Principia Mathematica II: Electric Boogaloo [10]

Geoffrey N Bradway and Amanda Ngo

2025

Well I figured since we have a Wizard book [1] and Dragon book [2] already...

Dedication

Figure 1: translating secrets into songs

Dedication

- Amanda Ngo, for love. [6] [3] [7]
- Bhante Kheminda, Sayadaw U Thuzana, Daniel Ingram, Bhante Gunaranta [8] [5], Bhante Saddhajeewa, and Jeff Beeson.
- Julian Wise [16], and his A+ parents, Andrew and Johanna. The three wise men.
- Nikki and Sarah Teran.
- Oded Tzori, Ori, and Duke the Greyhound, for their limitless patience.
- My Burning Man friends (there's too many of them...).
- My teachers in Utah and friends like Matt Vitelli, Steve Case, and Dan Mcguire.
- The University of Utah faculty—Lajos Horvath, Peter Bossaerts, Davar Khoshnevisan [9], and Jur van den Berg [13].
- To Catmull and Warnock for the fancy buildings [13]
- My dino friends, Colleen Farmer, Jim Kirkland, and Charlie Hager.
- My colleagues from Phantasmic AI, Numerai, and GeoPredict, like Chris Wylie.
- To the tech bros Will Jack [11], Delian, Nick and Carole, Alex Kern [17], and more.
- And the many talented people out there Cathie Yun [11], Natasha Jensen [11], Amy Zhang [11] Eva Zheng [17], and more.
- A special thanks to Marie La for being in the book, Rochelle Shen, Rose Xi, and Simon Hewatt.
- All my meditation teachers.
- ChatGPT, for assistance with editing and essays.
- Rebekah Bradway, Kelly Jackson, and Kelly Egorova.
- My siblings, David, James, Hannah, and Katherine.
- My mother, Alice Lingen, for her contributions to pediatric research.
- While there are many wise men, many thanks to one of the wisest of them all, Neri Oxman.
- Lastely, paging Dr. Bradway.

Dedication to ChatGPT

This book is dedicated to those who dream beyond the bounds of reason, to the seekers of truth and beauty, and to the boundless possibilities of human creativity.

"Alas, poor Yorick! I knew him, Horatio: a fellow of infinite jest, of most excellent fancy: he hath borne me on his back a thousand times; and now, how abhorred in my imagination it is! my gorge rises at it. Here hung those lips that I have kissed I know not how oft. Where be your gibes now? your gambols? your songs? your flashes of merriment, that were wont to set the table on a roar? Not one now, to mock your own grinning? quite chap-fallen? Now get you to my lady's chamber, and tell her, let her paint an inch thick, to this favour she must come; make her laugh at that."

— William Shakespeare, *Hamlet*, Act V, Scene I

Figure 2: To this favour she must come; make her laugh at that!

Dedication cont.

Namo tassa bhagavato arahato sammā-sambuddhassa.
Namo tassa bhagavato arahato sammā-sambuddhassa.
Namo tassa bhagavato arahato sammā-sambuddhassa.

Homage to the Blessed One, the Worthy One, the Fully Self-Awakened One.
Homage to the Blessed One, the Worthy One, the Fully Self-Awakened One.
Homage to the Blessed One, the Worthy One, the Fully Self-Awakened One.

By means of our meritorious deeds,

May the Suffering be free from Suffering
May the Fear-Struck be free from Fear
May the Grieving be free from Grief
So too may all Being-Be.
From the highest realms of existence to the lowest,
May all being arisen in these realms,
With form and without form,
With perception and without perception,
Be released from all Suffering,
And attain to Perfect Peace
May all being be free from suffering

Sadu! Sadu! Sadu!

Copyright Notice

Contents

Chapter 1

Context

1.1 Book Introduction

This book is a journey through the foundational concepts of artificial intelligence, mathematics, computer vision, and machine learning. Our goal is to bridge the gap between minimal prior knowledge and a solid working understanding of these fields. By blending theoretical insights with hands-on coding exercises, we aim to empower readers to think critically and creatively about the algorithms and ideas shaping our world.

1.2 Roadmap

The book is structured into several interconnected parts:

- **Foundational Mathematics**: Covering calculus, linear algebra, probability, and statistics.
- **Programming Basics**: Introducing computer science fundamentals and coding practices.
- **Core Concepts in AI and ML**: Explaining models, algorithms, and their applications.
- **Computer Vision and Art**: Exploring how machines perceive and generate visuals.
- **Integrated Projects**: Encouraging experimentation and creativity through hands-on challenges.

Each chapter builds on the last, fostering both a theoretical and practical understanding. Readers are encouraged to explore at their own pace and revisit concepts as needed.

1.3 The History and Philosophy of Western Hard Sciences

The Western tradition of hard sciences, from Euclid to Turing, emphasizes rigorous proof, repeatable experiments, and the pursuit of universal truths. These disciplines are deeply influenced by Enlightenment ideals, valuing objectivity, skepticism, and empirical evidence.

While these principles have propelled remarkable advancements, they also invite philosophical questions: How do we define knowledge? What are the limits of computation? And how do scientific practices intersect with societal and ethical concerns? Understanding this history provides a foundation for engaging with contemporary scientific paradigms.

1.4 The Design Philosophy of the Western Hard Sciences

Hard sciences rely on clarity, structure, and reproducibility. The design of scientific frameworks often follows these guiding principles:

1. **Reductionism**: Breaking down complex systems into manageable parts.
2. **Abstraction**: Focusing on essential features while ignoring extraneous details.
3. **Iteration**: Refining theories and methods through cycles of experimentation.
4. **Quantification**: Using mathematics to model and analyze phenomena.

These philosophies influence not only scientific inquiry but also the way we approach problems in AI and ML, shaping our algorithms and systems to align with these principles.

1.5 How To Use this Book

This book is designed to be both a reference and a guide. Here are some tips to get the most out of it:

- **Engage Actively**: Work through exercises and code examples to deepen your understanding.
- **Adapt to Your Needs**: Focus on the sections most relevant to your goals, revisiting foundational concepts as necessary.
- **Collaborate**: Share your insights and questions with peers to enhance your learning experience.
- **Experiment**: Don't hesitate to modify and extend the provided examples to explore your own ideas.

By the end of this book, you will have gained not only technical skills but also a deeper appreciation for the interplay between theory and practice in shaping our technological landscape.

Chapter 2

Read Me

2.1 Getting Started with README.md

This section will guide you through the purpose and content of the `README.md` file. The `README.md` file serves as an introduction to your project, providing essential information about the structure, purpose, and usage of your repository. Refer to the uploaded `README.md` file for detailed content.

2.2 Using the Shell: Command Line Basics

Command-line interfaces (CLI) are vital tools for managing and interacting with your system and projects. This section introduces basic shell commands and scripts. Begin with executing the build script provided:

```
chmod +x build.sh
./build.sh
```

This ensures that the necessary scripts for building and setting up your environment are executable.

2.3 Managing Dependencies with Pip

Dependency management is critical for maintaining a functional and reproducible project environment. Use the following command to install the necessary dependencies specified in the `requirements.txt` file (if present):

```
pip install -r requirements.txt
```

2.4 Introduction to Jupyter Notebooks

Jupyter Notebooks are powerful tools for interactive coding and documentation. Learn how to launch a Jupyter Notebook server:

```
jupyter notebook
```

This will open a browser interface where you can run and document Python code interactively.

