: Visualization

ABSTRACT

This research is to determine the functionality and applicability of procedural computer-based modeling techniques in the field of nautical archaeology. To demonstrate this approach, an interactive procedural model of the lower hull timbers of a 16th century European merchant ship was developed through an iterative process of prototype implementation. Evaluation of the usefulness and effectiveness of the prototypes based on their potential research applications was conducted by Dr. Filipe Castro of the Texas A&M Department of Anthropology.

The 3D model was created using Houdini, a procedural node-based 3D software package. First, a basic collection of main timber components that go into the construction of a ship's hull was determined. Functional rules were created for each timber based on real-world ship design and construction processes. These rules were incorporated into the logic of the procedural modeling algorithm. I built into this model the ability of varying specific dimensions of each component, while adhering to the rules of the algorithm. Each component was updated, in real-time, as revisions were made to interdependent components.

The resulting procedural approach created a flexible and interactive model which can be iterated with parametric control, thus reducing revision time. The results of this project provide evidence of the time-saving effectiveness of a procedural approach to creating 3D models as a research tool. This is useful because once procedural models are created, they provide an accessible means for researchers to create multiple interpretations. Based on my experience with this project, I believe that the flexibility provided by procedural modeling provides an overall time efficient solution for 3D

modeling in nautical archaeology. The construction of a procedural model requires a significant investment in design, construction, and trouble-shooting; however, this is outweighed by the flexibility it offers.

DEDICATION

For my loving family: Mario, Belinda, Samantha, and Clare.

ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. Frederic Parke, for his continued feedback and guidance throughout the entire process; I could not have completed this thesis without his consistent support and reliable feedback.

Also, I would like to thank Dr. Felipe Castro for his contagious enthusiasm, patient tutoring, and committed involvement throughout our research together. His passion for teaching and innovation was an inspiration throughout the project.

I also wish to thank Tim McLaughlin for his encouragement and guidance. Taking his class was one of my favorite experiences at Texas A&M; his engaging lectures, intertwined with intriguing tales of the industry, excited my interests and inspired my academic and professional career.

I would like to thank André Thomas for his instrumental role and inspiration for this interdisciplinary research project. I have had the honor and pleasure of working with André in multiple capacities; his infectious determination and ambition in all aspects of his life are a huge inspiration. I cannot thank him enough for his continued support and mentoring throughout my academic and professional career.

Thank you also to the Thomas family for their support of this research through the contribution of the Thomas Family Scholarship.

Further, I wish to thank my friends and colleagues, as well as faculty and staff, in the Viz and LIVE Lab; thank you all for making me feel so welcome during my stay at Texas A&M. Also, thanks to the Gen Art group for the comic relief between all the work.

Finally, a huge thank you to my family and fiancé, I could have never done any of this without your unwavering love and support.

TABLE OF CONTENTS

Page

LIST OF FIGURES

I. INTRODUCTION

The goal of this project was to reduce the large investment of time and expertise that is currently required, using typical modeling methods, to create 3D reconstruction models for nautical archaeological research. The solution is an approach which leverages computer based parametric and rule-based modeling. To demonstrate this, a procedural model of the lower hull of a 16th century European merchant ship was developed through a cyclical process of prototype implementation. Evaluation of the usefulness and effectiveness of the prototypes, based on their potential research applications, was provided by Dr. Filipe Castro of the Texas A&M Department of Anthropology. This project is intended to provide evidence of an efficient and intuitive means of building a 3D computer-based model for nautical archaeological research.

1.1. COMPUTER-AIDED MODELING IN ARCHAEOLOGY

Computer-aided modeling offers archaeological researchers a perspective through which they can analyze archaeological data. Such models allow visualization of the data collected from a shipwreck. Models offer researchers an opportunity to view what is left of the remains as it might have been before the wreck. Since there are generally only partial remains, clues to identification of a particular shipwreck typically come from the cargo, the construction features, and the materials employed. In any given time period, ship shapes and hull structures vary from region to region. This variation means that a ship's 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 a highly iterative and interpretive process, continually evolving as evidence is uncovered. To assist in visualizing the data collected, researchers will often create models; hand-drawn, physical construction, 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 a computer based 3D model. Model precision is maintained by the computer, therefore models are subject to fewer opportunities for human error and fatigue. Scalability of 3D computer models allows a model to be constructed at full scale. 3D modeling also allows for the automation of redundant tasks. Moreover, computer processing power allows larger quantities of data to be included in a model. One drawback however of typical 3D modeling techniques is the large overhead of expertise and knowledge in 3D modeling and 3D modeling software required to create a 3D model. A second drawback to typical computer-based modeling methods 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 iterative process that is ship reconstruction and can deter researchers from implementing 3D models as research tools in their work.

II. BACKGROUND

2.1. COMPUTER-BASED MODELING IN ARCHAEOLOGY

During the 1960's and 1970's computers were mainly used in archaeological research for statistical applications such as classification and seriation, archaeological techniques which predate computers. The first textbook published on the topic of computers in archaeology, "Mathematics and Computers in Archaeology" by Doran and Hodson in 1975, focuses on the application of data classification and quantification in archaeological research (Doran and Hodson 5). Use of computers at the time was limited largely due to their cost and limited 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 cost and facilitated access. By the late 1970's 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 focused on atheoretical tasks (Richards 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 1). The rapid development of graphics, computer-based visualization, and computer software in the mid 1980's to the 1990's 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 211).

A model is defined as "a simplification of something more complex to enable understanding" (Lock 6). In 1974, in *Models in Archaeology*, Andrew Fleming and David Clarke described a model as "ideal representations of observations which are heuristic, visualizing, comparative, organizational, and explanatory devices" (Fleming and Clarke 316). With this definition, he describes that these qualities of a model 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 317).

Prior to the mid 1970's, an archaeological model was either a 2-dimensional 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 atheoretical, meaning that they were not intended to explore or inspire interpretations but rather to measure and document existing data (Evans, 11). According to this use, computers and computer models were not yet used to inspire or explore alternative ideas.

Another view is that digital developments create or influence the creation of theory, similar to the traditional use of 2D drawings or physical models (Haegler, Meuller, and Van Gool 1). One of the opposing arguments of this view is summarized by Miller and Richards in their 1994 essay “The good, the bad, and the downright misleading: archaeological adoption of computer visualization”;

> Each project was a result of 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 interposed between them and their data. The archaeologists did not have direct control of the modelling themselves (Miller and Richards 20).

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

2.2. PROCEDURAL MODELING

This section defines the term *procedural modeling* as it applies to this thesis. ‘Procedural modeling’ is a general term for techniques in computer graphics which create 3D models or textures from a set of rules. L-Systems, fractals, and generative modeling are all included in this family of techniques. For my work, the term procedural modeling refers to creating 3D models through rules which are configurable by parameters.

2.3. PROCEDURAL MODELING IN ARCHAEOLOGY

"Procedural Modeling for Digital Cultural Heritage" by Simon Haegler examines the application of procedural modeling in archaeology. He argues "the efficiency and compactness of procedural modeling make it a tool to produce multiple models, which together sample the space of possibilities." The core of his argument is what he refers 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, he argues, because a reconstruction is an educated guess among several other hypotheses. He argues that procedural modeling addresses this concern because the variation between different models express levels of uncertainty implicitly. Haegler goes on to discuss examples of procedural modeling in archaeology, some of which are mentioned in this section, and instances where the notion of 'reconstruction uncertainty' is addressed successfully using procedural modeling (Haegler, Mueller, and Van Gool 1).

In the paper "Procedural 3D Reconstruction of Puuc Buildings in Xkipche" by Pascal Muller et al., procedural modeling is used to efficiently create a 3D reconstruction of an archaeological site in Mexico. This implementation is 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 CitiEngine by Esri (Muller, Vereenooghe, Wonka, Paap, and Van Gool 1). Using this shape grammar, a rule set is created which can be used to create 3D models of Puuc-style architecture with minimal effort. In the following quote, the authors discuss their approach in contrast to traditional 3D modeling:

> Traditional 3D modeling tools often require too much manual work and their application is therefore overly expensive for archaeological projects. In contrast, our procedural modeling approach allows for the testing of several hypotheses by adjusting some of the parameters. (Muller, Vereenooghe, Wonka, Paap, and Van Gool 1)

The procedural model presented in their paper was created in three days. According to the authors, each of the buildings within Xkipche can be created within minutes using this procedural model (Muller, Vereenooghe, Wonka, Paap, and Van Gool 6).

"Ting Tools: Interactive and Procedural Modeling of Chinese Ting" by Chun-Yen Huang and Wen-Kai Tai, presents a procedural approach for modeling a detailed Chinese ting, or pavilion. Huang and Tai propose that the use of procedural modeling 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 modeling tool to accomplish three tasks. The three tasks were: (1) model an existing ting from a reference photo; (2) model a ting to match a reference model; (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 minutes (576 seconds), 10.3 minutes (618 seconds), and 7.2 minutes (618 seconds) respectively. On average, the users rated their experience using the procedural modeling 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 modeling experience an efficient means of constructing detailed 3D models (Huang and Tai 1303).

"An Integrated Approach to the Procedural Modeling of Ancient Cities and Buildings" by Marie Saldana, demonstrates the use of procedural modeling 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 in this paper 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. However, the procedural grammar was not well-equipped to describe such geometry, which made the writing of this rule a rather tortuous and long-winded process. (Saldana 6)

III. RELATED WORKS

This section is a review of precedents in computer aided modeling in Nautical Archaeology to demonstrate the need for more efficient ship reconstruction modeling tools.

3.1. ALEXANDER HAZLETT

In his 2007 dissertation, “The Nau of the *Livro Nautico*: Reconstructing a Sixteenth-century Indiaman from Texts”, Alexander Hazlett sought to synthesize the data from various sources on the subject of the Portuguese nau to create a timber by timber model. From the model, constructed in Rhinoceros 3D, Hazlett created annotated and illustrated construction diagrams of the nau. His process involved examining and deciphering ship building documents of the period, like theoretical treatises and sets of standard dimensions for parts of shipbuilding structures, also known as “scantling lists”, to determine the placement of each timber. Hazlett’s project serves as an example of typical, or ‘manual’, modeling techniques implemented in ship reconstruction. This refers to the fact that every point of the 3D model was created and positioned directly by the user using a GUI, or graphic user interface. His dissertation also provides evidence of the iterative and interpretive processes of ship reconstruction. These instances of iteration emphasize the need for tools which embrace the iterative and interpretive aspects of Nautical Archaeological research.

Hazlett describes in detail the modifications made during and after the modeling process:

> Later, I replaced approximately 20 percent of the forward end of the model (frames, beams, knees, clamps, stringers and all the other timbers above the keel) after I had modeled the ship up to the weather deck, in order to rebuild the bow with a rounder, more appropriate shape. Multiple variations of hull shape, frame patterns, hatches and castle shapes were modeled as different layers of the model, to 'try out' different configurations of timbers. Two stern sections (one without a tiller port or rudder, and one that incorporated both) were constructed. This sort of wholesale modification, particularly so late in the building process, would have been much harder with traditional pen-and-ink drafting. (Hazlett 186)

In this quote Hazlett speaks to the iterative and interpretive process that is involved in ship reconstruction. He mentions how certain aspects of the ship were constructed just to 'try out' different configurations. He also mentions how computer-based modeling software facilitated these revisions and speaks to the strengths of 3D models compared to 2D illustrations. Later he elaborates on this subject:

> Building the model has forced me to face not only the limits of our understanding of Portuguese shipbuilding nomenclature and methods but also the limits of modern graphically-based methods of design (in particular the limitations of the standard 3-view illustrations). It has also shown how useful computer modeling can be for projects like this (both for the ease of modification and for the ease of viewing the ship from any angle). (Hazlett 18)

Although computer modeling made the process easier than 2D illustration, directly manipulating each point of a highly complex object with interdependent parts, like a wood timber ship, is very time consuming. In a non-procedural workflow like the one implemented by Hazlett, revisions require a significant time investment, since this requires the adjustment of every subcomponent affected by the changes. Hazlett speaks to these difficulties in his conclusion:

> Modeling line by line from a 16th-century document has been at times painstaking, exciting, challenging, and frustrating in equal measure. The task was painstaking in the placement (and subsequent replacement or modification, in some cases more than once) of each frame and plank. (Hazlett 189)

In my work I addressed this shortcoming by developing a procedural workflow that enforces defined relationships between components. This facilitated an interactive revision process. As changes are made to one component others update accordingly. Procedural modeling embraces the iterative process, inviting the researcher to investigate alternative hypotheses.

3.2. AUDREY WELLS

In her thesis, “Virtual Reconstruction of a Seventeenth-Century Portuguese Nau”, Audrey Wells created a detailed reconstruction of the Portuguese vessel, *Nossa Senhora dos Martires*, lost in 1606, also known as the Pepper Wreck. This reconstruction could be viewed using a real-time immersive visualization system. To construct this model with archaeological efficacy in mind, Wells relied on the direction and evaluation of nautical archaeologist Dr. Filipe Castro. Wells proposes that the most successful applications of virtual archaeology are the result of collaborations between archaeologists and visualization specialists (Wells 15). My approach resembles the collaborative approach employed by Wells in the initial modeling phase as the procedural model is first constructed. Contrary to Wells’ approach, after the model is created, the procedural framework of the model empowers the researcher with an efficient means to test alternative hypothesis independent of a 3D artist.

The collaborative development cycle coupled with the iterative process that is required in ship reconstruction required diligent communication and frequent meetings between Wells and Dr. Castro; Wells states that meetings between the two were held as often as three times a week. Wells explains the 'iterative refinement' process employed in ship reconstruction and in particular how it was implemented in her project with a collaborative workflow:

> First, the initial model is evaluated. Second, any new data that is found or created should be integrated into the existing pool of data. Third, the model is refined by utilizing the new information. Then the model is re-evaluated and if it is now satisfactory, the model is done. If not, the model development loop iterates again. (Wells 39)

In her conclusion, Wells goes on to explain that due to the continual discovery process of archaeology, she feels that the work is never truly finished because the uncovering of new information could continue to inform the model:

> The interpretive development cycle discussed in Chapter I and the iterative project methodology discussed in Chapter IV make this project a constantly evolving one. I believe it will never be truly "finished" because archaeological research is always uncovering new information, which could be applied to the model. (Wells 86)

By developing a model which the archaeologist can use to test alternative hypothesis, independent of a visualization expert, the research application of the 3D model can be extended for as long as the research continues.

3.3. JUSTUS COOK

Justus Cook's research, "A Parametric Model of the Portuguese Nau", in which he created a parametric computer model of a nau hull, largely influenced my research.

Cook gathered information from 16th and 17th century treatises to create a scripted interface which allows the user to visualize multiple interpretations by inputting various hull parameters, such as beam measurements. In contrast to the modeling techniques implemented by Hazlett and Wells, Cook discusses the benefit of using parametric models in his research; “In general, 3D models represent only one specific example; however, a parametric model can be used to explore variations of a given model. By utilizing a parametric model in this project, multiple hull shapes were created and their cargo capacities calculated” (Cook 14). Similar to his approach, my procedural modeling approach is intended to extend the research applications of a model.

The diagrams and proportions contained in Cook’s thesis largely informed my research. To construct my model, I used the geometric proportions, which Cook derived from Fernando Oliveira’s *Liuro da Fabrica das Naus,* as a guide as well as the direction of Dr. Castro.

IV. METHODOLOGY

The methodology of this study is to develop a procedural model of the lower hull timbers of 16th century European merchant ship to demonstrate the effectiveness of a procedural modeling approach to ship reconstruction in Nautical Archaeology. My approach is to construct each timber parametrically and maintain a procedural relationship between each timber and the rest of the ship. Each component can be updated, in real-time, as revisions are made to interdependent components. The intent is to develop an approach which can construct a model with each timber component adjusted automatically, dramatically reducing the iteration time currently required using traditional modeling techniques.

The development process for this model was a cycle of prototyping using Houdini modeling software. Revisions were based on the analysis and direction of Dr. Castro.

V. GOALS AND OBJECTIVES

The goal of this research was to demonstrate a procedural approach to creating computer-based 3D models of the lower hull of any 16th century European merchant ship. To accomplish this:

1. A taxonomy describing each of the scoped ship components and its relationship to the other components was created.
2. A procedural model was then constructed for each ship component based on the taxonomy.
3. The component parts were connected together using the taxonomy to create a procedural model of the main components of a ship's lower hull.
4. The usefulness and effectiveness of the models were evaluated by Dr. Castro based on their potential research applications.

VI. IMPLEMENTATION AND APPROACH

6.1. SOFTWARE

For this project I chose to use Side Effects Software's Houdini, 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 GUIs. Houdini organizes the parts of a model into networks of nodes. Each node defines a part of the parametric data flow which 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. Once created, the digital asset file can be loaded into any Houdini scene file. The asset can then be placed inside of the scene using the TAB menu or tool palette. I have created a collection of Houdini digital assets, HDAs, which can be installed into a scene file and then used to efficiently create 3D models of a hull.