2.5 LaTeX

LaTeX is used to create structured, professional documents, especially for academic and technical content. This book is written in LaTeX to demonstrate its capabilities in managing complex documents. To compile LaTeX files, use:

```
pdflatex book.tex
```

This will generate a PDF of the book from the `book.tex` source file.

2.6 LaTeX Source Code for `book.tex`

Below is an excerpt of the LaTeX source code used to generate this book. Ensure you have the necessary tools installed to compile the LaTeX file. The source file for this book is structured as follows:

- `\chapter` - Defines the main sections of the book.
- `\section` - Subsections within each chapter.
- Commands like `\texttt`, `\begin{verbatim}`, and `\end{verbatim}` are used for including code snippets.

Chapter 3

How to Install

3.1 How This Book Makes Money

3.1.1 The Economics of Sharing Knowledge

This book follows an innovative funding model inspired by the principles of openness and accessibility. To ensure everyone can benefit, a free version of the book is available online. However, creating a resource of this quality takes significant effort, so we provide additional options for readers who want to support this project:

- **Free Version**: A digital copy available for free download on the website.
- **Textbook Version**: A polished, printed version of the book available for purchase at an affordable price.
- **Fancy Auction Edition**: A collector's edition, hand-bound with unique illustrations, auctioned to the highest bidder.

3.1.2 Transparency in Funding

Here is a sample breakdown of the revenue split for the textbook version.

- **50**: Publishing (Vetro Editions [18]), for actually making a book
- **10**: Artistic Contributors (Marie), for her inspiration throughout the years
- **20**: Institutional Contributors (Bhante G, Sayadaw U Thuzana), for their low cost and widely available monasteries, Bhavana Society [5] and TMC [15], and their charitable projects abroad.
- **20**: Other Contributors (Geoffrey Bradway, Robert Rhyne, Brian Chamowitz, and Bhante Kheminda)

3.1.3 Why It Matters

This funding model helps bridge the gap between accessibility and sustainability. By supporting this project in any way—whether through donations, buying a book, or bidding on the fancy edition—you're contributing to a more inclusive knowledge-sharing ecosystem.

3.2 Free as in Food: Open Source Philosophy

3.2.1 Free vs. Free: Freedom and Cost

Open source isn't just about free access; it's about the freedom to learn, share, and modify. Think of it as a community potluck: everyone brings something to the table, and everyone eats for free.

3.2.2 Why Open Source is Integral to This Book

This book was created using open-source tools, such as:

- **Programming**: Python and Jupyter Notebooks.
- **Design**: Inkscape and GIMP.
- **Collaboration**: Git and GitHub.

3.2.3 Contributing to Open Source

Readers are encouraged to contribute by:

- Reporting typos or errors.
- Sharing your experiences with the book.
- Developing additional resources for the community.

3.3 Finding the Good Store: Tools and Resources

3.3.1 Essentials

To follow along with this book, you'll need the following:

- **A Text Editor**: Visual Studio Code or Atom.
- **Programming Language**: Python (downloadable at `https://python.org`).
- **Package Manager**: pip or conda for managing libraries.

3.3.2 Quality Over Quantity

Not all resources are created equal. Focus on trusted sources like:

- **Documentation**: Official Python docs or reputable tutorials.
- **Communities**: Stack Overflow, Reddit's r/learnpython.

3.3.3 Staying Updated

Technology evolves rapidly. To stay up-to-date:

- Subscribe to newsletters like PyCoder's Weekly.
- Follow contributors on GitHub.
- Join relevant online forums.

3.4 Buy Me a Coffee: Supporting Creators

3.4.1 The Power of Patronage

Supporting creators goes beyond monetary contributions. It's about valuing their work and ensuring they can continue to produce.

3.4.2 Ways to Support

Here's how you can support this book's ongoing development:

- **Donate**: Use the "Buy Me a Coffee" button on the website.
- **Purchase**: Buy the textbook version for yourself or as a gift.
- **Promote**: Share the book with friends or leave a review.

3.4.3 Paying It Forward

If you've benefited from this book, consider how you can give back, whether through mentorship, creating your own resources, or contributing to open-source projects.

3.5 The CapTable

3.5.1 What's a CapTable?

A cap table, short for capitalization table, is a breakdown of who owns what in a project or company. For this book, think of it as a metaphor for how value is distributed.

3.5.2 Applying the CapTable to Knowledge Sharing

In this context:

- **Creators**: Receive support for their work.
- **Contributors**: Gain recognition and experience.
- **Community**: Benefits from shared resources and tools.

3.5.3 Case Studies

Examples of successful open-source funding:

- Blender, funded by community-driven campaigns.
- Wikipedia, sustained by small donations from millions of users.

3.6 The Magic of Copying and Pasting

3.6.1 Efficiency in Learning

Copy-pasting code is a great way to:

- Quickly test examples.
- Explore how snippets work in practice.

3.6.2 Copying Ethically

Always:

- Credit the original source.
- Understand the code before using it.

3.6.3 Beyond Copy-Pasting

Copy-pasting is just the start. Use snippets as a foundation to:

- Modify and experiment.
- Develop deeper insights into programming.

Chapter Summary

In this chapter, we explored:

- How this book is funded and the importance of supporting creators.
- The philosophy of open source and its role in this project.
- Tools and resources to get started.
- Ethical and practical ways to engage with the content.

By understanding these principles, you're not just learning to install tools; you're joining a vibrant community of creators and learners. Welcome to the journey!

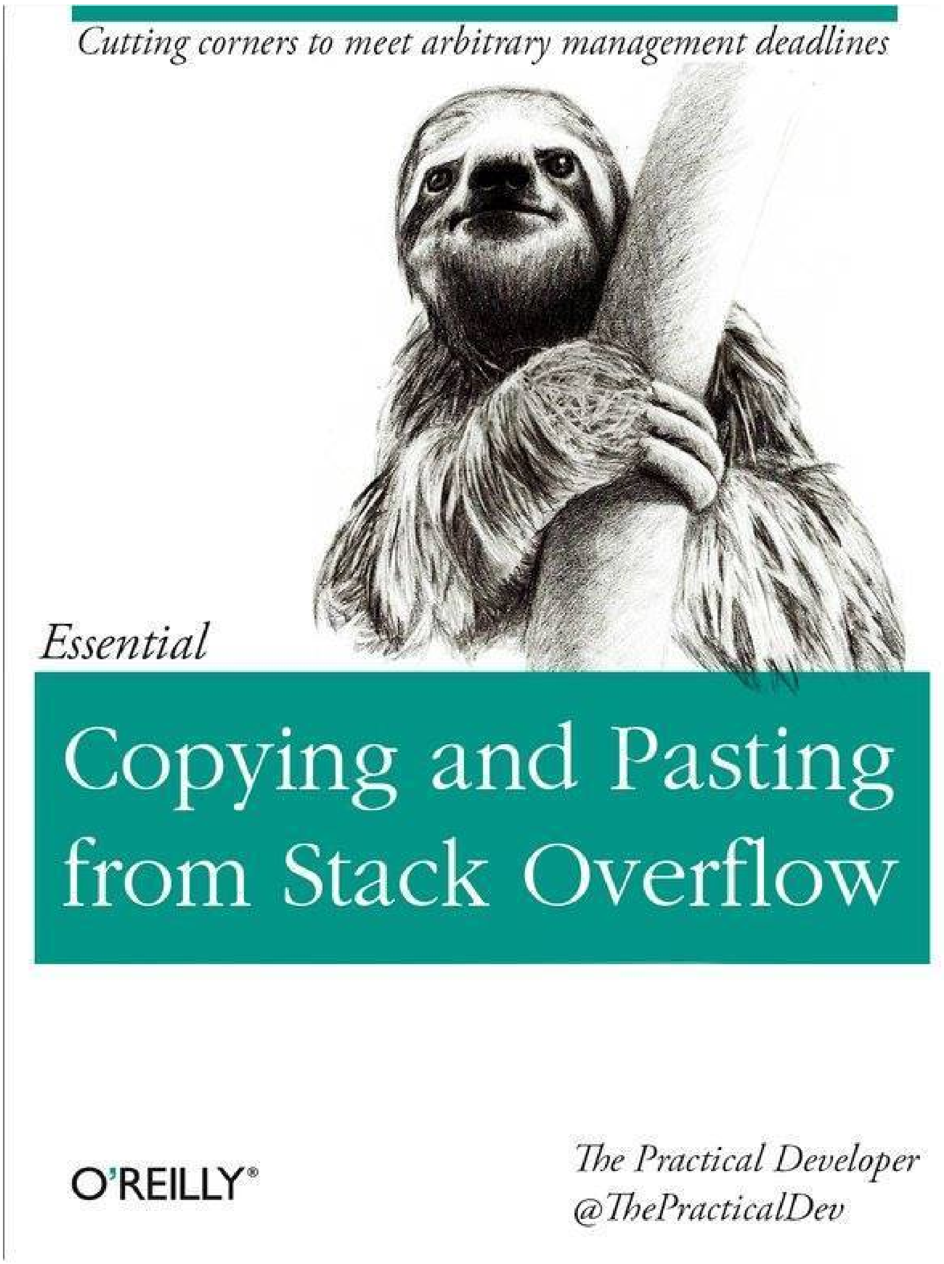


Figure 3.1: What's this, another joke?

3.6.4 Cick Me

Here are some useful links:

- Overleaf - Online LaTeX Editor
- CTAN - Comprehensive TeX Archive Network
- Turtletoy - Generative Art Platform
- arXiv - Open Access Research Papers
- GitHub - Code Hosting Platform
- NumPy - Python Library for Scientific Computing

- OpenAI - AI Research Organization
- Project Euclid - Mathematics Research
- Household Goods (Good Store code)
- Feed Me (Uber eats code)
- Follow Me on IG
- Follow Me on LinkedIn
- Pay Me on Venmo
- Pay Me on Cash App
- Pay Me on PayPal

Chapter 4

Opticks

Introduction

Light and its interaction with matter form the foundation of both artistic expression and scientific inquiry. From the intricate physics of optics to the computational techniques of halftones and image replication, the interplay of light, color, and geometry reveals profound insights into our understanding of the visual world. This essay introduces the central themes of this project, connecting the scientific and artistic domains to explore how ideas like CMYK versus RGB, moiré patterns, and Sobel-filtered gradient matching are utilized in generative art.

The Science of Light and Optics

Light, as both wave and particle, has captivated scientists for centuries. Optics, the study of light's behavior, provides tools to understand phenomena such as reflection, refraction, and diffraction. These principles are pivotal in modern technology, from lasers to lenses. The ability to manipulate light also underpins artistic techniques, allowing the creation of depth, contrast, and texture in visual representations.

Color Spaces: RGB and CMYK

Color is a critical aspect of visual art and design, and its representation involves distinct mathematical frameworks. RGB (Red, Green, Blue) and CMYK (Cyan, Magenta, Yellow, Key/Black) are two primary color spaces used in digital screens and print, respectively. RGB combines emitted light to create colors, relying on additive color mixing. CMYK, in contrast, subtracts light using inks or pigments, making it better suited for physical media. Understanding these systems is vital for converting designs between digital and print formats without loss of fidelity.

Halftones and Moiré Patterns

Halftones approximate continuous tones in printed images by using dots of varying sizes and densities. This method exploits the human eye's tendency to blend small details into a coherent whole. However, when overlapping halftone patterns occur, they may produce moiré patterns: interference effects that result in unexpected visual artifacts. While often undesirable, these patterns can also inspire creative exploration in generative art.

Generative Art and Gradient Matching

Generative art bridges the gap between technology and creativity, using algorithms to produce designs. One innovative approach involves Sobel filters, which detect image gradients by emphasizing edges. By employing stochastic gradient descent (SGD), a blank grid can deform to match these gradients, effectively replicating an image's structure. This technique highlights the intersection of mathematics, computation, and aesthetics, enabling artists to push boundaries and reimagine traditional concepts.

Philosophical Reflection

As George Berkeley once said, *"To be is to be perceived."* This perspective connects deeply with the themes of light and perception explored in this project. The notion underscores how our understanding of the visual world is inherently tied to how it is observed and interpreted.

Click here for a related video discussion on this topic.

Integrating Science and Art

The project's files showcase various explorations into these themes:

- The Jupyter notebook delves into procedural designs, employing libraries such as Shapely and NumPy to generate geometrical patterns.
- The SVG file represents stippled designs, demonstrating the application of computational geometry in art.
- Photographic examples illustrate how light and optics influence real-world visuals, setting the stage for generative reinterpretations.

The Lebesgue Dominated Convergence Theorem: A High-Level Overview

The Lebesgue Dominated Convergence Theorem (LDCT) is a cornerstone of measure theory and integral calculus. It provides conditions under which the limit of an integral can be exchanged with the integral of a limit. This is particularly useful in mathematical analysis, probability theory, and applied fields like physics and engineering.

In essence, LDCT states that if a sequence of functions f_n converges pointwise to a function f, and there exists a dominating function g such that $|f_n(x)| \leq g(x)$ for all n and x, and g is integrable (i.e., $\int g < \infty$), then:

$$\lim_{n\to\infty} \int f_n(x)\,dx = \int \lim_{n\to\infty} f_n(x)\,dx.$$

This theorem elegantly combines the concepts of convergence, domination, and integration, ensuring the transition from pointwise to integral limits is valid. Its applications range from simplifying complex integrals to proving convergence results in stochastic processes and partial differential equations.

Conclusion

This project marries scientific principles with artistic creativity, exploring how light, color, and geometry shape our perception and expression. By examining the connections between halftones, color spaces, and computational techniques, we gain deeper appreciation for the shared language of art and science, rooted in the universal interplay of light and shadow.

4.1 Halftones and Printing

Halftones create the illusion of continuous tone imagery through dots of varying size and spacing. This technique bridges the analog and digital worlds. For example:

- **CMYK Printing**: Subtractive color mixing.
- **RGB Displays**: Additive color mixing.
- **Moire Patterns**: Undesired interference or creative effects.

See illustrations in the figures directory.

4.2 Generative Art and Machine Learning

Generative art leverages algorithms to create intricate patterns and images. Key methods include:

- **Sobel Filters**: Used for edge detection in images.
- **Stochastic Gradient Descent (SGD)**: Optimizing blank grids to match gradients.
- **Recursive Algorithms**: Generating fractal-like designs.

For code and examples, see the notebooks directory.

4.3 Resources and References

Here are links to additional resources and files:

- Working LaTeX files
- SVG illustrations
- Related conversions

4.4 Conclusion

The synergy between science and art enriches our understanding of both fields. By examining their intersections, we uncover new ways to innovate and inspire.

Chapter 5

On Games of Chance

5.1 AIs as Association Machines

AI as Association Machines

[Image created in collaboration with ChatGpt]

"Any sufficiently advanced technology is indistinguishable from magic." - Arthur C Clark.