6.2. GRAPHICAL USER INTERFACE

An important step in my implementation was to create Graphical User Interfaces, or GUIs, for each of the procedural timbers of the *nau* model. The GUI of each component, or HDA, allow the user to set parameters to affect the modeling 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 labeled tabs (see fig. 1).

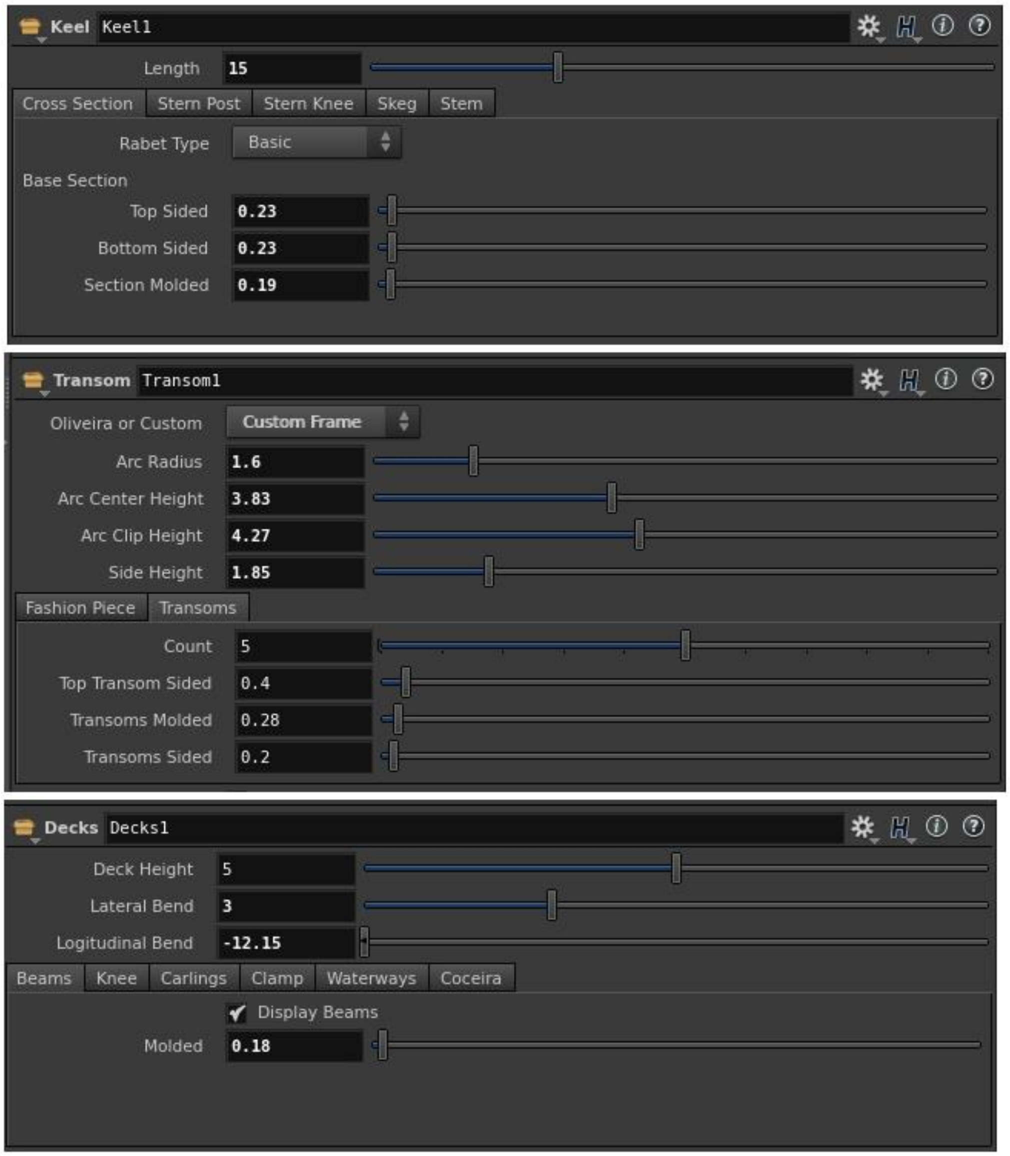

Figure 1. Keel, transom, and deck GUIs. Showing compartmentalized organization of sub-component parameters

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 rabet type (see fig. 2).

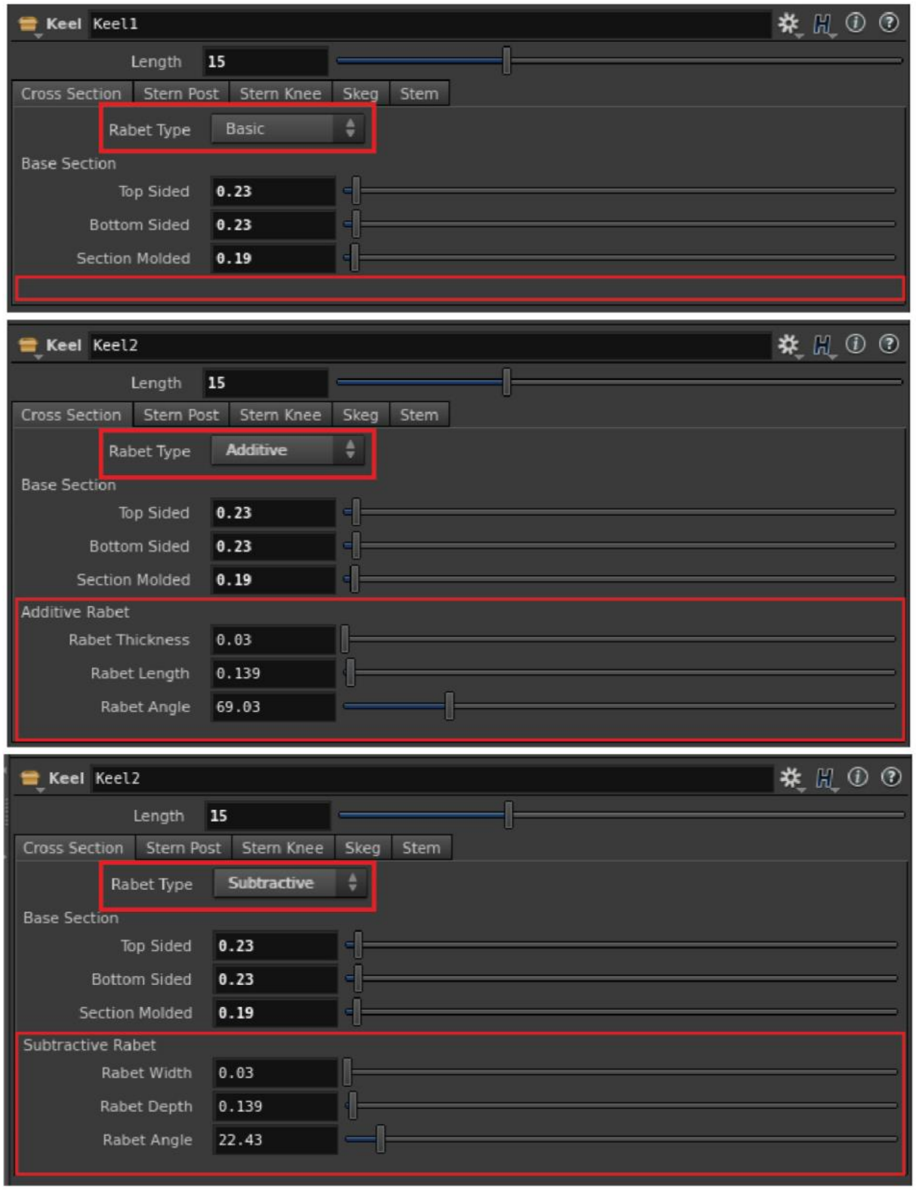

Figure 2. Keel GUI. Showing context sensitive parameters based on *Rabet Type* value

Dynamically populated parameters is another interface mechanism which I implemented in the planking GUI. 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 update when the number of input frames is changed (see fig. 3).

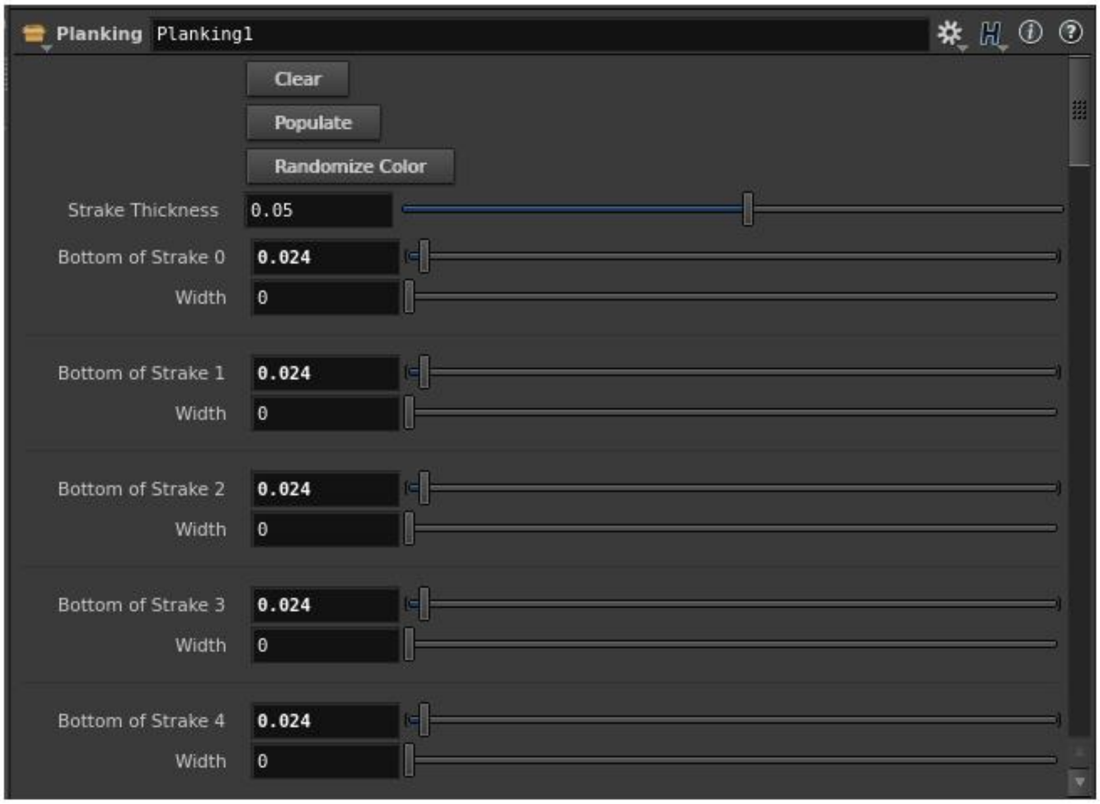

Figure 3. Planking GUI. Showing dynamically created parameters

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 provides 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, ones which adapt to the user's needs as the model changes and more parameters are needed.

6.3. 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 taxonomy. The scoped components are divided into three groups; longitudinal timbers, transversal timbers, and planking. Within the longitudinal timbers I have established the following sub-groups (each of the following terms will be defined later in this chapter):

1. Main ship spine – keel, stem, stern post, keelson
2. Longitudinal reinforcements – wales, stringers and breast hooks

Within the traversal timbers I have established the following sub-groups:

3. Frames – floor timbers, futtocks
4. Stern panel timbers – fashion pieces, transoms
5. Deck timbers – deck beams, knees, clamps, waterways, *coceira,* and carlings

The deck timbers group contains both longitudinal and transversal timbers, however these were grouped together by their common purpose, the construction of a deck. Based on Dr. Castro's expertise in the area, 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. I built into this model the ability of varying specific dimensions of each component, 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 was provided for variables which were traditionally “eye-balled” by the ship designer.

6.4. FERNANDO OLIVEIRA

Fernando Oliveira was a Portuguese priest and intellectual who wrote *Liuro da Fabrica das Naus* in 1850, a shipbuilding treatise about the Portuguese Nau. For the purpose of focusing the scope of this project, my procedural model was built using the standards and rules of thumb described in Oliveira’s treatise. This treatise was selected because of Dr. Castro’s extensive knowledge and research in the area.

6.5. COORDINATE SYSTEM

The coordinate system used for this project is a Z-forward, Y-up, right handed coordinate system (see fig. 4). This means the keel is created along the positive Z axis. In this coordinate system, longitudinal reinforcements run along the Z axis and transversal timbers lie in defined XY planes.

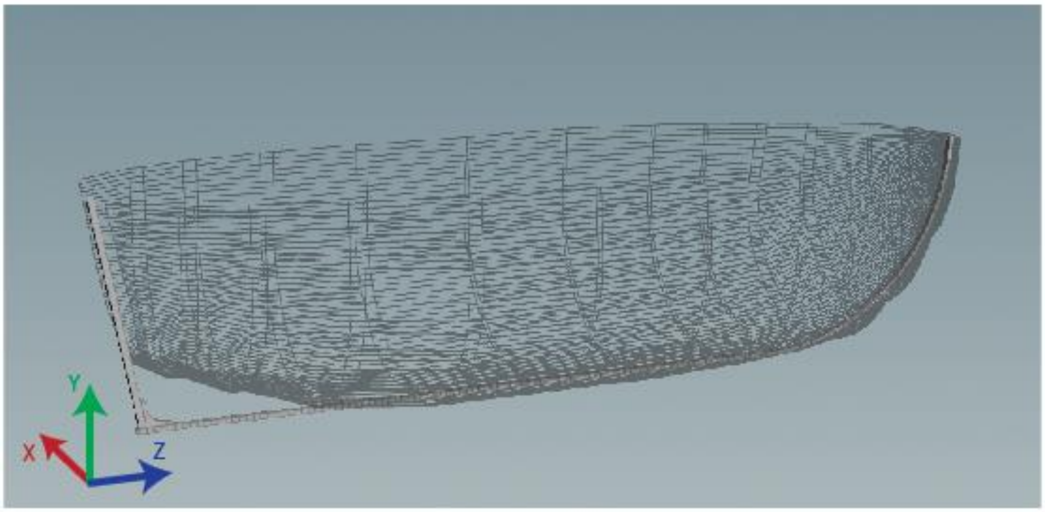

Figure 4. Y-up right handed coordinate system

6.6. KEEL

The keel is the long horizontal portion along which the frames are mounted. It can be constructed of several sections. Together with the stern post and stern panel, the keel makes up the ship's main longitudinal structure. The stern panel, which is an angled semi-vertical post, is composed of two curved timbers named fashion pieces and horizontal transverse timbers named transoms. The stem post is the curved portion of the keel, at the bow of the ship (see fig. 5).

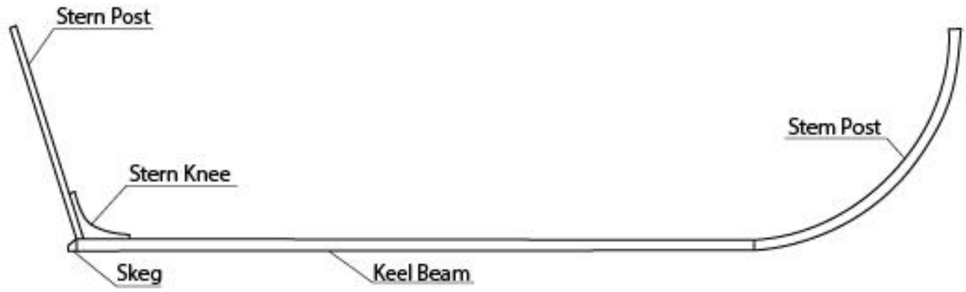

Figure 5. Keel sub-components

6.6.1. KEEL TAXONOMY

The ship's longitudinal axis is constructed of three main components; the stern post –an angled semi-vertical post where the fashion pieces and transoms are fixed, the keel – which is the long flat portion along which the frames are mounted, and the stem post – the curved portion of the keel at the bow of the ship (see fig. 5). We divided the keel into five sub-groups of parameters; the length, cross-section, stern post, stern knee, and skeg. The most important parameter, the keel's length, is the length in meters of the horizontal portion of the keel, from stern post to stem post.

The cross-section is a two dimensional shape which is swept along the keel and stem post. For this model I provide three options for cross-section types (see fig. 6). Basic –a trapezoidal shape with variable top and bottom widths. Subtractive –which is the same as the basic trapezoidal shape but with the subtraction of rectangular shapes known as *rabbets*, a notched channel along the length of the keel and stem post where the planking meets the keel. The subtractive rabbets have a variable length, depth, and angle. The third cross-section type is additive –instead of negative rabbets, additive has protruding rabbets with variable length, thickness, and angle.