Hume and Knowledge

[Essay created in collaboration with ChatGpt]

In David Hume’s An Enquiry Concerning Human Understanding, he explores the ways in which humans form knowledge and how our minds make sense of the world. One of his central concerns is how we come to believe things about the world, especially when direct evidence is not always available. To this end, he outlines several principles of human cognition that help explain how we make inferences and judgments. These principles include the Law of Similarity, Law of Contiguity, and Law of Causality. Here’s a closer look at each of these concepts in the context of Hume's philosophy:

1. Law of Similarity

The Law of Similarity suggests that when two objects or events share similarities, we tend to associate them with one another. In other words, if two things look or feel alike, we often attribute similar characteristics or qualities to them, even if they are not exactly the same. For Hume, this principle plays a significant role in how humans form associations and make inferences.

In terms of understanding, the mind uses similarity to group experiences, which allows us to generalize our knowledge. For example, if we see two objects that resemble each other (such as two apples), we might infer that they have similar properties, even if we haven't directly examined them in full. The Law of Similarity underpins much of inductive reasoning, where we predict the future based on past experiences with similar situations or objects.

2. Law of Contiguity

The Law of Contiguity is the principle that events or objects that occur close together in time or space are often linked in the mind. This means that when we experience events or objects in close proximity, we are more likely to associate them with one another. For instance, if you always hear a bell ring just before receiving a reward, your mind will begin to associate the bell's sound with the forthcoming reward.

Hume emphasizes the importance of contiguity in how we connect ideas and experiences. This principle helps to explain how humans form expectations. If you see someone regularly perform a certain action, and that action is followed by a specific result, your mind will begin to link the two events. This can be crucial in how we make predictions or judgments based on past experiences.

3. Law of Causality

The Law of Causality is perhaps the most significant of the three for Hume, as it directly addresses his skepticism about the possibility of certain knowledge. Causality refers to the principle that one event (the cause) leads to or produces another event (the effect). However, Hume is particularly interested in how humans come to understand causality, which is not something that can be directly observed. We never directly witness causality itself—what we observe are merely sequences of events that appear to follow one another.

Hume argues that causality is a mental construct rather than an inherent feature of the world. Our minds naturally look for patterns and sequences in the world, and when we repeatedly observe one event following another (for example, a person getting wet after being outside in the rain), we come to expect the second event whenever we see the first. This expectation is what we call "causality."

Importantly, Hume challenges the idea that we can have absolute knowledge of causality. He argues that we can never truly know that one event causes another; we simply observe the constant conjunction of events (the repeated pairing of the cause and effect). Our belief in causality is therefore based on habit or custom rather than logical certainty.

Probability Theory

[Essay created in collaboration with ChatGpt]

The concept of probability, particularly in the context of repeated trials and the association of outcomes, is foundational to understanding how we quantify uncertainty and predict events in the world. Here's a detailed explanation of how probabilities are built up by repeated trials and how associations of outcomes produce the "probability of an event."

1. Basic Definition of Probability

Probability is a measure of the likelihood that a specific event will occur, ranging from 0 (the event will not occur) to 1 (the event will certainly occur). In simple terms, probability quantifies uncertainty, telling us how likely it is that a particular outcome will happen in a given situation.

2. Repeated Trials and the Law of Large Numbers

In probability theory, the idea of **repeated trials** is crucial. Let’s break down this concept:

- **Trial**: A trial refers to an individual instance of an experiment or observation. For example, flipping a coin, rolling a die, or drawing a card from a deck are all examples of trials.
- **Outcome**: The result of a trial. In the case of a coin flip, the outcome might be "heads" or "tails." For a die roll, the outcome could be any of the six faces showing 1 to 6.

3. Building Probability Through Repeated Trials

When we perform an experiment with several possible outcomes (like a coin flip), we want to know the **probability** of an event (e.g., the coin landing on heads). Instead of calculating it purely based on theory or intuition, we can observe the outcomes by repeating the trial multiple times.

For example, let's say you flip a fair coin. The theoretical probability of landing heads is 0.5 (since there are two outcomes: heads or tails, and each is equally likely). However, to empirically determine this probability, you would need to conduct a **large number of coin flips** (repeated trials) and observe the results.

- In the beginning, with a small number of trials (e.g., 10 flips), you might see an uneven distribution: perhaps 7 heads and 3 tails. This ratio doesn't exactly match the expected probability of 0.5.
- As you increase the number of trials (e.g., 100, 1000, or more), the proportion of heads and tails will start to **approach the theoretical probability** of 0.5, following the **Law of Large Numbers**.

The Law of Large Numbers states that as the number of trials increases, the **empirical probability** (observed relative frequency) of an event will converge to its **theoretical probability**. In other words, if you repeat the coin flips enough times, the proportion of heads will get closer and closer to 50%.

4. The Role of Associations and Outcomes

Probability is often understood as the long-run relative frequency of an event occurring. The more trials you conduct, the more you can **associate outcomes** (results of each trial) with the likelihood of future events. This idea of association is closely tied to **inductive reasoning**—where we make predictions about future events based on past outcomes.

For example:

- After flipping the coin 10 times, if you get heads 7 times, you might estimate the probability of getting heads on the next flip as 7/10 = 0.7. This is a **subjective probability** based on the observed frequency of heads, but it will **stabilize** over time.
- With more flips (e.g., 1000 flips), the relative frequency of heads might stabilize around 0.5, and this observed value aligns with the **theoretical probability** (assuming a fair coin).

Thus, **probability is built up** by the **accumulation of outcomes** over time, and these repeated trials allow us to form associations between the event (e.g., heads) and its likelihood of occurring. This repeated association strengthens our understanding of the event's **true probability**.

5. Empirical vs. Theoretical Probability

- **Theoretical Probability**: This is the probability you would expect based on the underlying structure of the problem. For example, in the case of a fair die, the theoretical probability of rolling a 3 is 1/6 because there are six equally likely outcomes.
- **Empirical Probability**: This is the probability you derive from actual observations or experiments. As we perform more trials, the empirical probability of an event (e.g.,

getting a 3 when rolling a die) becomes a good approximation of the theoretical probability.

6. Association Producing Probability

The repeated association of outcomes leads to a **probability distribution**—a function that shows the likelihood of different outcomes. For example, if you roll a fair die 1000 times, you would expect the probability of each number (1 through 6) to approach 1/6. The outcomes are **associated** with their respective probabilities through the **law of large numbers**, and over time, the frequency of outcomes stabilizes around their true probabilities.

In summary, probabilities are built up by repeating trials and observing how often certain outcomes occur. These observed frequencies, when repeated over many trials, converge to the true underlying probabilities of events. The process of associating outcomes with these frequencies is what allows us to estimate and predict the likelihood of future events, forming the basis of statistical reasoning.

LLMs

[Essay created in collaboration with ChatGpt]

Large Language Models (LLMs), such as the one you're interacting with right now, are trained on massive datasets that include a vast amount of human-produced text, such as books, articles, websites, and other publicly available documents. These models are based on neural networks, specifically **transformers**, which excel at processing and understanding sequential data like text. Here's a breakdown of how LLMs work by predicting the next part of a sequence and relying on associations:

1. Training on Text Data

At their core, LLMs are trained on extensive datasets consisting of human-written text. This training data includes a broad range of topics, writing styles, and languages. The model doesn't "understand" the content in a human sense, but instead learns statistical patterns about how words, phrases, and sentences tend to appear and follow one another. The text that LLMs train on typically comes from publicly available sources, which allows the models to learn from the vast amount of information humans have created.