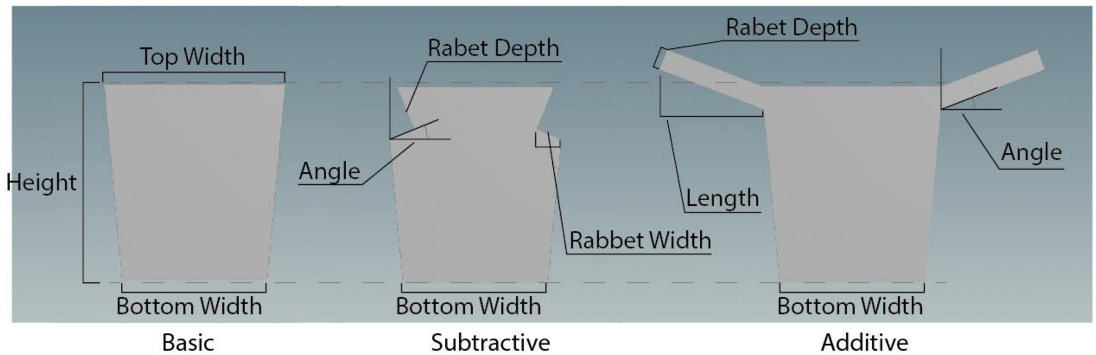

Figure 6. Cross-section dimensions

The stern post is described using three measurements; length, depth, and angle (see fig. 7). These three measurements are provided as parameters to the user. The stern post length parameter can use Oliveira's proportion of one-third the keel's length or can be set by the user.

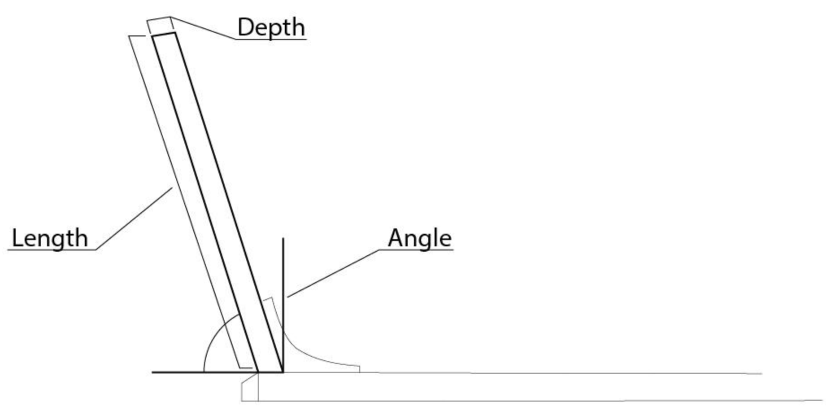

Figure 7. Stern post dimensions

The stern knee is a curved support timber at the base of the stern post. The stern knee is described by three measurements; height, length, and thickness (see fig. 8).

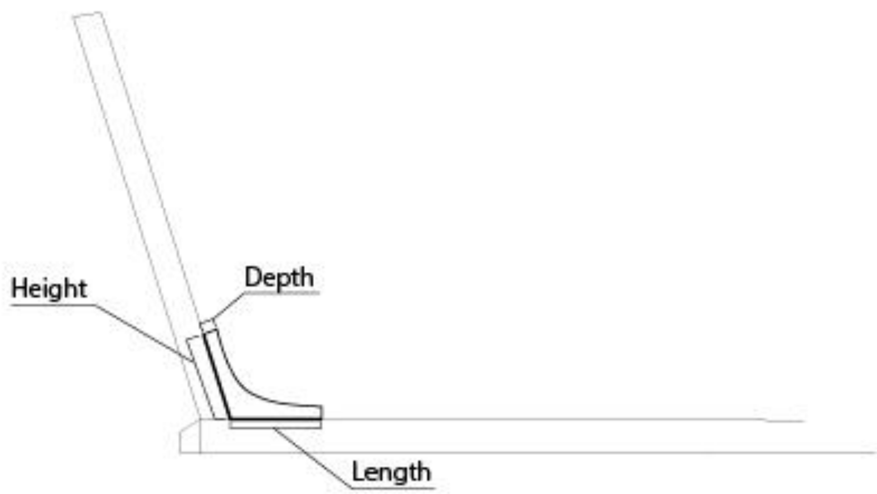

Figure 8. Stern knee dimensions

The skeg is the protruding bump at the base of the stern post whose function was to protect the ship's rudder in the event of beaching or hitting a reef (see fig. 5). For this model, the skeg is described using three measurements; back edge Y, skeg height, and length (see fig. 9).

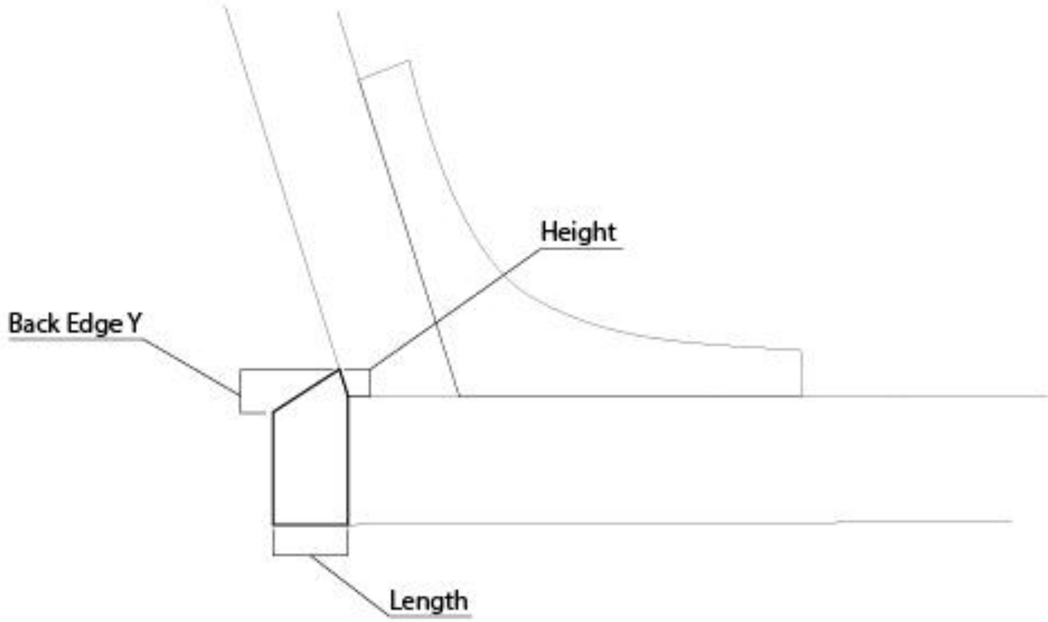

Figure 9. Skeg dimensions

6.6.2. KEEL CONSTRUCTION

The first step in the construction of the keel is defining the shape of its cross-section. The model defines a cross-section by first creating a trapezoid based on the *Top Width*, *Bottom Width*, and *Height* parameters (see fig. 6). Based on the *Rabbet Type* parameter, the model will either subtract from or add rectangular shapes to both sides of the trapezoidal shape. Additive rabbets are defined by three parameters; *Thickness, Length*, and *Angle.* Subtractive rabbets are also defined by three parameters; *Width, Depth*, and *Angle* (see fig. 6).

Next, the main horizontal portion of the keel, extending from the base of the stern post to the base of the stem, is created by sweeping the cross-section shape along a line of N length, defined by the *Keel Length* parameter.

The stem post is created according to Oliveira's treatise using a simple arc. A circular arc is placed with its center point one-third the length of the keel above the fore end of the horizontal portion of the keel. Its radius is equal to the height of the center point. The arc is cut where it meets the keel beam and at the height of the center point (see fig. 10). The keel's cross-section is then swept along this curve to create the stem post.

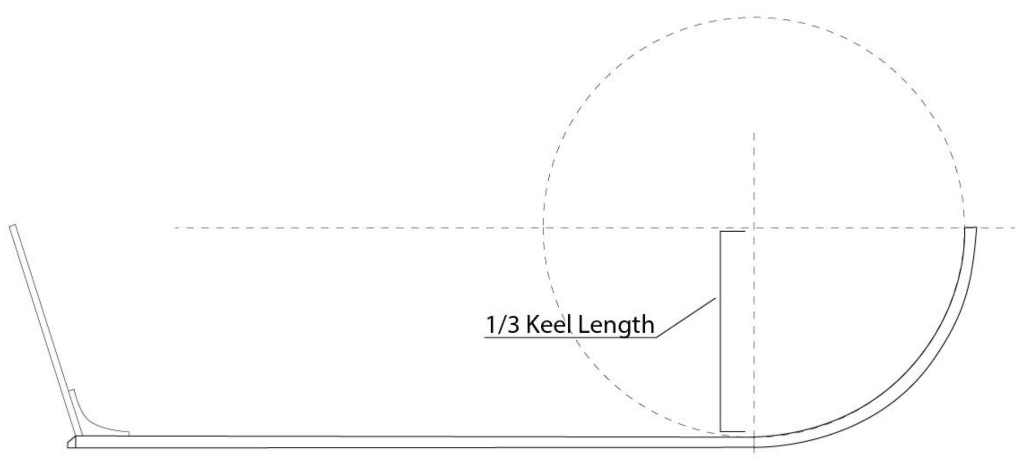

Figure 10. Stem dimensions

Given the length, depth, and angle parameters, the stern post is created by placing a box shape of the given dimensions at the aft end of the keel beam. The stern post uses the same width as the top of the keel cross section (see fig. 7).

The stern knee is created using a rectangular polygon placed where the stern post and keel meet. The polygon has the same width as the stern post. The bottom edge of the polygon is placed at the crease where the stern post meets the keel. The height of the polygon is determined by the *Height* parameter. The polygon is rotated about its base edge the same angle as the stern post. The bottom edge of the polygon is then extruded in the positive Z direction according to the *Length* parameter. The 'L' shaped geometry created is then extruded inward according to the *thickness* parameter. The inside edge is beveled to create a smooth curve (see fig. 11).

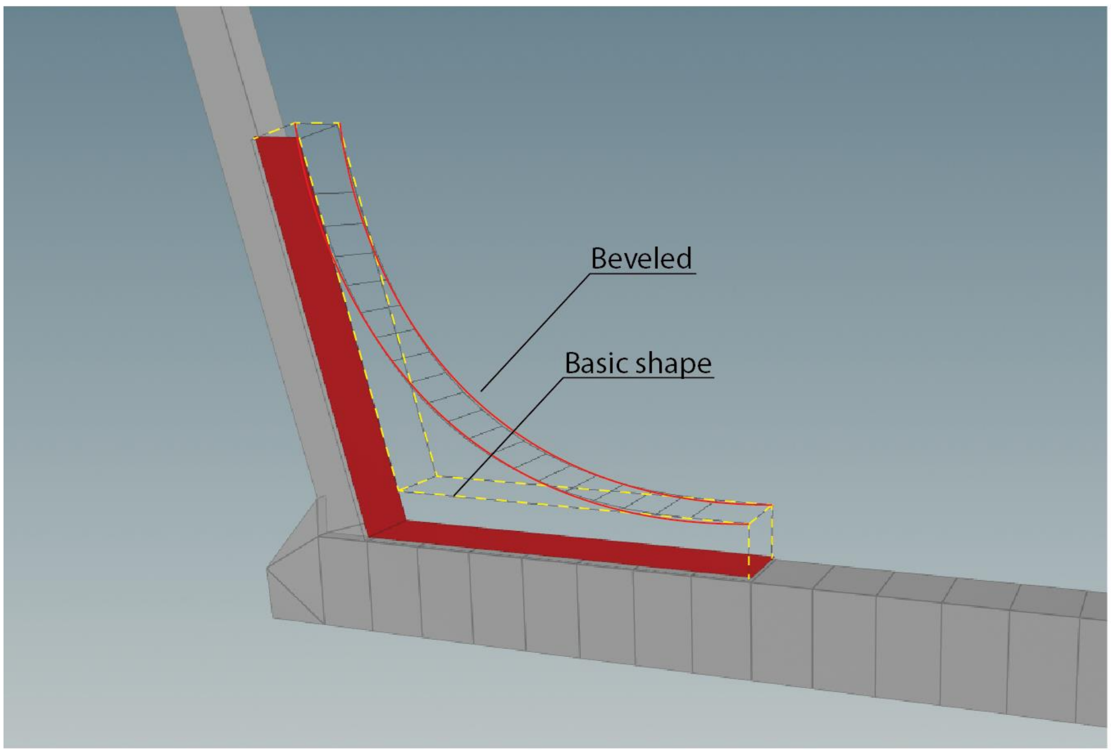

Figure 11. Stern knee construction

The skeg is 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 defines how much taller the skeg is than the keel cross-section height. 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 keels (see fig. 9).

All the subcomponents described above are assembled together to create the final ship axial structure. Houdini allows assignment of arbitrary data as attributes to any geometry. I utilize 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.

6.6.3. KEEL IMPLEMENTATION

The resulting model of the keel, or HDA, can be created inside of a Houdini geometry node. This will place a keel, of 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 custom keel model to be used to create an entire ship hull. Frames created based on the keel will be automatically updated by changes made to the keel HDA.

6.7. FRAMES

The *frames* are transverse timbers that provide the hull with support perpendicular to the keel. The widest of the frames, the master frame, divides the fore and aft portions of the ship. The *flat* of a frame is the horizontal portion of frame along the ship's bottom. The *turn of the bilge* is the curved portion of frame between the flat and the *side*, the near vertical portion of the frame. The *rising* and *narrowing* of the frames fore and aft the master frame are the terms to describe the narrowing of the flat and the rising of the bilge (see fig. 12).

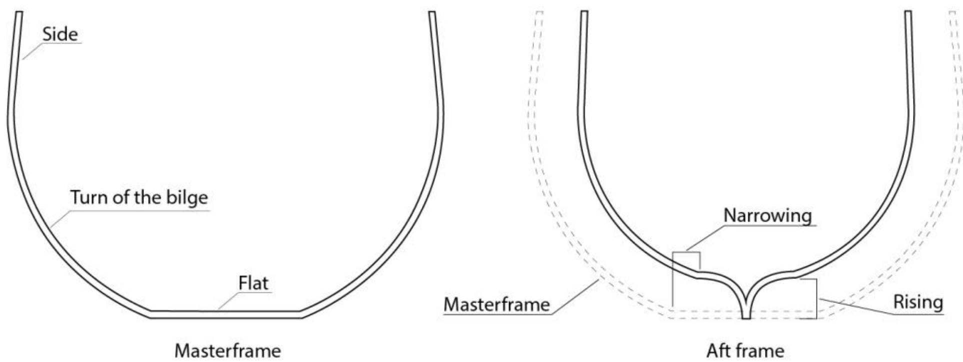

Figure 12. Frames

6.7.1. FRAME TAXONOMY

To describe Oliveira's frames, I used three basic shapes; a horizontal line for the flat, a semi-circular arc, and a straight line tangent to the end of the arc (see fig. 13). By describing these shapes, their relationship to each other, and their relationship to the keel, we can develop rules which guide the procedural model.

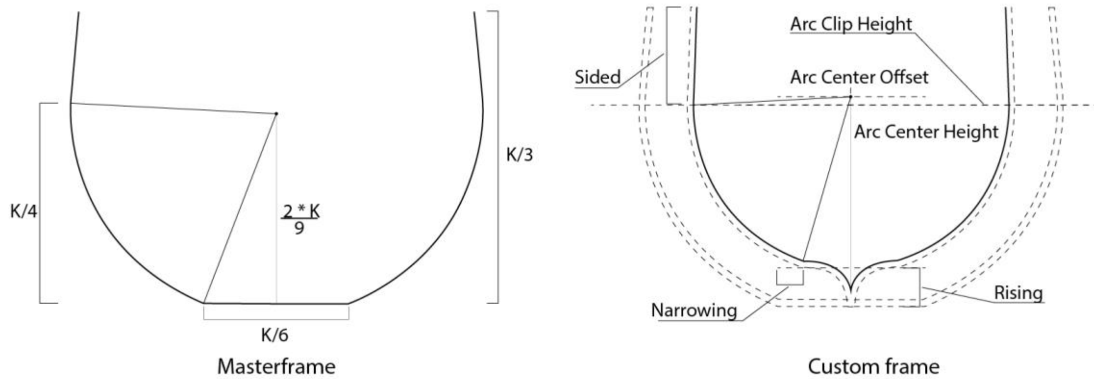

Figure 13. Frame parameters

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. According to Oliveira's treatise, the flat of the master frame is around one-sixth the length of the keel. For the rest of the frames, the length of the flat can be determined by the amount of narrowing of the flat, or the difference between the flat of the frame and the flat of the master frame. For my procedural frame model, the narrowing amount is provided as a parameter (see fig. 13).

A point is then placed above the center point of the flat, the height of which is provided as a parameter. The distance between this point and the end of the flat is the radius for the arc representing the turn of the bilge. According to Oliveira's recipe, the arc of master frame is drawn from the end of the flat until it reaches a height of around one-fourth the keel's length. For my model, if a frame is set to *Master Frame,* the width of the flat, the center point of the arc, and the height of the end of the arc are automatically determined based on the length of the keel. For all other timbers, I provide these user-defined values as parameters of the frame HDA: The center point of the arc can be adjusted in the X and Y dimensions. The height at which the arc is terminated can be customized. A *narrowing* parameter defines the width of the flat based on the master frame (see fig. 13). Offsetting the X-position of the center point of the arc is used to adjust the steepness of the turn of the bilge. This is typically used for the frames along the stem where the ship's hull narrows.

From the end of the arc, a straight line is drawn representing the semi-vertical portion of the frame along the ship's side. According to Oliveira's treatise, the height of the side of the master frame is one-twelfth of the keel's length. In my model, the height of the side at the master frame is automatically determined according to Oliveira's recipe. For all other frames, the height of the side is provided as a user parameter of the frame (see fig. 13).

The final parameter describing the shape of the frame is the *rising* dimension. In my model the rising describes the distance frame flat is offset vertically from the keel. By placing the flat of the frame above the keel creates a Y frame. To create a Y frame, my model connects the end of the flat to the base of where the frames connect to the keel using two lines which meet a 90-degree angle. The corner of the two lines is connected and beveled (see fig. 13).

To complete the frame's parameterization, we also need to describe its position along the keel. There are two categories of frames, the master frame and all other frames. Based on Oliveira's treatise, we know that the master frame is placed fore of the mast step by 1.5 * K/18 (K = Keel length) (Cook 50). Oliveira's treatise also says that the mast step is placed at half the keel's length. From these two proportions, we can calculate the exact position of the master frame (see fig. 14). The rest of the frames do not have specific position along the keel. To mimic the relationship the master frame has to the keel, in my model the placement of all other frames are determined using a ratio to the keel's length. I chose to divide the frames into two sub-groups; those placed along the horizontal portion of the keel and those placed upon the stem post. Frames are then placed along either the keel or the stem post. Their placement along the keel or stem is determined by a value between zero and one; zero representing the aft end and 1 the fore end. By using the normalized position instead of the exact distance along the keel, this makes the frame models more flexible. If the keel length is changed, the frames maintain their relative positions along the keel and thus maintain the hull proportions.

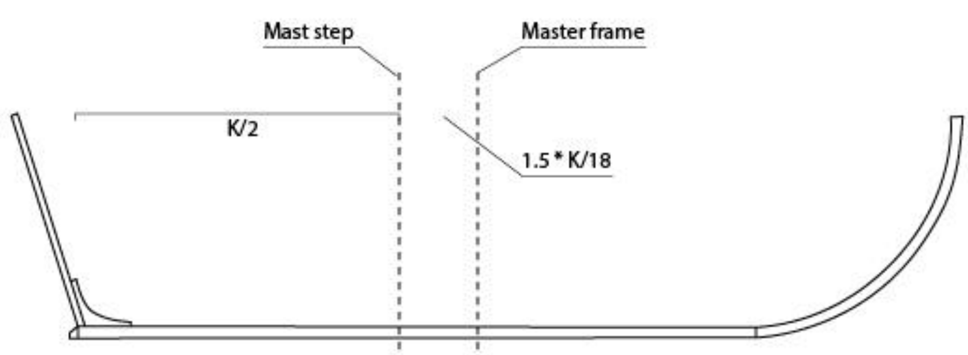

Figure 14. Master frame position

6.7.2. FRAME CONSTRUCTION

Using the parameter values and the rules from the taxonomy, we can construct spline curves which act as construction guides for the frame geometry. In my model I allow the user to specify whether or not to create futtocks. This option allows models to be double-framed, as most ships were in the late 17th century, or to have frames composed of floor timbers and futtocks, as they were before that. If the *Futtock* parameter is unchecked, the frame geometry will be created by sweeping a rectangular cross section along the guide curve. The dimensions of the rectangular cross section are user defined via parameters. If futtocks are turned on, four additional parameters are needed; *First Futtock Height, Second Futtock Height, First Futtock Overlap,* and *Second Futtock Overlap.* To create the futtocks, three copies of each frame are created. These duplicates are placed one after the other in the Z-dimension, creating three sequential frame geometries for each frame. The First Futtock Height parameter defines the height (Y-dimension) at which the middle frame ends and the outside frames begin. The Second Futtock Height parameter defines the height at which the outside frames end and the middle frame begins again. From the height of the second futtock, only the middle frame continues to the final height of the frame. The Futtock Overlap parameters define the amount the middle and outside futtocks overlap one another (see fig. 15).

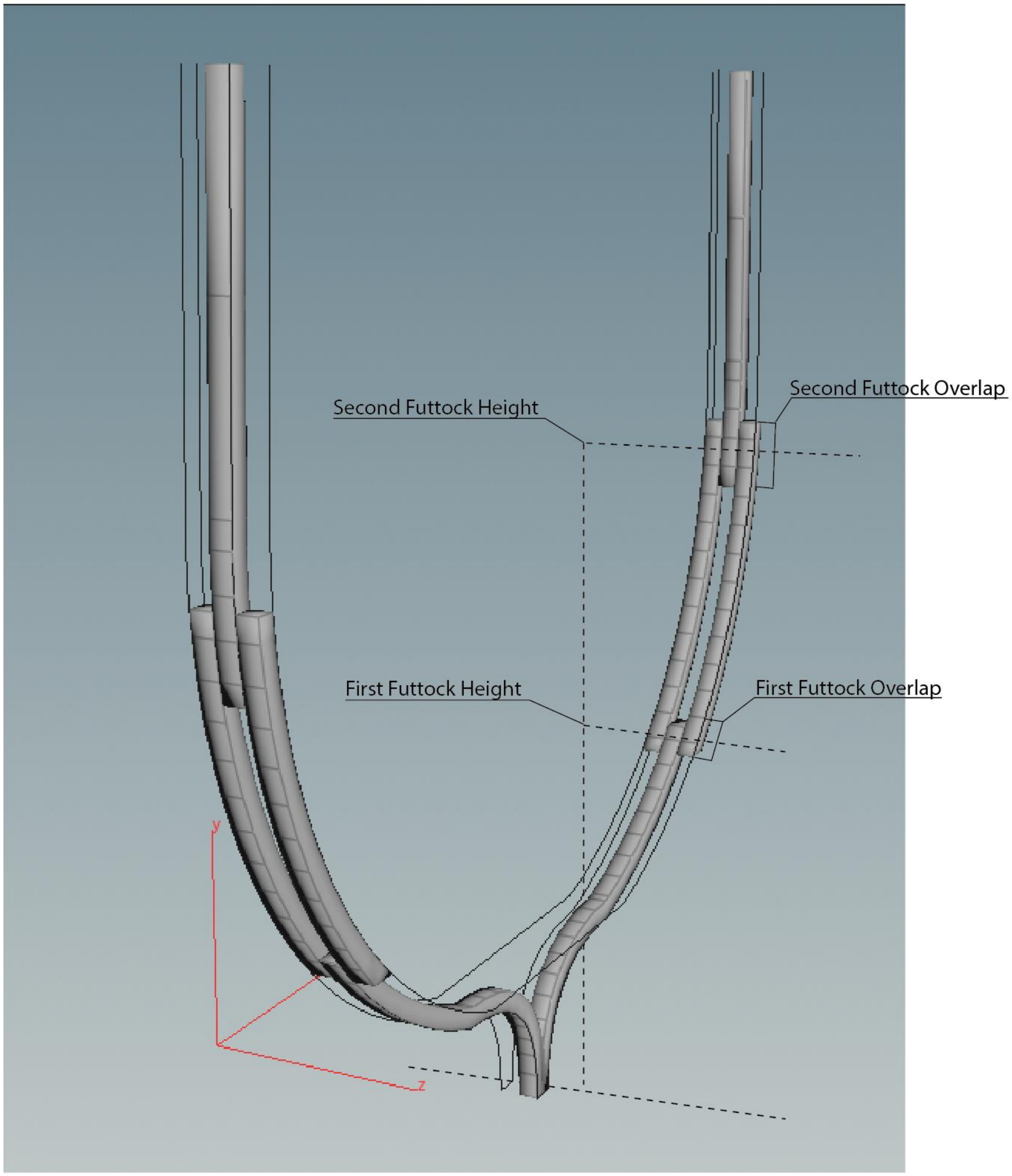

Figure 15. Frame futtocks

6.7.3. FRAME IMPLEMENTATION

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 *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 of the hull. Any changes to the keel will automatically update the proportions of the frames.

6.8. STERN PANEL

The stern panel is another sub-group of timbers within the transversal group. The stern panel sub-group consists of fashion pieces and transoms. The fashion pieces are fixed to the stern post and the transom sits atop the fashion pieces and the stern post. Within this sub-group there are also lower transoms, which span the distance between the fashion pieces (see fig. 16).

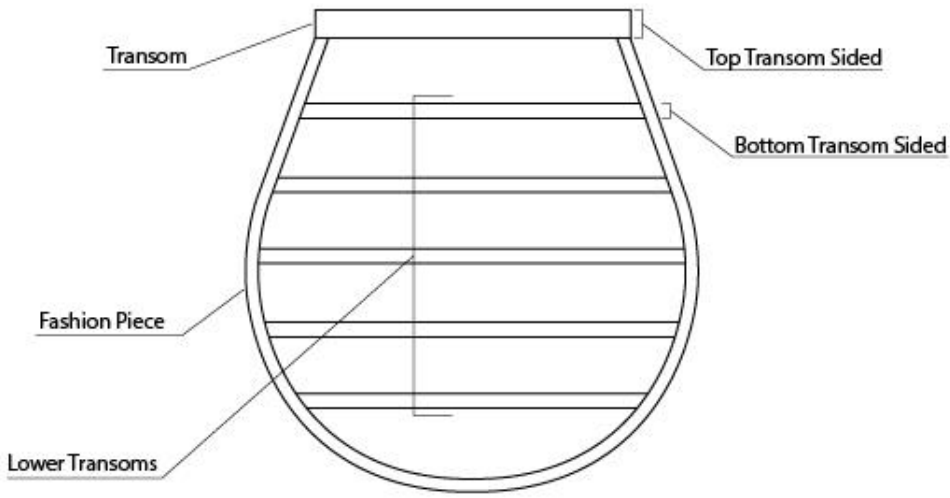

Figure 16. Stern panel components

6.8.1. STERN PANEL TAXONOMY

For Oliveira's recipe, the fashion pieces of the transom are designed similar to the frames. Oliveira's treatise calls for the center of a circle to be placed at a height of around two-ninths of the keel's length. The radius of the circle should be one-ninth the keel's length. The circle is drawn until it reaches a height of one-fourth of the keel's length. A line is drawn tangent to the end of the circle to a height of one-third the keel's length creating the side of the fashion pieces. Oliveira's treatise also specifies that the width of the transom should be around one-fourth the keel's length. For my model, I allow the user to create the stern panel according to Oliveira's proportions or as a custom stern panel with variable dimensions. The fashion pieces of the custom stern panel are created very similar to the custom frames (see fig. 17). The main difference between the two is that the fashion pieces have no flat and the height of the center point of the circle and radius of the circle can be determined independent of one another. Since the transom is placed atop the fashion pieces, the height of the semi-vertical portion of the fashion piece determines the placement of the transom. The lower transoms are spaced evenly to fill the space between the bottom of the fashion pieces and the bottom of the transom. The lower transoms width can be determined by the distance between the fashion pieces at a particular height along the sternpost. The spacing and shape of the lower transoms is therefore determined by the number of transoms and the shape of the fashion pieces.

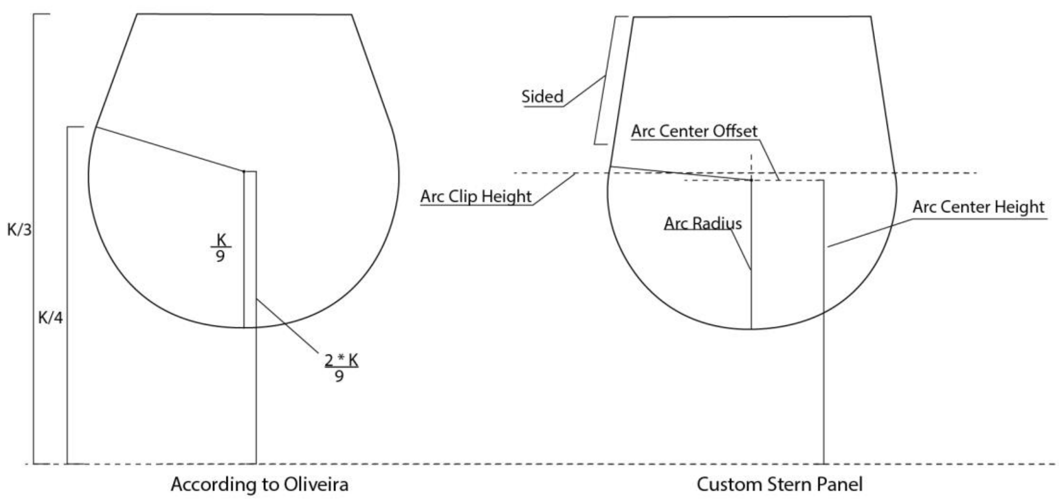

Figure 17. Stern panel parameters

6.8.2. STERN PANEL CONSTRUCTION

Similar to the construction of the frames, by using the parameter values and the rules from the taxonomy, we can construct a spline to guide the construction of the stern panel geometry. The fashion pieces are created by sweeping a rectangular cross section along the guide curve created by the arc and sided dimension. The dimensions of the rectangular cross section are user defined via parameters. The molded dimension of the top transom and lower transoms can be defined via two parameters, *Top Transom Sided* and *Lower Transoms Sided* (see fig. 17).

6.8.3. STERN PANEL IMPLEMENTATION

The resulting model of the stern panel, or HDA, can be created inside of a Houdini geometry node. To create a stern panel, a keel HDA must be used as input. By connecting the keel's output to the stern panel HDA's input, the keel passes along its attributes and the stern panel will be created according to its relationship to the keel as defined by the taxonomy. The stern panel can be set to *Based on Keel* or *Custom Frame.* Setting the frame to "based on keel" will automatically create a stern panel based on the Oliveira recipe. Stern panels set to "custom" will have all parameters available to create a stern panel according to Oliveira's treatise with variable dimensions. Adjustments to the parameters on the stern panel HDA will have an immediate effect on the model shown in the viewport. Any changes to the keel will automatically update the proportions of the stern panel HDA.

6.9. LONGITUDINAL REINFORCEMENTS

Longitudinal reinforcements is a group which contains whales, stringers and breasthooks. Whales are timbers which provide longitudinal tensile support along the length of the hull on the exterior of the frames. Stringers also provide a hull with longitudinal support along its length, but stringers run on the interior of the frames. The breasthook is a large V-shaped timber at the bow of the ship, used to connect the stringers together, and to the stem (see fig. 18).

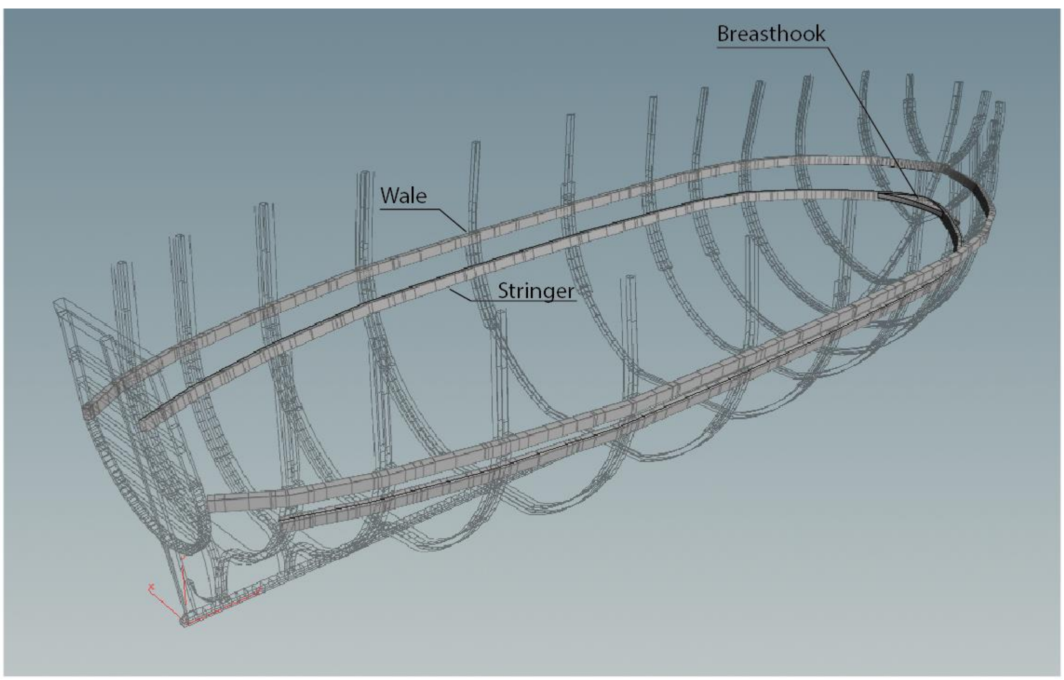

Figure 18. Longitudinal reinforcements: wale, stringer, and breasthook

6.9.1. LONGITUDINAL TAXONOMY

The timbers in this group are determined by the shape of the frames. The whales follow the exterior of the frames, the stringers follow the interior of the frames, and the breasthook follows the interior of the frames at the bow of the ship. Typically, a shipwright would eye-ball the placement of a stringer or whale by gazing down the length of the timber to determine a “fair” curve and shape. The stringers and whales can be imagined as curves that lie in the intersection of two surfaces which curve in two dimensions. One surface is defined by the frames. The other is a nearly horizontal surface establishing the placement of the whale or stringer, either on the inside of the frames or the outside (see fig. 19). The breasthook is created using the same method as the stringer.