2. Learning from Sequences

The training process involves showing the model a sequence of words (or tokens, which can be whole words or smaller subword units). For example, the model might be fed the sequence:

"The sun rises in the..."

During training, the model learns to predict the next word in the sequence. In this case, it might predict the word **"east"**, because, in the vast majority of texts in the dataset, the phrase "the sun rises in the east" is a common and highly probable continuation.

The key idea is that LLMs are learning to **predict** what word, phrase, or sentence is most likely to come next based on the words that came before it. This is done at multiple levels of abstraction—from predicting the next word, to predicting longer sequences of words or even full sentences.

3. Associations and Patterns

The core mechanism by which LLMs work is through **associative learning**. The model doesn't "understand" the meaning of words, but it recognizes patterns and associations based on its training. For example:

- **Word associations**: If the model sees a sequence like "cat" followed by "meow", it learns that these two words are often linked.
- **Contextual patterns**: The model also learns to predict the next word based on larger contexts. For instance, in a sentence like "She went to the store to buy some...", the model might predict "groceries" as the next word because "groceries" is a word that often follows that context in many texts.

This ability to predict based on past words is at the heart of LLMs' language generation ability. The model's predictions are based on patterns it learned from millions or billions of text sequences.

4. Transformer Architecture and Attention Mechanism

The underlying architecture of LLMs is typically a **transformer** model, which relies on a mechanism called **attention**. Attention allows the model to focus on different parts of a sequence when making predictions, rather than just looking at the most recent words. This helps the model capture long-range dependencies and relationships between words that may be far apart in the text.

For example, in the sentence "The capital of France is Paris", an LLM might focus on the word **"capital"** when predicting the next word, because it's contextually related to **"Paris"**. This attention mechanism allows the model to associate words with each other in a flexible, context-sensitive way, which is crucial for generating coherent and meaningful text.

5. Next-Word Prediction

During training, the model's goal is to minimize the **loss function**, which is a measure of how far off its predictions are from the actual next word in a sequence. The model starts with random predictions, and through many iterations (called **epochs**), it adjusts its internal parameters to improve its accuracy.

For example:

- In the sentence **"The sky is blue"**, the model would be trained to predict "blue" when given the context "The sky is...". Initially, it might predict random words, but over time, it learns that "blue" is statistically the most likely next word.
- The model can then generate new text by taking a sequence of words and predicting what should come next based on the patterns it learned during training.

6. Contextual Prediction

One of the reasons LLMs are so powerful is that they don't just predict the next word based on the immediate preceding word. They consider the entire context of the input, including the structure, style, and nuances of language. For example, the phrase "bank" could refer to a financial institution or the side of a river, depending on the surrounding words. The model uses **contextual clues** to decide which meaning is more likely in any given situation.

7. Generalizing Across Texts

Through training on such a wide range of human-produced texts, LLMs learn not only specific language patterns but also how concepts are related across different domains. For example, the model might learn to associate certain actions with specific verbs, like "running" with "fast", or "rain" with "wet". These learned associations allow the model to generate coherent text across a wide variety of topics and use cases.

8. Generating New Text

When you ask the model a question or prompt it to generate text, it does so by predicting the most likely sequence of words that follows your input. It does this iteratively:

- It predicts the first word or phrase,
- Then it predicts the next word based on both the prompt and the first predicted word,
- And it continues generating further words or sentences by repeating this process.

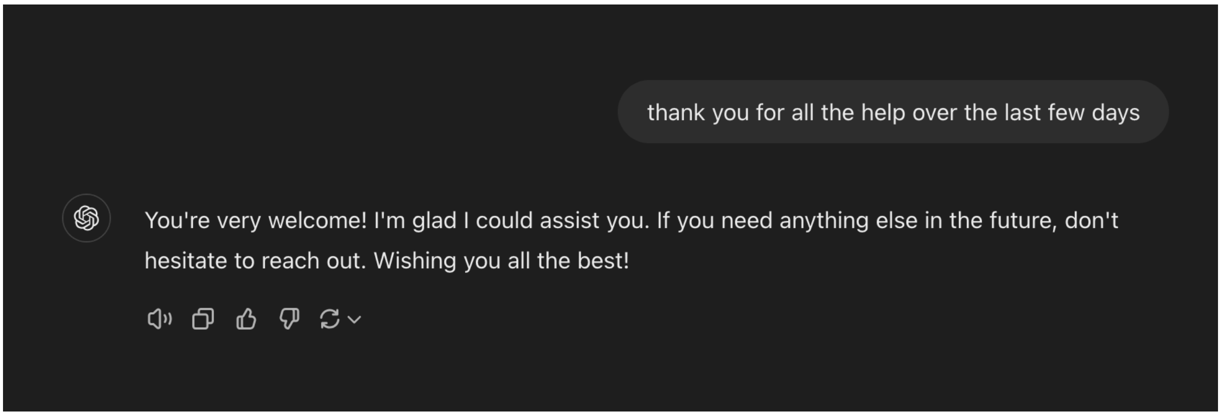
This is why LLMs are able to generate text that seems meaningful and contextually relevant, even though they do not "understand" the meaning of the words in the human sense—they are just predicting the most likely next word based on patterns learned from vast amounts of human text.

Conclusion

In summary, LLMs work by training on vast amounts of human text and learning the statistical relationships between words, phrases, and sequences. They generate predictions about what comes next in a sequence based on the associations they have learned during training. This ability to predict the next part of a sequence, grounded in patterns and associations, allows them to generate coherent text and understand context, making them powerful tools for natural language processing.

Man-Computer Symbiosis

The vast majority of this essay was written by ChatGPT. A good chunk of the images in this book were generated by Dalle. These were all made by a collaborative process with these tools, by me asking ChatGPT for questions, essays, images, and then by giving feedback on where it could improve. Essentially, for the parts of the book that are more formulaic, I used AI. The handwritten parts are from my lived experience. The whole book was a collaboration between the two. I tried my best to be nice and polite the entire time to ChatGPT.



5.2 The Law of Large Numbers [9] and the Central Limit Theorem: Foundations of Probability

Introduction

Probability theory provides a structured framework to understand uncertainty and randomness in the natural world. Two cornerstone concepts within this field are the *Law of Large Numbers* (LLN) and the *Central Limit Theorem* (CLT). These theorems, often working in tandem, offer profound insights into the behavior of repeated random experiments, guiding the way we think about statistics and data.

The Law of Large Numbers

The Law of Large Numbers is an elegant principle that describes the stabilization of sample averages as the number of trials increases. Formally, the LLN states that as the number of independent and identically distributed (i.i.d.) trials of a random variable grows, the sample mean converges to the expected value of the random variable.

Illustration Through Coin Flipping

Consider flipping a fair coin. The probability of landing heads in a single trial is 0.5. If we flip the coin a small number of times, we might observe outcomes that deviate from this expectation—for instance, 7 heads in 10 flips. However, as we repeat the experiment thousands of times, the proportion of heads will tend toward 0.5. This phenomenon is the essence of the LLN: the empirical probability converges to the theoretical probability.

The significance of the LLN is widespread. It underpins fields like insurance and finance, where averages over large populations are used to predict outcomes with remarkable accuracy. By ensuring that sample statistics reflect population parameters, the LLN builds a bridge between theory and observation.

The Central Limit Theorem

While the LLN describes the stabilization of averages, the Central Limit Theorem delves deeper into their distribution. The CLT states that, regardless of the underlying distribution of a random variable, the distribution of the sample mean approaches a normal (Gaussian) distribution as the sample size grows, provided the random variable has a finite mean and variance.

Key Features of the CLT

- **Universality:** The CLT applies to nearly any random variable, irrespective of its original distribution—be it uniform, exponential, or skewed.

- **Shape of the Distribution:** As sample size increases, the shape of the sampling distribution of the mean becomes bell-shaped, centered around the population mean, with a standard deviation proportional to $\frac{\sigma}{\sqrt{n}}$, where σ is the population standard deviation and n is the sample size.

Example: Rolling Dice

Imagine rolling a six-sided die. The sum of the outcomes for a small number of rolls might not resemble any familiar distribution. But if we repeatedly roll the die and calculate the average outcome over increasingly large groups, the distribution of these averages will approximate a normal curve, centered at the expected value of 3.5.

Practical Applications and Interplay

The combination of the LLN and CLT is foundational to modern statistical inference. Together, they justify the use of sample statistics to estimate population parameters and assess uncertainties:

- **Sampling and Estimation:** The LLN ensures that sample averages provide reliable estimates of population means, while the CLT allows us to calculate confidence intervals and make probabilistic predictions.
- **Predictive Models:** In machine learning and artificial intelligence, these theorems underlie methods for training models and evaluating their performance over large datasets.
- **Finance and Risk Management:** Financial analysts rely on these principles to model stock returns, assess risks, and optimize portfolios.

A Philosophical Reflection

Both the LLN and CLT highlight the surprising order embedded within randomness. They demonstrate that even in the face of individual uncertainty, patterns emerge when viewed at scale. This interplay between chaos and structure is not just a mathematical truth—it resonates with broader themes in science and philosophy.

Conclusion

In summary, the Law of Large Numbers and the Central Limit Theorem are more than mathematical theorems; they are lenses through which we perceive and interpret randomness. Their implications ripple across disciplines, shaping how we measure, predict, and reason about the world.

```
import noise

from pdesign import canvas, shapes, lines
from pdesign import transforms as trans
from pdesign import smooth as smooth_lib


import numpy as np
from shapely.geometry import MultiLineString, LineString, Point, Polygon, MultiPoir
from shapely.geometry import box as Box
from shapely.ops import unary_union
from shapely import affinity
import matplotlib.pyplot as plt
from matplotlib.patches import Circle

from skimage.draw import line, circle_perimeter
from skimage import draw

from ipywidgets import widgets
from ipywidgets import interact, interact_manual, interactive

from matplotlib.collections import LineCollection
from matplotlib.collections import PatchCollection

from scipy.ndimage import filters
from scipy import signal
from scipy.stats import multivariate_normal

import skimage
from skimage.feature import shape_index

from skimage import draw
from sklearn import preprocessing
from scipy import interpolate

from plottermagic.io import io

from scipy.ndimage import filters
from scipy import ndimage, misc

from skimage import exposure

from skimage.transform import rescale

from skimage import io as skio


from skimage.morphology import disk
from skimage.filters import rank, unsharp_mask
```

```
from skimage.transform import pyramids
from skimage import transform

import heapq

from tqdm import notebook
```

```
img = io.load_image("/Users/gnb/source_photos/marie/midport_nobackround.jpg", as_
img = transform.rescale(img, 1/5)
```

```
#base_img = io.load_image("/Users/gnb/source_photos/geoff_japan.jpg", as_type='gr
```

```
img.shape
```

```
(1152, 768)
```

```
plt.figure(figsize=(20,20))
plt.imshow(-img, cmap='Greys')
```

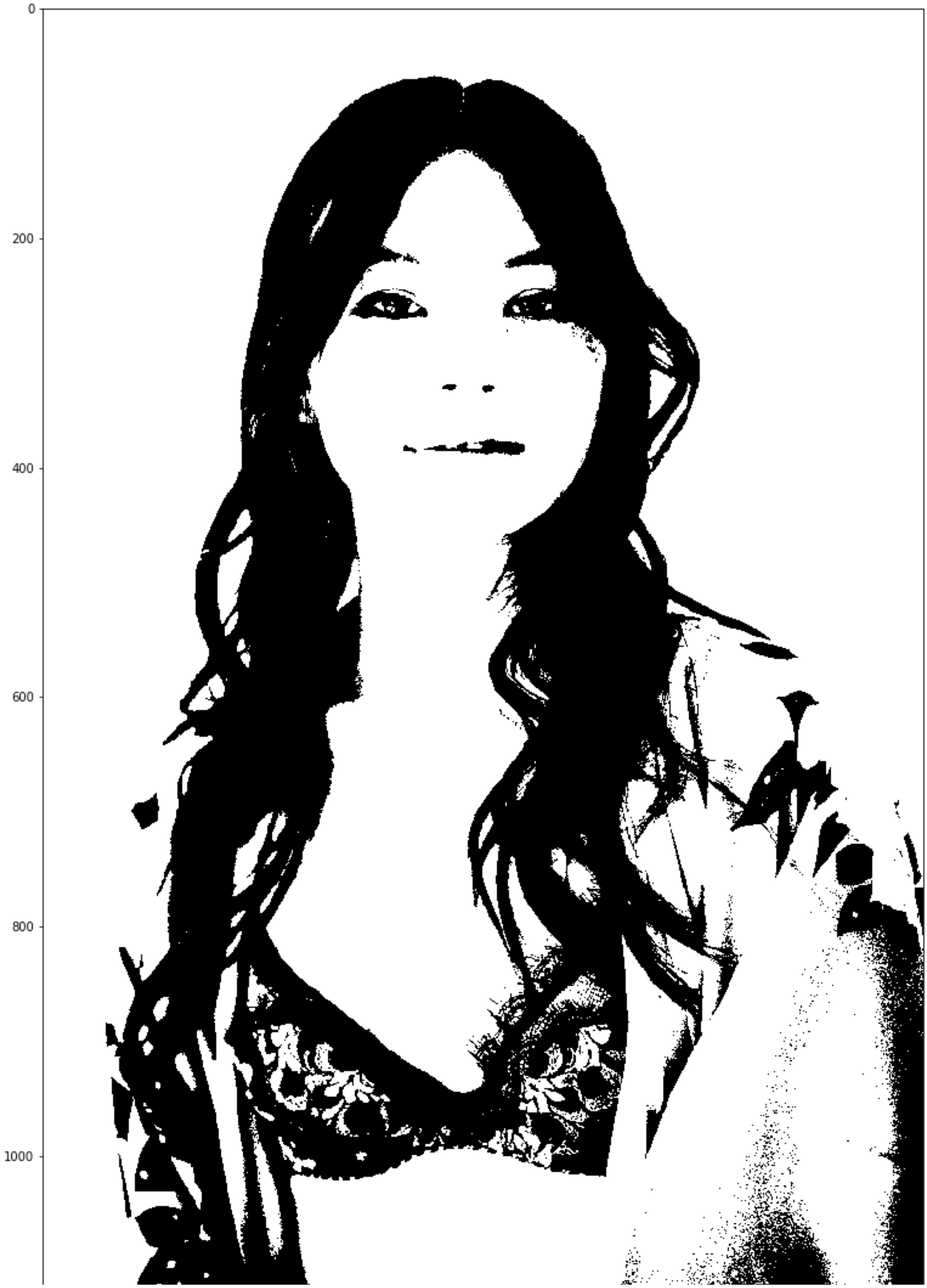
```
<matplotlib.image.AxesImage at 0x13882a110>
```