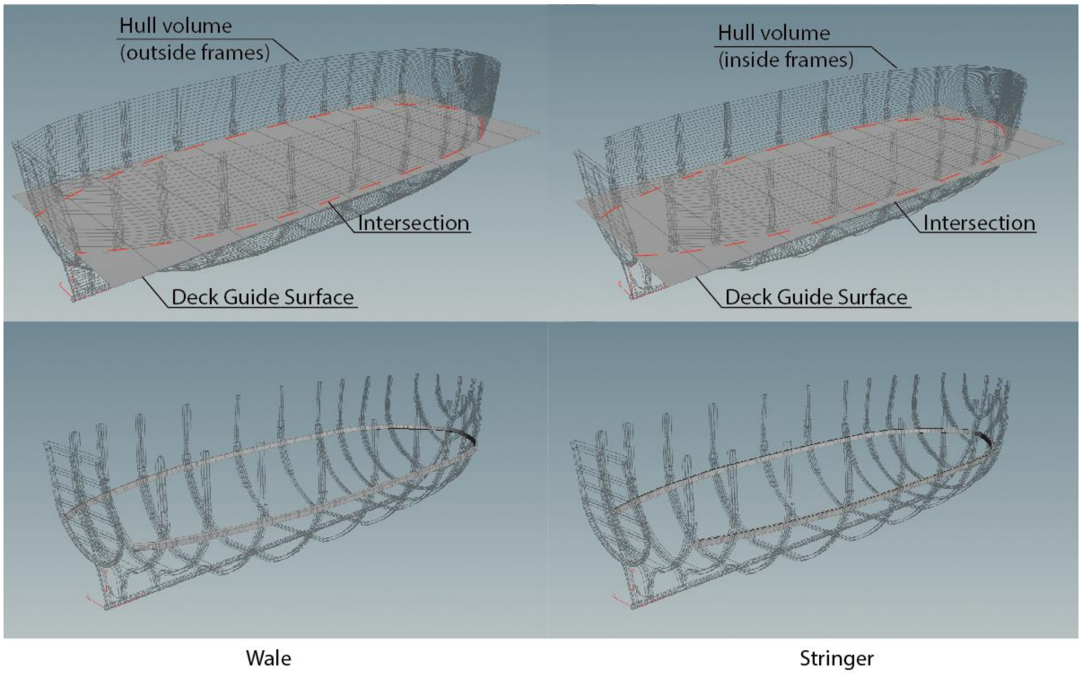

Figure 19. Longitudinal reinforcements: wale and stringer construction

6.9.2. LONGITUDINAL CONSTRUCTION

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. A horizontal surface is then created from the stern to the stem. The intersection surface curves along its longitudinal axis. The curvature of this surface is determined by the height of the plane at the stern, the height at the stem, and the height at the midship, via parameters. Oliveira suggests that the heights of the lower stringers and wales on the stem and stern posts should be around one-third of the total height, or one-twelfth of the keel length. The curve defined by the intersection of this surface and the hull shape from the outside surface of the frames is used as a guide to create the whales. The intersection of the surface and the hull shape from the inside surface of the frames is used as a guide to create the stringers (see fig. 19). The intersection plane can be rotated about its Z axis, which allows the digital asset to place a stringer along the flat of the hull by rotating the intersection surface to intersect with the lower portion of the hull.

The model for these timbers is created by sweeping a rectangular cross section along the spline created by the intersection of the surface and the hull shapes determined by the frames. The dimensions of the rectangular cross section are user defined via parameters. The cross section can be rotated by a parameter to precisely adjust the orientation of the wale or stringer along the exterior or interior of the frames.

The breasthook is constructed similar to the stringers, as it is a continuation of these timbers into the stem; by finding the intersection of the hull shape defined by the inner surface of the frames and an imaginary 3D plane. In the case of the breasthook, the 3D intersection surface is not curved and is oriented horizontally. Unlike the stringers, the breasthook does not extend the length of the hull, it tapers from its widest point at the stem towards the stern. For my model the length of the breasthook is determined by a parameter. The two sides of the curve created by the intersection are then connect by a corner whose depth and bevel are be determined by two parameters; *Inner Curve* and *Inner Bevel Amount* (see fig. 20).

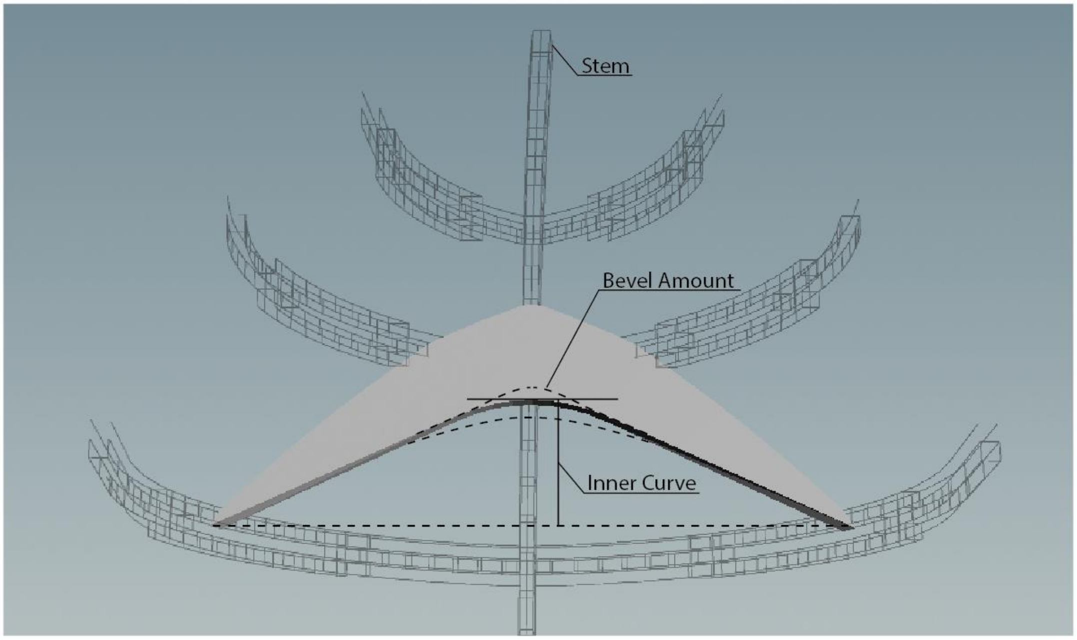

Figure 20. Breasthook: inner curve and bevel

6.9.3. LONGITUDINAL IMPLEMENTATION

There are two HDAs which make up the longitudinal reinforcements group, the longitudinal reinforcement HDA, which can create a model of a wale or stringer. The second, creates a model of a breasthook.   These HDAs can be created inside a Houdini geometry node. A longitudinal reinforcement needs three sources of input; a keel HDA, a stern panel, and a group of frames. The output of the keel HDA is connected to the first input of the longitudinal reinforcement HDA. The outputs from all of the frame HDAs need to be first connected to a merge node. The output from the merge node can then be connected to the second input of the longitudinal reinforcement HDA. Finally, a stern panel HDA should be connected to the third input. This will provide the longitudinal reinforcement HDA with all the necessary information to create a fair whale or stringer. The shape of the intersection surface can be controlled via the parameters; *Height at Stern, Height at Stem, Height at Midship*, and *Midship Z Shift.* The breasthook HDA requires the same input as the longitudinal reinforcement HDA. The height in the Y dimensions of the breasthook at the bow of the ship is determined by the *Breasthook Height* parameter. The distance the breasthook spans, along the inside of the frames, toward the stern from the stem, is determined by the *Length* parameter. The sided measurement of the timber can be adjusted via the *Sided* parameter of the breasthook HDA.

6.10. DECK STRUCTURES

The deck structures is a group of timbers which contains both transverse and longitudinal timbers. These timbers include; beams, knees, carlings, clamps, waterways, and *coceira*. The coceira is a timber that runs vertically over the waterways. These timbers were grouped together because of their common purpose of constructing a horizontal platform within the volume of the hull (see fig. 21).

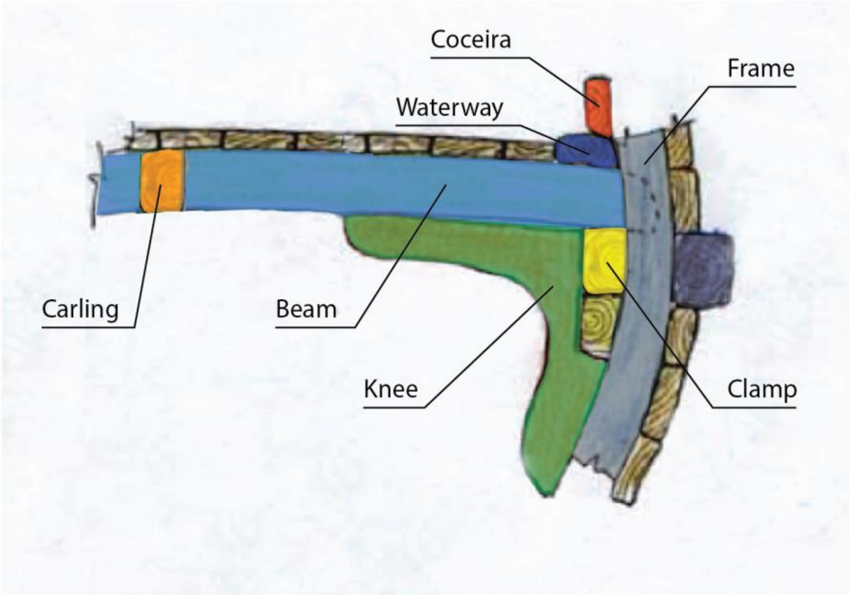

Figure 21. Deck component section

6.10.1. DECK TAXONOMY

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 can 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 (this will be elaborated further for each sub-component of the deck) (see fig. 22).

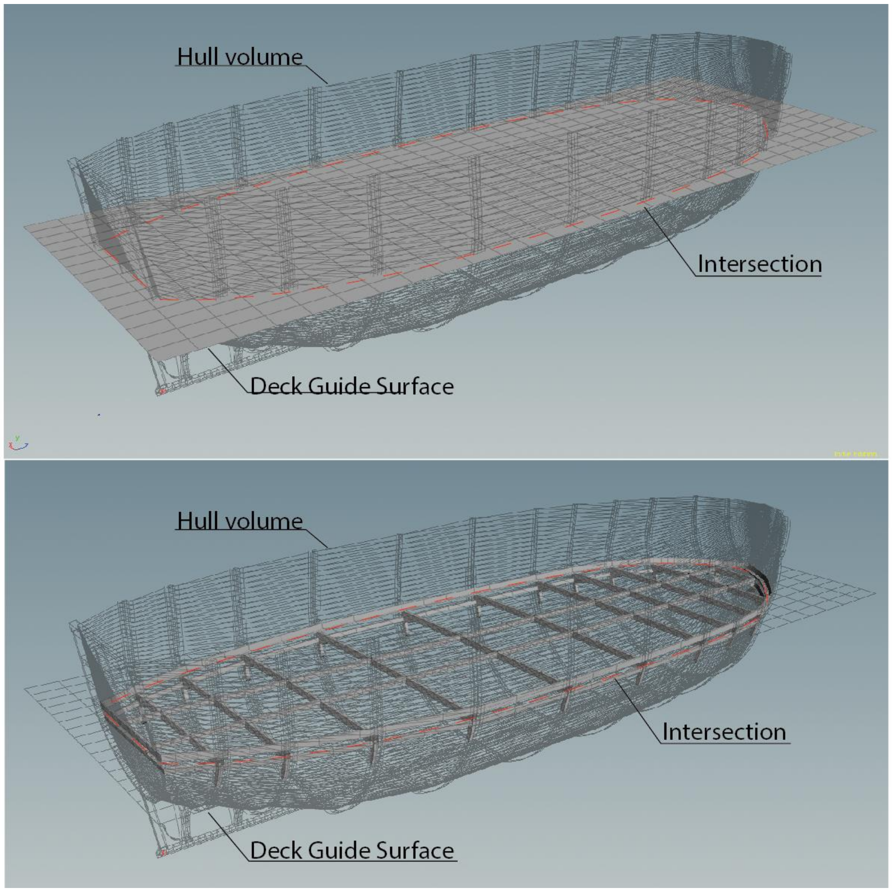

Figure 22. Deck structures: guide surface construction

The placement of the beams 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 molded dimension (see fig. 23).

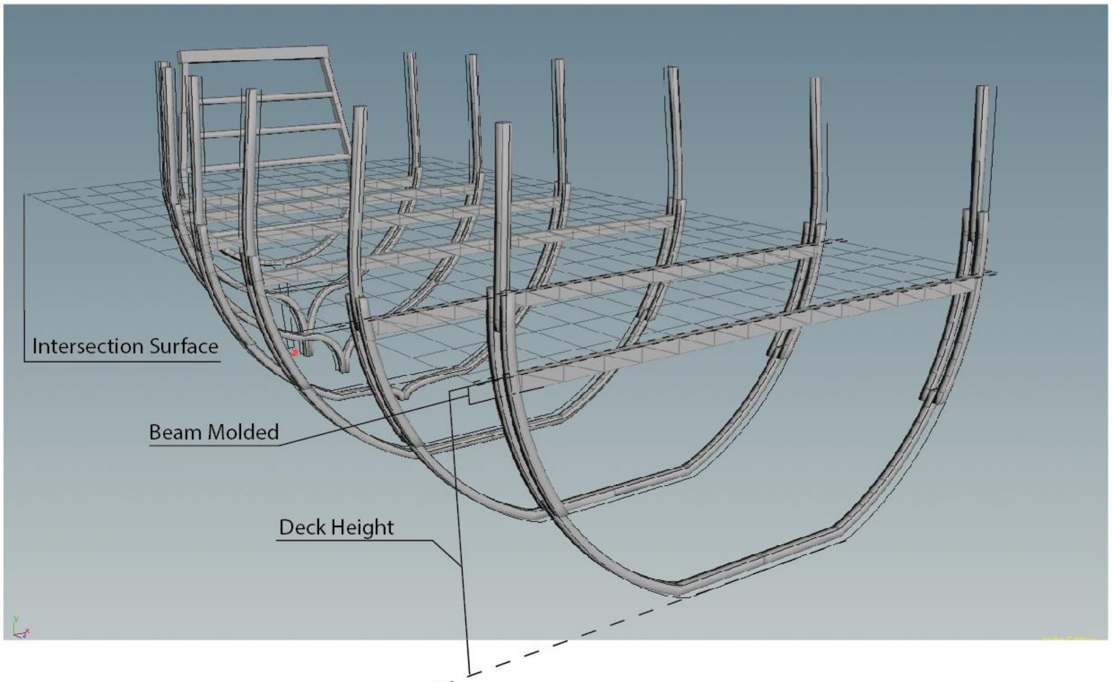

Figure 23. Deck beams: guide surface construction

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. For this model I created *hanging knees,* which is an upside-down 'L' shaped timber which the beam sits on. Some ships have *standing knees,* which fit over the waterway and *coceira.* The scope of this project only includes hanging knees, but the same techniques could be extended to include standing knees as well. The shape of any knee can be determined by the inside surface of the frame where it's attached, and by the bottom surface of the beam which meets the frame. By offsetting from these two surfaces, by the amount of the knee's molded measurement, we can determine the knee's shape (see fig. 24).

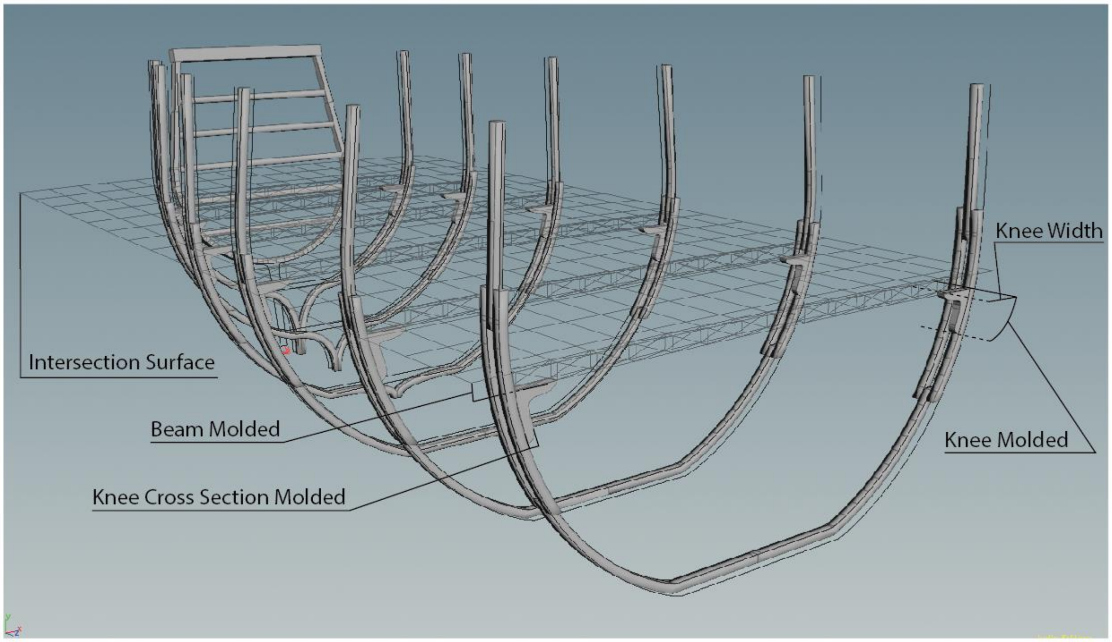

Figure 24. Deck knee: guide surface construction

With the transverse deck timbers defined we can now define the longitudinal deck timbers. Carlings are two longitudinal beams, equidistant from the center line of the deck, which run longitudinally the length of the deck. The top surfaces of the carlings are aligned with the top surfaces of the beams. This means it's shape can be inferred directly from the deck intersection surface. Using the beams sided and molded measurement and the distance from the center line, we can fully describe the carlings (see fig. 25).