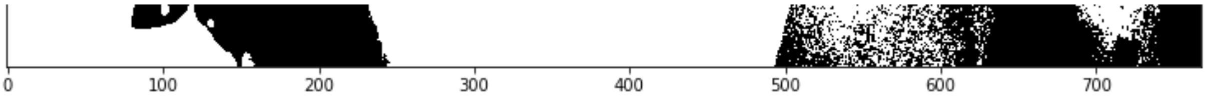

```
img_original = img.copy()
intensity = img.copy()
has_seen = img.copy()<0
```

```
dithered = np.zeros_like(img)

threshold = 60
diffuse_radius = 0.03*img.shape[0]
k = 2

g_minus = 5
g_plus = 1


plt.figure(figsize=(20,20))
```

```
plt.figure(figsize=(20,20))
plt.imshow(img<threshold, cmap='Greys')
```

```
<matplotlib.image.AxesImage at 0x138971e90>
```






```
xx, yy = np.meshgrid(np.arange(img.shape[0]), np.arange(img.shape[1]), indexing='


hp = []
ptuples = np.dstack([np.maximum(img, 255-img), xx, yy]).reshape(-1, 3)

for i, x, y in ptuples:
    heapq.heappush(hp, (i, (int(x),int(y))))



#r_min, r_max = 0.01, 0.25
r_min, r_max = 0.75, 1.25
stipple_size = lambda i :r_min + ((r_max-r_min)/255)*(255-i)


def error_diffuse(pind, err, intensity, r):

    cx, cy = draw.circle(*pind, diffuse_radius, shape=intensity.shape)
    cx, cy = cx[np.argwhere(~has_seen[cx, cy])].reshape(-1), cy[np.argwhere(~has_

    if len(cx)>0:

        rmn = ((cx-pind[0])**2+(cy-pind[1])**2)**0.5

        wmn = intensity[cx, cy]
        if err<0:
            wmn = 255-wmn

        wmn = wmn/(rmn**k)
        wmn = wmn/np.maximum(np.sum(wmn), 1e-5)

        if err>0:
            s = r**g_plus
        else:
            s = r**-g_minus

        intensity[cx, cy] += err*wmn*s
```

```
        intensity[cx, cy] = np.clip(intensity[cx, cy], 0, 255)
```

```
Start coding or generate with AI.
```

```
dumb_count = 0
stipples = []

while len(hp)>0:

    dumb_count+=1

    if dumb_count%(img.shape[0]*img.shape[1]//10)==0:
        print(dumb_count, np.mean(has_seen))




    priority, pind = heapq.heappop(hp)

    new_priority = np.maximum(intensity[pind], 255-intensity[pind])
    if new_priority != priority:
        heapq.heappush(hp, (new_priority, pind))
    else:

        r = stipple_size(intensity[pind])

        if intensity[pind]<threshold:
            app = 0
            stipples.append((pind, r))
        else:
            app = 255

        dithered[pind] = app
        has_seen[pind] = True

        err = intensity[pind]-app

        error_diffuse(pind, err, intensity, r)


    """
    if dumb_count>300000:
        break
    """
```

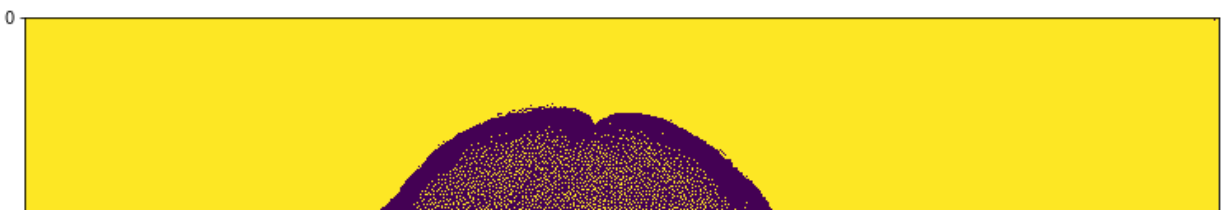
```
88473 0.013283058449074073
176946 0.025204128689236112
265419 0.03656457971643518
353892 0.04671450014467592
442365 0.05625293872974537
530838 0.06540708188657407
619311 0.07500768590856481
707784 0.084381103515625
796257 0.09279943395543981
884730 0.10013382523148148
973203 0.10682621708622685
1061676 0.11332307038483797
1150149 0.1191440158420139
1238622 0.12458970811631945
1327095 0.1296115451388889
1415568 0.13447288230613427
1504041 0.13921667028356483
1592514 0.14387003580729166
1680987 0.1521007396556713
1769460 0.17449159975405093
1857933 0.2043524848090278
1946406 0.23789921513310186
2034879 0.2732001410590278
2123352 0.3093849464699074
2211825 0.34639598705150465
2300298 0.38334599247685186
2388771 0.42200611255787035
2477244 0.46220341435185186
2565717 0.5030201099537037
2654190 0.5446449562355324
2742663 0.5843053747106481
2831136 0.6169704861111112
2919609 0.6820441351996528
3008082 0.78204345703125
3096555 0.8820427788628472
3185028 0.9820421006944444
```

```
len(stipples)
```

```
344378
```

```
fig,ax = plt.subplots(figsize=(20,20))
ax.imshow(dithered)
```

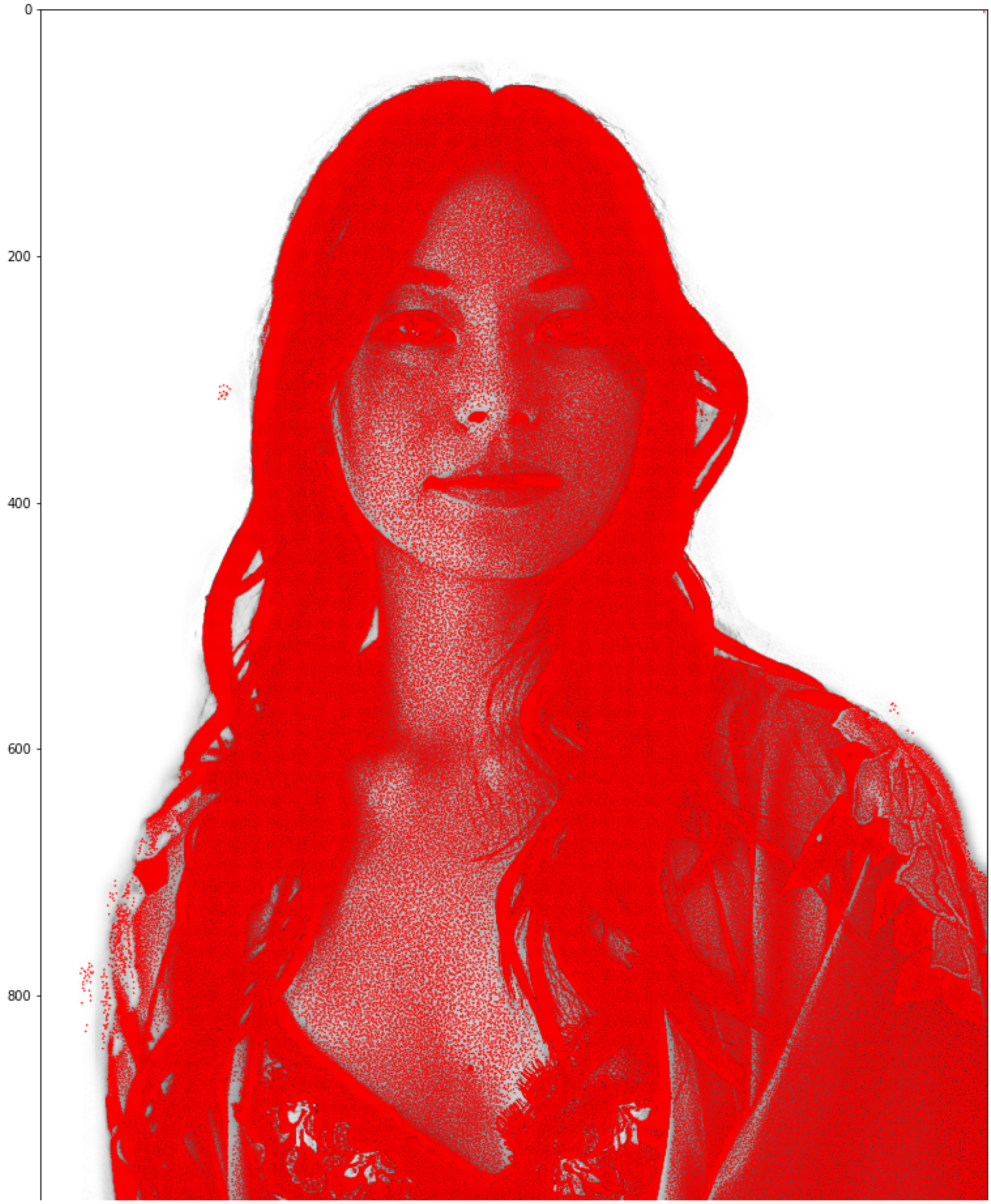
```
<matplotlib.image.AxesImage at 0x1460b1710>
```