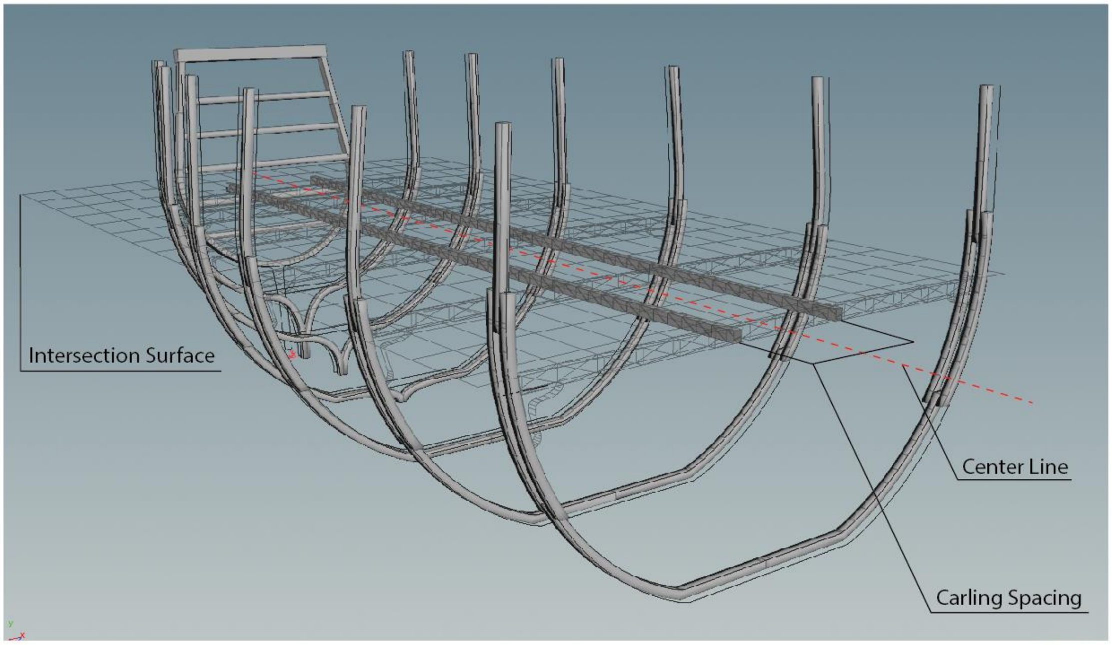

Figure 25. Deck carlings: guide surface construction

The clamp is a longitudinal timber which sits beneath the deck beams, along the inside surface of the frames. The shape of the clamp can be determined by using the molded and sided measurement to offset vertically and horizontally from the intersection surface and the surface of the hull volume determined by the inside faces of the frames (see fig. 26).

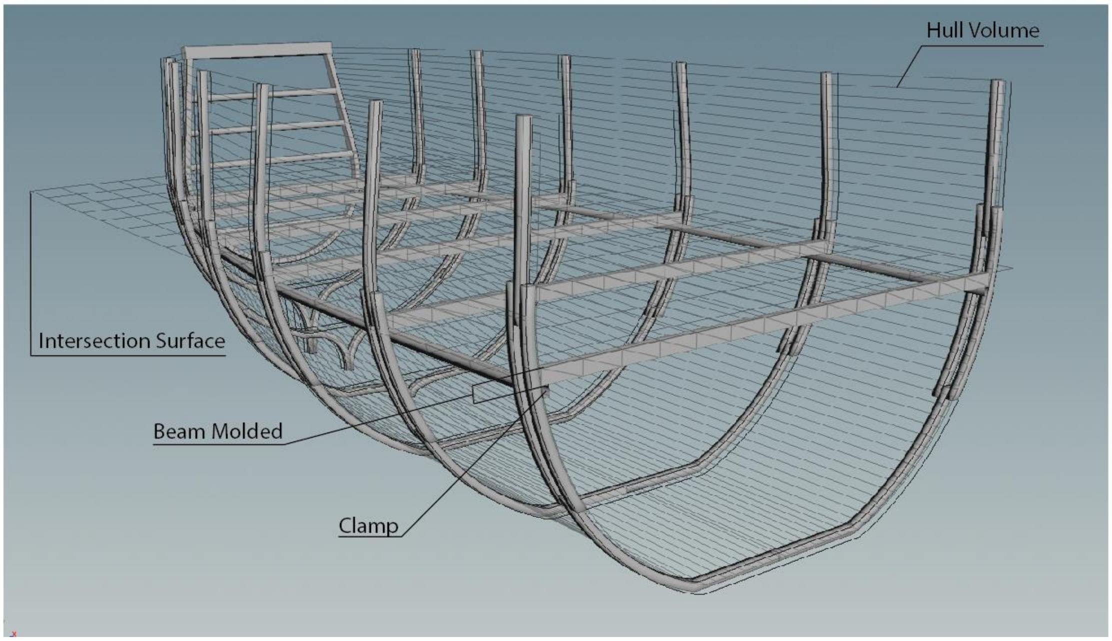

Figure 26. Deck clamp: guide surface construction

The waterway, another longitudinal deck timber, is placed on top of the beams following the surface of the hull volume, defined by the inside faces of the frames. The shape of the waterway can be determined by offsetting from the intersection surface and the hull volume. The geometry for the molded surfaces can be found by offsetting from the deck intersection surface by the amount specified by the waterway's molded measurement. The sided faces of the waterway can be determined by offsetting from the hull volume surface by the amount specified by the waterway's sided measurement (see fig. 27).

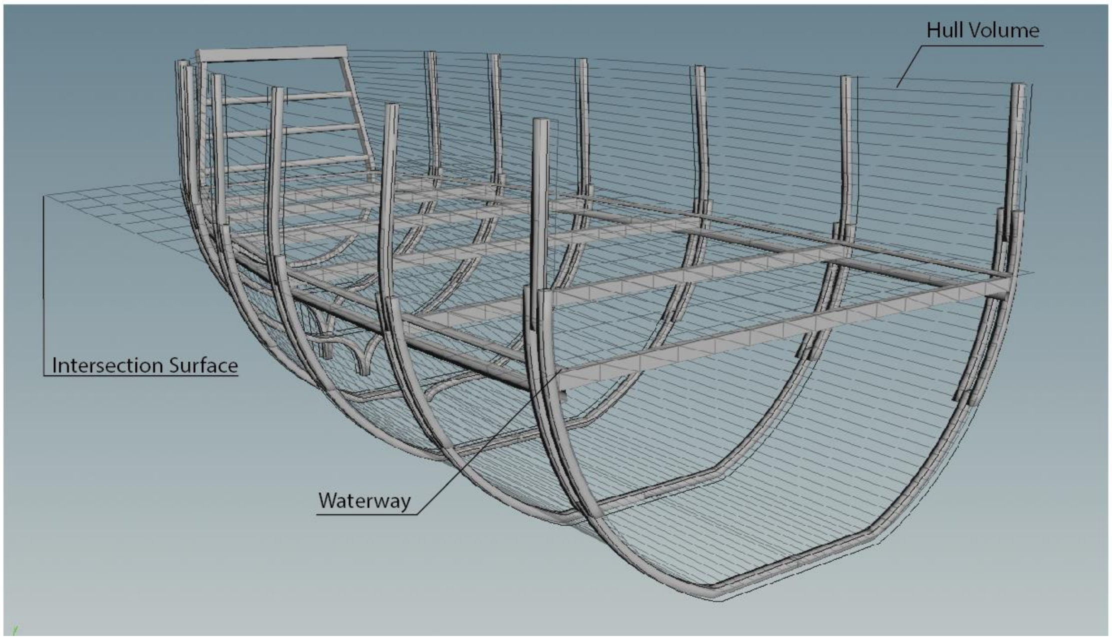

Figure 27. Deck waterways: guide surface construction

The *coceira*, which aligns perpendicularly to the waterway, can be determined by offsetting from the hull volume and the top surface of the waterway. Offsetting from the hull volume surface by the amount of the coceira's molded measurement defines the coceira's molded faces. Offsetting from the top surface of the waterway by the amount of its sided measurement creates the sided faces of the coceira (see fig. 28).

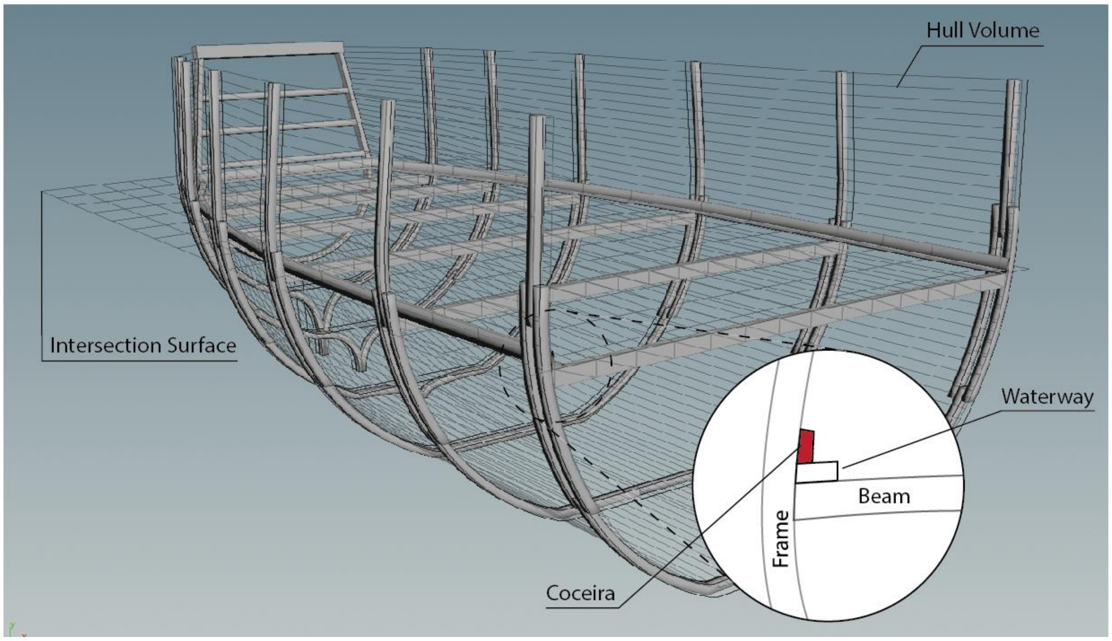

Figure 28. Deck coceira: guide surface construction

6.10.2. DECK CONSTRUCTION

The construction of the deck timbers is very similar to the definition of their taxonomy. The timbers are created based on their relationship to the deck intersection surface, which is user defined, and by their relationship to one another. The first step in the creation of the deck timbers is to create a hull volume surface from the frames and the intersection surface. The hull volume is created by connecting the inside faces of the frames to one another to create a continuous surface. The intersection surface is a horizontal surface which represents the desired deck. The intersection surface is placed at a user defined height at the center of the ship in the Z-dimension. The intersection surface can be bent in both the longitudinal and transverse directions to create the desired deck camber (see fig. 29). The portion of the intersection surface, inside the hull volume, represents the extents of a deck at that particular height.

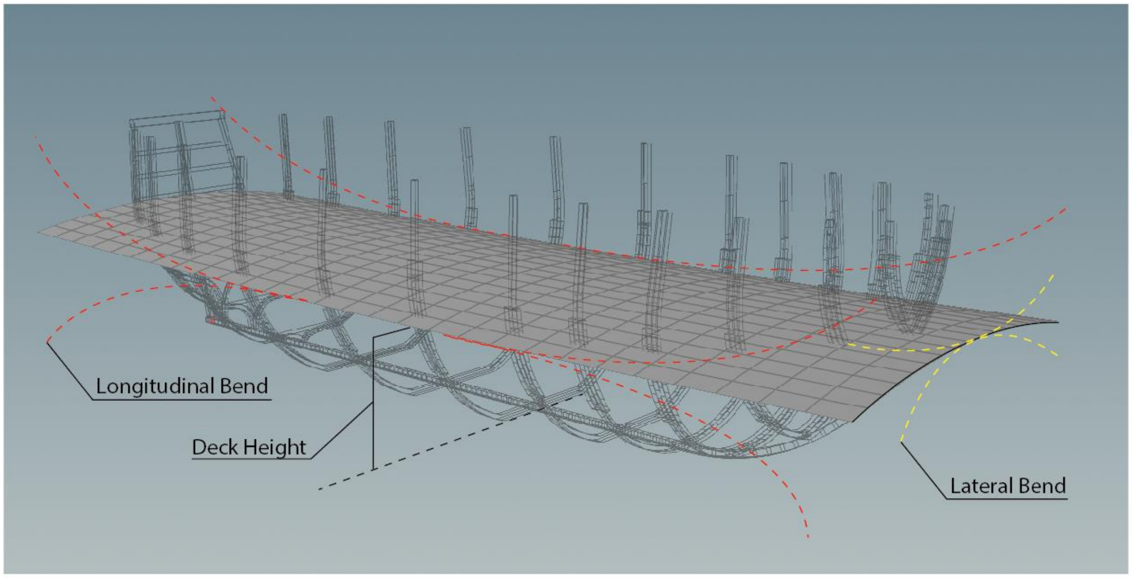

Figure 29. Deck bend: guide surface construction

The deck beams are created by cutting the intersection surface where it intersects the hull volume created by the inside surfaces of the frames. The intersection surface is then cut fore and aft of each frame, discarding the spaces between, to create strips of geometry between each frame at the height of the deck. The geometry strips created represent the top surfaces of the deck beams (see fig. 23). The intersection surface is then lowered by the amount of the deck beam's molded measurement and another set of geometry strips are created representing the bottom surfaces of the beams. The two strips are connected by polygons to create the final beam geometry.

The deck knees are constructed by clipping the bottom surface of the deck beam in the YZ plane, at a distance defined by the *Knee Width* parameter. This distance is offset from the point where the deck beam meets the frame. The inner surface of the frame is cut then in the XZ dimension at the height where the beam meets the frame and also offset from that height by the amount defined by the *Knee Molded* parameter. The two geometry strips created are connected and extruded inward, towards the center of the hull, to make the basic shape of the knee (see fig. 30). The inside corners are then beveled to create a smooth knee shape.

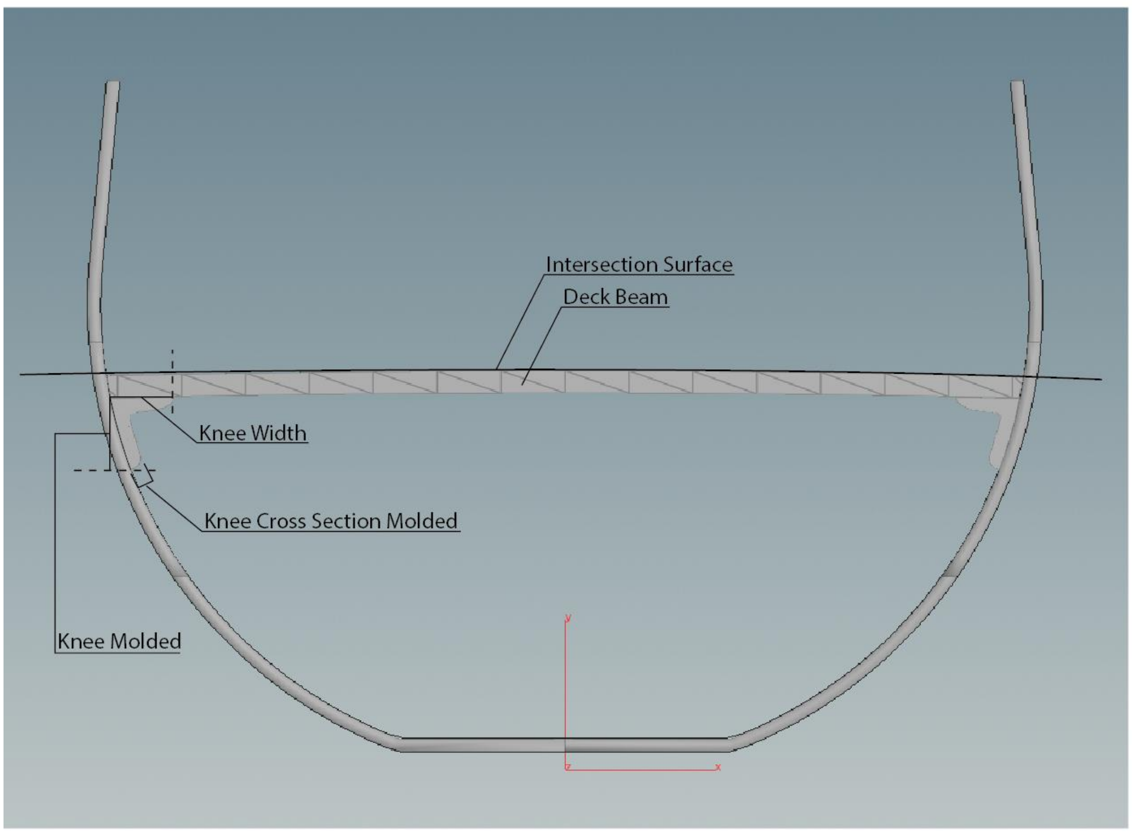

Figure 30. Deck knee dimensions

To create the carlings, two longitudinal splines are placed along the intersection surface at the user defined distance from the center (see fig. 25). Traditionally these two carlings would define the width of the hatches, through which cargo would be lowered to the holds. The splines are then cut at the point where they meet the hull volume surface. By sweeping a cross section, of user defined dimensions via parameters, along the splines we create the geometry of the carlings.

The clamp is created by offsetting from the intersection surface in the negative Y direction by the amount of the deck beam's molded measurement. Next, the intersection surface is cut at the crease line where it meets the hull volume surface.   The cut line represents the top edge of the clamp. The intersection surface is then offset again in the negative Y dimensions by the amount of the clamp timber's molded measurement, which is user defined via a parameter. The geometry strip created between the top and bottom crease lines represents the outward molded face of the clamp timber. By extruding this strip inward by the amount of the clamp timber's sided dimension, we create the geometry of the clamp (see fig. 26).

The waterways are constructed very similar to the methods used for the clamp. First, the intersection surface is cut at the crease line where it meets the hull volume surface. Next, the intersection surface is offset in the positive Y dimension, by the amount of the waterway timber's molded measurement, and cut again at the new crease line. The two crease lines are connected to form a geometry strip which represents the outward molded faces of the waterway. The geometry strip is then extruded inward by the amount of the waterway's sided measurement, creating the final shape of the waterway (see fig. 27).

The *coceira* is constructed exactly like the clamp, however the intersection surface is first offset in the positive Y dimension by the amount of the waterway's molded measurement. The intersection surface is cut at the crease line created hull volume surface. The intersection plane is then offset in the positive Y dimension by the amount of the coceira's molded measurement. The geometry strip created, represents the outward molded face of the coceira. The geometry strip is then extruded inward to create the final coceira model (see fig. 28).

6.10.3. DECK IMPLEMENTATION

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, waterways, and *coceiras*, 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.

6.11. STANCHIONS

Stanchions are vertical timbers which provide support for the deck beams.

6.11.1. STANCHION TAXONOMY

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. In each XY frame plane, the stanchions are positioned between the carlings of the deck above and deck below.

6.11.2. STANCHION CONSTRUCTION

To construct the stanchions, a bottom-most stringer, where the base of the stanchions attach and at least one deck must be defined. The first step is to create the bottom most set of stanchions, these connect the stringer to the lowest deck. A stanchion is placed where the stringer meets a frame. Transversely, the XY placement of the base of the bottom-most row of stanchions is based on the location of the stringer along the flat of the frames. The top of the bottom-most set of stanchions is positioned transversely by the location of the carlings of the lowest deck. Stanchions between the lowest deck and any remaining decks are positioned the same way, transversely by the position of the carlings, longitudinally by the position of the frames, and spanning between upper and lower beams. When the stringer HDA was created, it used the frames as input. This allowed the longitudinal points at which the stringer meets a frame to be stored as an attribute (see fig. 31). 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.

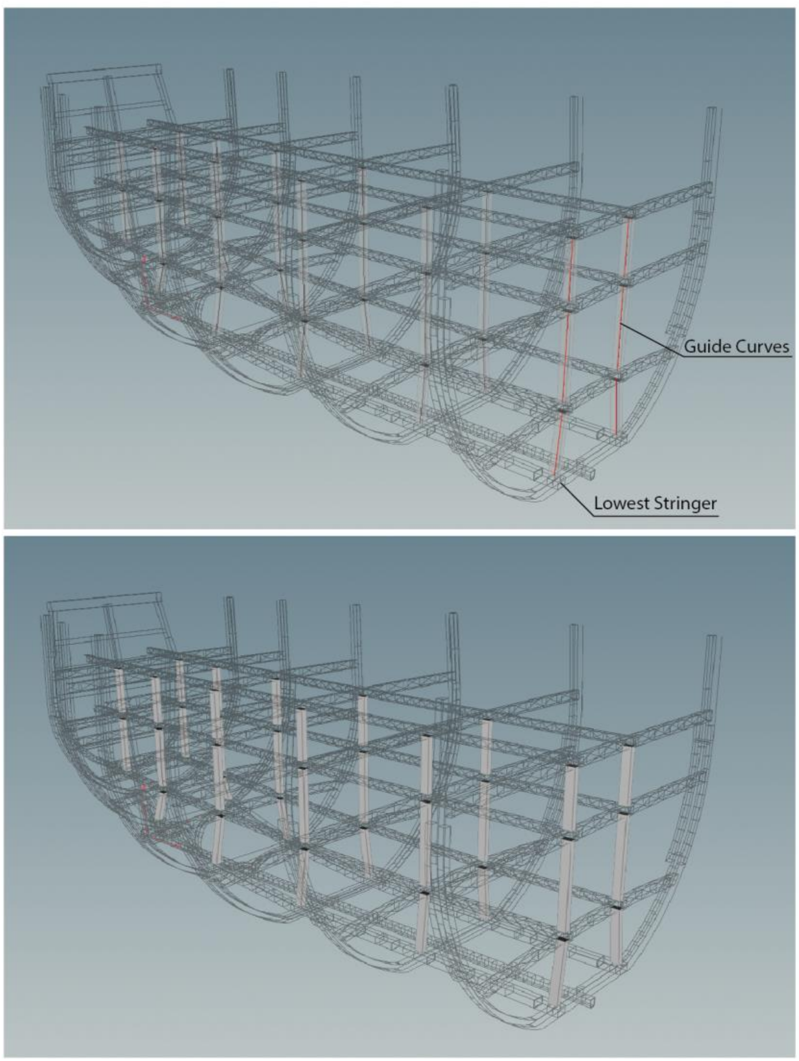

Figure 31. Stanchions: construction

6.11.3. STANCHION IMPLEMENTATION

The resulting model of the stanchions, or HDA, can be created inside of a Houdini geometry node. To create stanchions, there needs to be two inputs; a longitudinal reinforcement HDA which is the bottom-most stringer, and a deck or merged group of deck HDAs. This will provide the stanchions HDA with the necessary information to create stanchions between the decks of the hull. The stanchions HDA will automatically create stanchions between the stringer and the decks. The user can define the dimensions of the stanchion cross section via parameters. Any changes made to the keel, frames, stern structure, stringer, or decks will automatically update the stanchions timbers.

6.12. KEELSON

The keelson is a part of the main ship spine group. The purpose of the keelson is to provide stability to the frames along the horizontal portion of the keel by fastening the frames to the keel below. At the middle of the keelson is the maststep, an enlarged section of the keelson with a rectangular mortise for the base of the mast (see fig. 32).

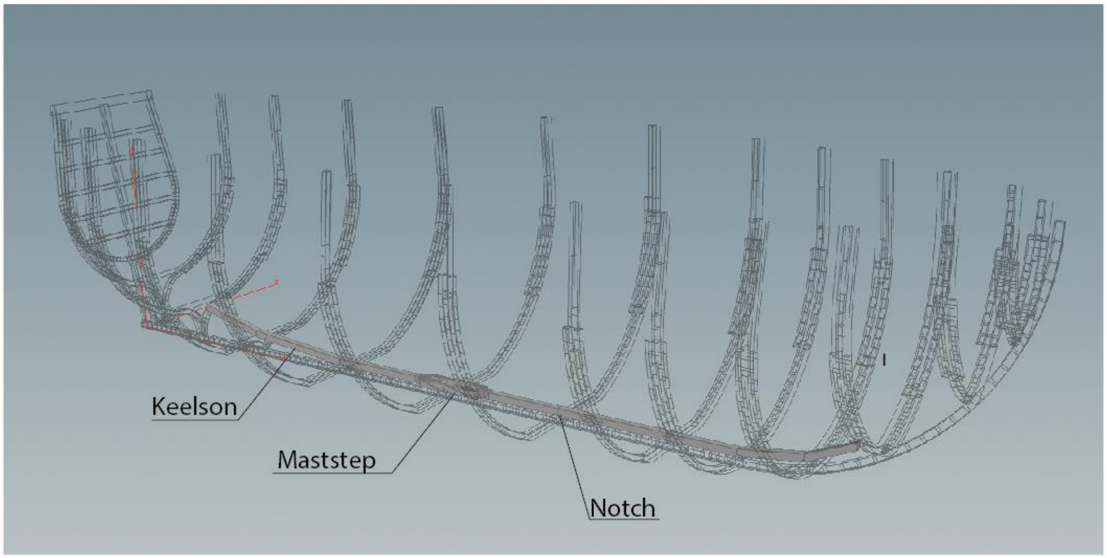

Figure 32. Keelson

6.12.1. KEELSON TAXONOMY

The keelson sits atop the keel, sandwiching the frames along the flat. The keelson does not span between all of the frames. It typically starts at the point where the frames change from ‘V-shaped’ to ‘Y-shaped’ at the stern. The keelson is notched at each frame so that it locks in over them. The maststep is the wider portion of the keelson positioned slightly abaft the masterframe (see fig. 32).

6.12.2. KEELSON CONSTRUCTION

The keelson has a rectangular cross section and follows along the lowest point of the frames. A transverse notch is cut out of the keelson where it meets each frame. The depth of the notch is determined by the amount the keelson is lowered over the frames. Measurements of the maststep vary from ship to ship so for this model all the measurements are provided as parameters.

6.12.3. KEELSON IMPLEMENTATION

The resulting model of the keelson, or HDA, can be created inside of a Houdini geometry node. To create a keelson, a group of merged frames needs to be used as input for the keelson HDA. This will provide the keelson HDA with all the necessary information to create a keelson along the given frames. The keelson will automatically be 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 (see fig. 33). Any changes made to the keel or frames will automatically update the keelson HDA.

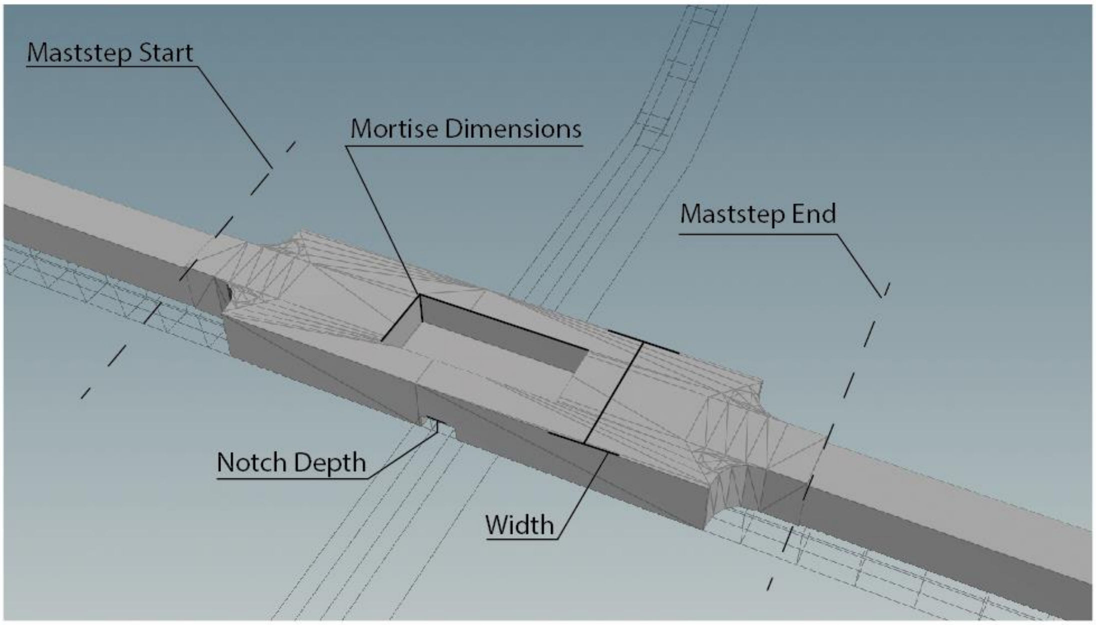

Figure 33. Maststep parameters

6.13. PLANKING

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 3).

6.13.1. PLANKING TAXONOMY

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 molded dimension along its length to compensate for the hull's three dimensional 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 molded 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 (see fig. 34).

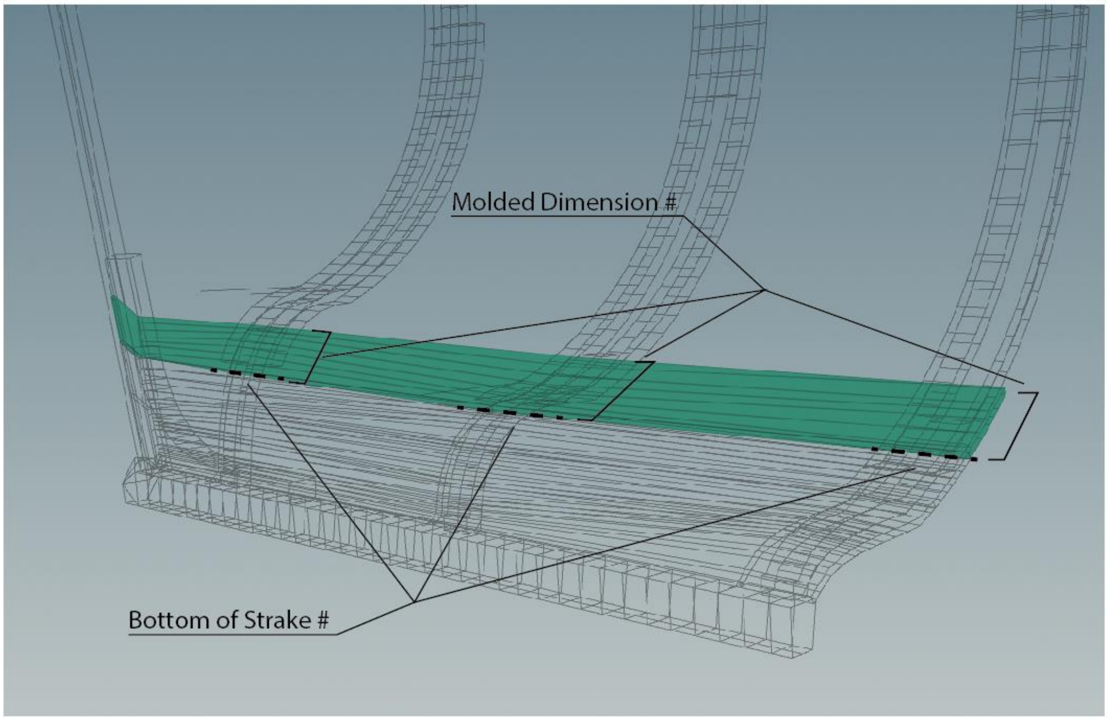

Figure 34. Planking parameters

6.13.2. PLANKING CONSTRUCTION

For each strake, using the bottom positon along the frame and the molded dimension at each frame, we can determine the shape of the strake at each frame. Since the orientation of the strake is determined by the curvature of the frame, we can take the outside surface of each frame and cut it at the top and bottom points of the strake at that frame. These sections, cut from the frames, represents the shape of the inside surface of the strake at each frame. By connecting the sequential section together with polygons, we can create a strip of geometry which represents the inside molded face of the strake. This strip is then extruded outward by the amount of the strake's sided, or Y measurement of the cross section, to create the final strake model.

6.13.3. PLANKING IMPLEMENTATION

The resulting planking 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 *Molded #* 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 *Molded #* parameter will adjust the strake's molded dimension at each particular frame. By adjusting these two parameters for each frame, each strake can be designed with a custom shape, to the create the desired flow of the hull planking, similar to the methods employed by a shipwright. If the *Molded #* 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, 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. Then use the first strake as input into the new strake. The second strake will maintain the same molded value but will be automatically placed along top of the first strake. The molded 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 *Molded #* 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 (see fig. 35). 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.

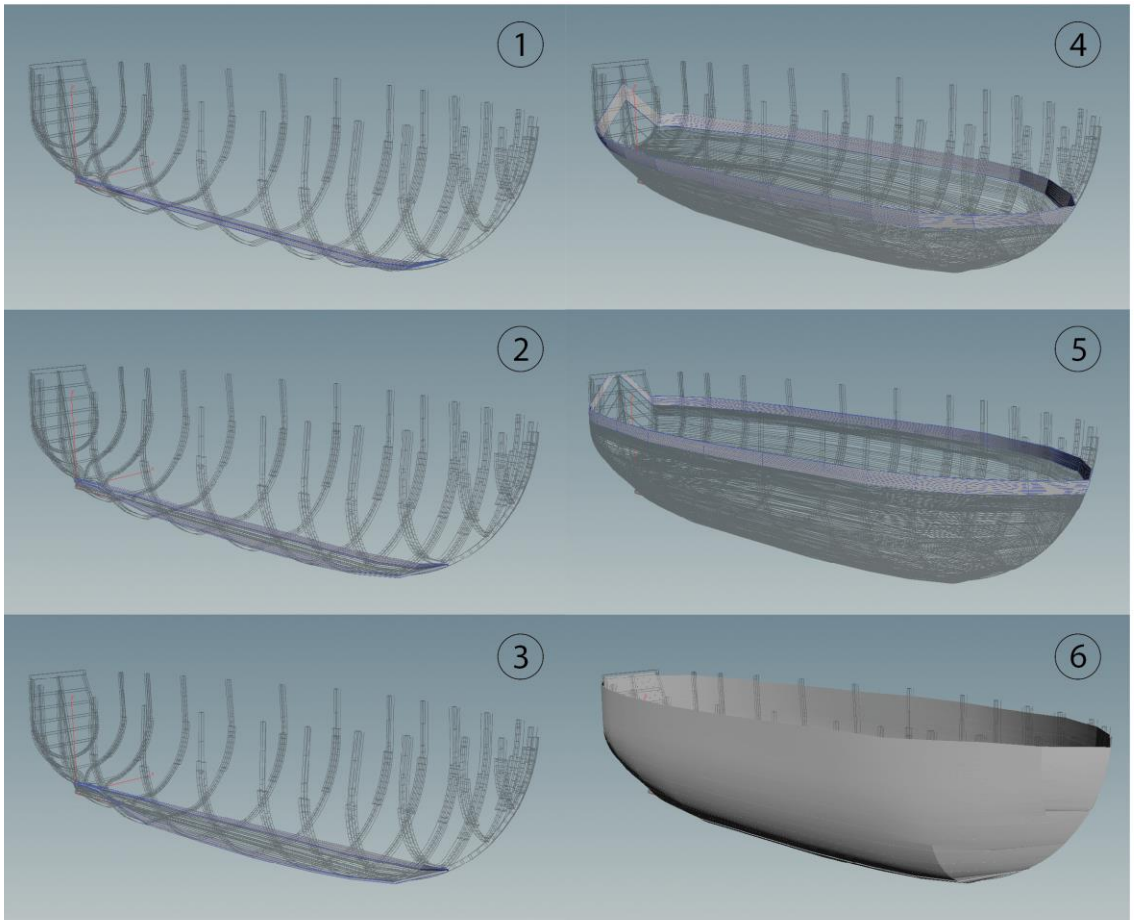

Figure 35. Planking process

VII. RESULTS AND CONCLUSIONS

I have developed procedural models that produce 3D models of hull timbers based on Oliveira's treatise. My models use parameter input using a GUI to manipulate the 3D models. The goal of this research was to demonstrate a procedural approach to creating computer-based 3D models of the lower hull of any ship within the bounds of Oliveira's recipe. To demonstrate the efficiency and effectiveness of the resulting procedural models and procedural approach, Dr. Castro provided archaeological data collected from a site known as *Belinho 1*, a small beach near Esposende, Portugal. This data was visualized using my procedural models.

7.1. RESULTS

Once the procedural models were created, using the graphic user interface to manipulate the models took very little time or effort. The entire process of visualizing Belinho 1 spanned the course of ten days. My correspondence with Dr. Castro during this period was almost exclusively through email, with the occasional video call. Our process was as follows; Dr. Castro would email me measurements to describe a component. I would enter them into the appropriate parameters of the procedural model, and then send screen captures showing the updated model from multiple views. The updates typically took only minutes to implement. Most of my time was spent collecting images from angles that sufficiently displayed the components in question. The Belinho

1 model included all components scoped for this project; keel, stem, stern post, keelson, wales, stringers, breast hooks, floor timbers, futtocks, stern panel timbers, deck timbers, and planking (see fig. 36).

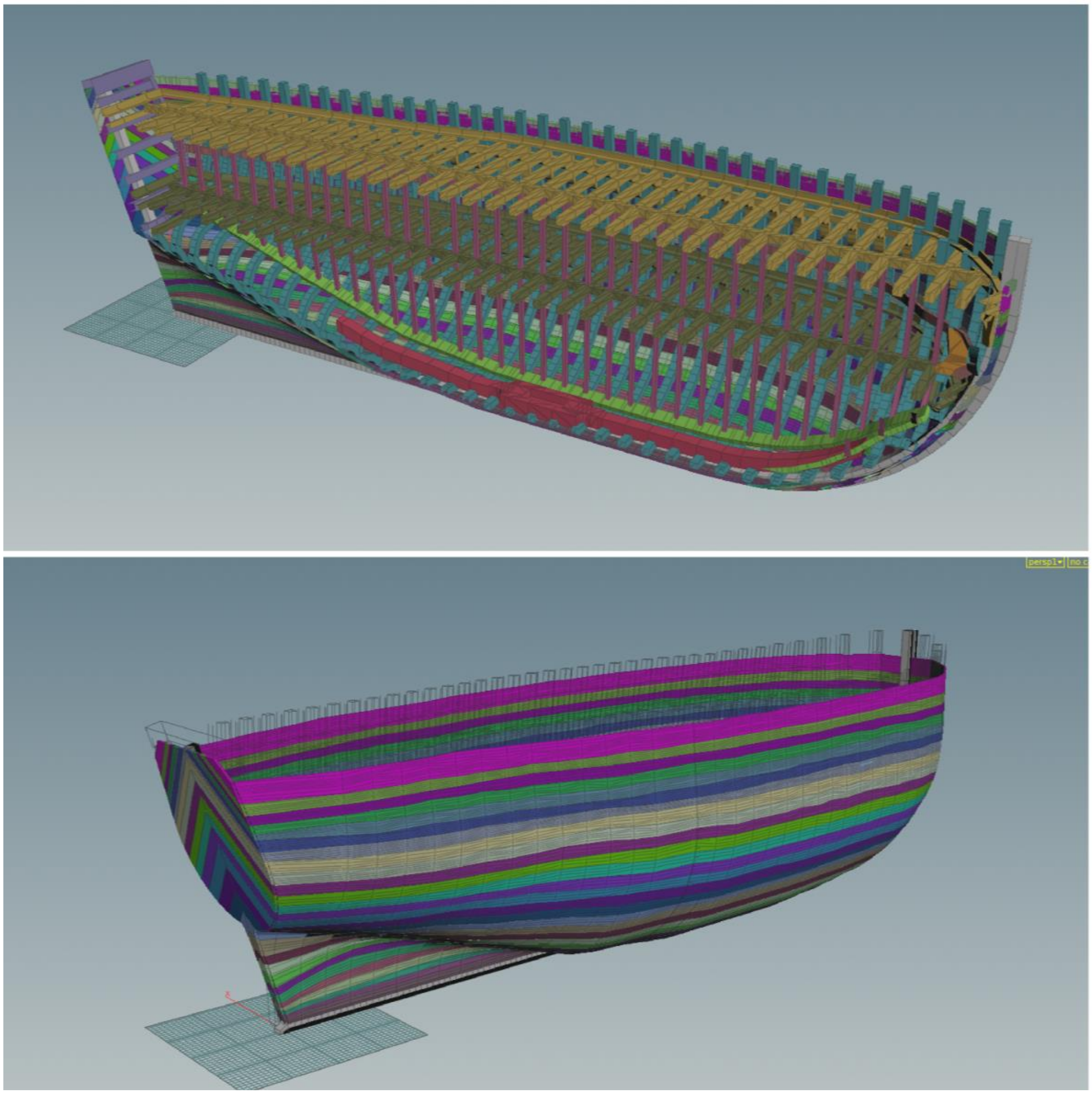

Figure 36. Belinho 1 model. Hull timbers (top), planking (bottom)

7.2. EVALUATION

Using procedural models provided great flexibility when integrating feedback from Dr. Castro. For example, during the construction of the Belinho 1 model, Dr. Castro requested that I add more curvature to the decks. Revision of the decks requires the remodeling of all of the deck timbers, for each of the decks. Using traditional modeling techniques, this would have taken many hours to implement. Using my procedural model, I was able to integrate the feedback within minutes. There were other similar instances during the modeling of Belinho 1 where the flexibility of the procedural models saved a substantial amount of time.

Procedural modeling however also exhibited specific problems. There were instances during the modeling of Belinho 1 where the procedural model failed. That is when a parameter or relationship caused the model to go into an unforeseen state which was not correctly handled by the algorithm. When these problems or oversights were identified, the algorithm needed to be updated to handle these cases. These problems were multiplied by the number of variables involved. This was particularly an issue for the deck knees where there are multiple interconnected components influencing the shape. A benefit of resolving these cases, is that when these are resolved for one component, or for one deck for instance, the solution can be propagated across all other instances of that component within the model. This feature of procedural modeling was very useful in the construction of the frames, where changes required updating over thirty components. The futtock feature of the frames model was implemented after other models had been built which were dependent on the frames, such as the longitudinal reinforcements and the decks. The ability to implement features, which can then propagate across multiple instances of an object, makes continual development on procedural models very efficient.

Procedural modeling can be technically challenging. In particular, developing a procedural workflow and parameterizing the process of planking a hull was the most challenging aspect of this project. It took several attempts and the exploration of several approaches before I came to the final implemented solution. The technique I implemented required technical knowledge in multiple areas, such as ship building, Python scripting, and advanced Houdini knowledge. This is a challenge unique to procedural modeling because in a typical modeling workflow, the artist is not required to be recreate or be concerned with the object's physical construction process, rather their concern is with recreating the object's form. Creating a procedural model however required additional knowledge of the physical construction process to successfully implement a modeling procedure that adequately parameterizes the planking process.

Another problem I experienced, which is not unique to procedural modeling, is the instability of Boolean operations in 3D modeling. In ship modeling there are several instances where it is necessary to calculate the intersection between two objects. Houdini currently does not have a Boolean operator which is consistently reliable. In the case of the deck knees, the clamp actually passes through each knee, however the hole in the knee for the clamp is not created in my model because of unstable results from Boolean operations. In a 'manual' modeling workflow, the artist would have the opportunity to clean up each Boolean operation. However, in a procedural workflow, if the operation does not produce correct results, any later steps in the procedure will fail.

A benefit of a procedural approach is the compactness and portability of procedural models. The Houdini digital assets have a small file size, typically less than one megabyte. The largest digital asset file for this project was 346 kilobytes. Models created with a digital asset can be stored by saving the parameter values rather than geometric data, which typically has a much larger file size. The compactness of procedural models facilitates portability and encourages the sharing of models.

A consideration during this project, was the question of whether or not to restrict user parameter value input within certain ranges. Constraining parameters within a range, prevents the user from causing the procedural algorithm to fail or create an implausible model. I decided to allow the user free range of each parameter because I felt that it was important to allow a broad sampling of the range of possibilities in order to encourage uninhibited exploration within the procedural model.

Based on my experience with this project, I believe that the flexibility provided by procedural modeling provides an overall time efficient solution for 3D modeling in nautical archaeology. The construction of a procedural model requires a significant investment in the design, construction, and trouble-shooting. However, this proved beneficial because after the procedural models were constructed, the iteration cycle was short enough that they could be used as tools for exploring alternative interpretations. This approach supported the collaborative nature of this project. It allowed Dr. Castro and I to optimize our meeting time by visualizing and discussing changes in real-time.

7.3. FUTURE WORK

There are several directions future work can be taken based on my procedural approach. Additional details of the timbers scoped for this project could be included in future procedural models, such as the addition of gun ports or small details such as nails and other fastening methods. The scope of modeling could be extended to other parts of the ship such as the upper decks or the ship's rigging. A model could be constructed using a different component taxonomy. Procedural models could be constructed which encompass the rules and grammar of other shipbuilding traditions, such as Chinese junks or Japanese war ships. A similar procedural workflow could be implemented in a solid modeling software, this would allow the researcher to physically simulate and test the resulting models. The models can be exported as a 3D printable mesh from Houdini and then 3D printed for physical examination and exploration.

This project did not include a user study, however one could be designed 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".

The Houdini digital assets created for this project could be integrated into a game engine, such as Unity or Unreal, using the Houdini Engine plugin. This is designed to allow for the manipulation of Houdini digital assets from within the game engine editor of either Unity or Unreal. This would create a widely accessible ship building utility deployed using Unity or Unreal engine. These are both free to download and use.

Using procedural models to explore research hypotheses is also applicable to a wide array of applications where interactive visualization of complex geometric construction data is desired, such as in mechanical engineering. By implementing a similar approach to constructing computer-based 3D models, future researchers may be able to use procedural 3D models as a tool to test and inform new research findings.

WORKS CITED

Antscherl, David. "A Primer on Planking" Admiraltymodels.com, n.d. Web. 15 Aug. 2016. http://www.admiraltymodels.com/Planking_primer.pdf

Cook, Charles. *A Parametric Model of the Portuguese Nau*. Thesis. Texas A&M University, College Station, TX, 2012.

Doran, Jim, and Frank Roy Hodson. *Mathematics and Computers in Archaeology*. Cambridge, MA: Harvard UP, 1975.

Evans, Thomas Laurence. *Digital Archaeology: Bridging Method and Theory*. Psychology Press, 2006.

Fleming, Andrew, and David L. Clarke. "Models in Archaeology." American Anthropological Association. *Man* 8.2 (1973): 315.

Haegler, Simon, Pascal Müller, and Luc Van Gool. "Procedural Modeling for Digital Cultural Heritage." *EURASIP Journal on Image and Video Processing* 2009 (2009): 1-11.

Hazlett, Alexander Dean. *The Nau of the Livro Nautico: Reconstructing a Sixteenth-century Indiaman from Texts*. Thesis. Texas A&M University, College Station, TX, 2007.

Huang, Chun-Yen, and Wen-Kai Tai. "Ting Tools: Interactive and Procedural Modeling of Chinese Ting." *The Visual Computer* 29.12 (2012): 1303-318.

Lock, Gary. *Using Computers in Archaeology: Towards Virtual Pasts*. London: Routledge, 2003.

Miller, Paul and Richards, Julian. "The Good, the Bad, and the Downright Misleading: Archeological Adoption of Computer Visualization." *Computer Applications and Quantitative Methods in Archaeology* (1994): 19-22.

Muller, Pascal, Tijl Vereenooghe, Peter Wonka, Iken Paap, and Luc Van Gool. *Procedural 3D Reconstruction of Puuc Buildings in Xkipché*. Thesis. Arizona State University, 2006.

"Products | SideFX." SideFX.com, n.d. Web. 23 June 2016. https://www.sidefx.com/filmtv/products/

Richards, Julian. "Computer Usage in British Archaeology." IFA Occasional Paper No. 1, Birmingham: Institute of Field Archaeologists, 1986.

Saldana, Marie. *An Integrated Approach to the Procedural Modeling of Ancient Cities and Buildings*. Thesis. University of Los Angeles, Los Angeles, CA, 2015.

Wells, Audrey Elizabeth. *Virtual Reconstruction of a Seventeenth-century Portuguese Nau*. Thesis. Texas A&M University, College Station, TX, 2008.

APPENDIX

This appendix contains a description of each file that was used to complete this thesis and is available for download along with the thesis document via the OAK Trust Texas A&M Open Access Digital Repository at: oaktrust.library.tamu.edu/handle/1969.1/3367.

The file name is underlined and followed by its description below. Any questions are welcome @ Mat.suarez29@gmail.com .

README.txt: A text file that gives instructions on the setup and use of the Nau.hiplc and all Houdini Digital Asset (.hdalc) files used for this project. For more information on each of the following files, please reference the README.txt.

Nau.hiplc: A Houdini file of the Belinho 1 model. The implementation of each of the Houdini digital asset files listed below is demonstrated by utilizing them in the construction of the Belinho 1 model.

Keel.hdalc: A Houdini digital asset which creates the procedural keel model.

Frame.hdalc: A Houdini digital asset which creates the procedural frame model.

Stern_panel.hdalc: A Houdini digital asset which creates the procedural stern panel model.

Deck.hdalc: A Houdini digital asset which creates the procedural deck models.

Longitudinal_reinforcement.hdalc: A Houdini digital asset which creates the procedural wale and stringer models.

Breasthook.hdalc: A Houdini digital asset which creates a procedural breasthook model.

Stanchion.hdalc: A Houdini digital asset which creates the procedural stanchion model.

Keelson.hdalc: A Houdini digital asset which creates the procedural keelson model.

Planking.hdalc: A Houdini digital asset which can be used to create a procedural planking model.