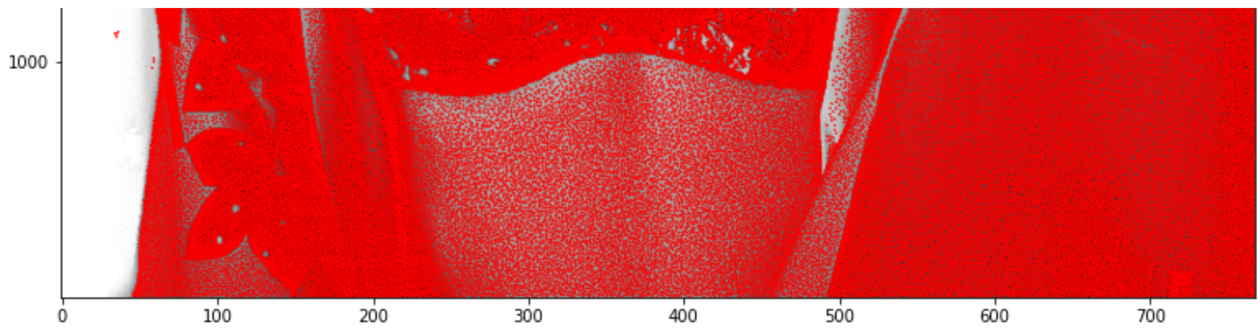



200
400
600
800
1000
0
100
200
300
400
500
600
700

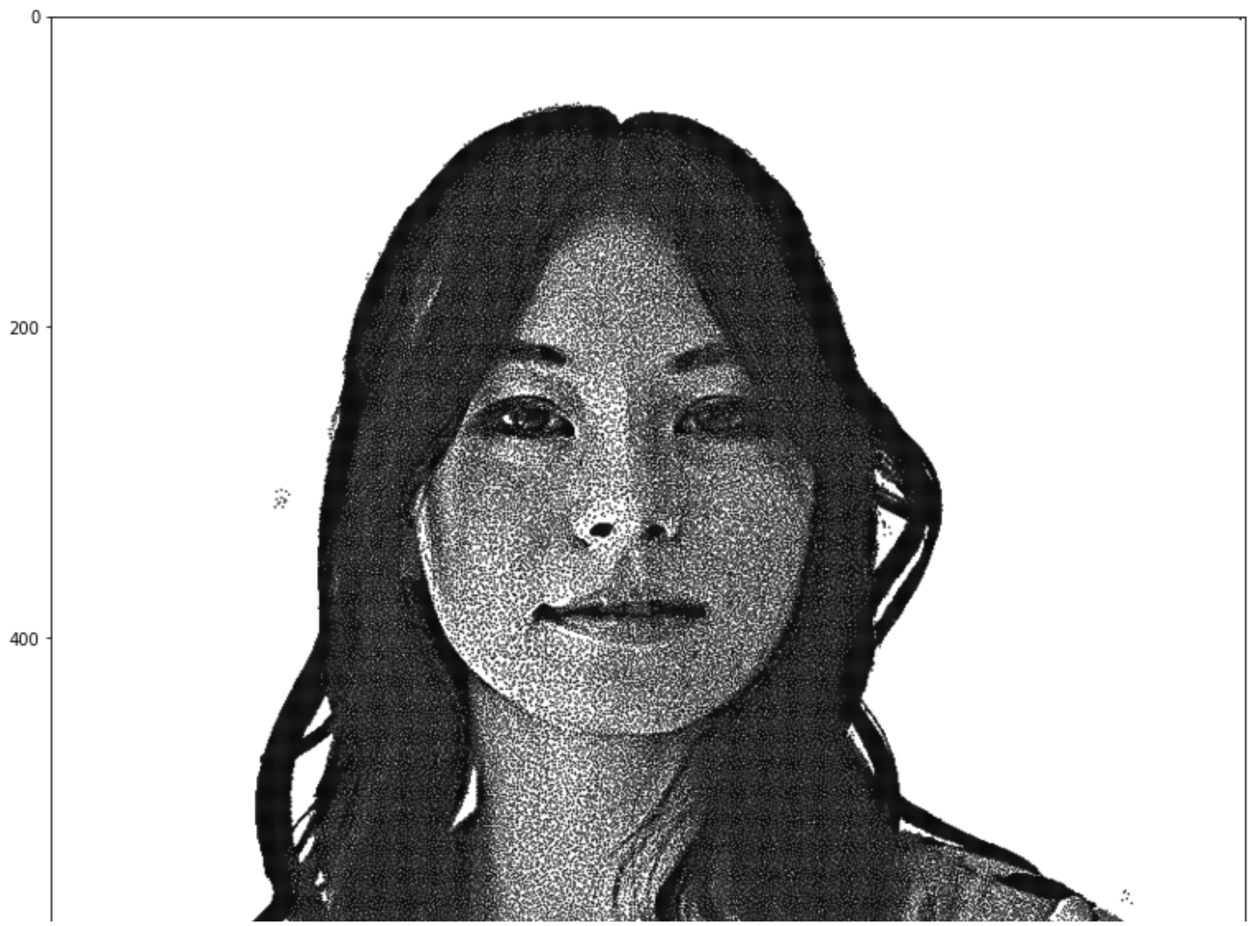
```
fig,ax = plt.subplots(figsize=(20,20))
ax.imshow(-img_original, cmap='Greys')
ax.add_collection(PatchCollection([Circle(xy[::-1], r/4) for xy, r in stipples],
```

```
<matplotlib.collections.PatchCollection at 0x1454d2bd0>
```





```
fig,ax = plt.subplots(figsize=(20,20))
ax.set_xlim(0, img_original.shape[1])
ax.set_ylim(img_original.shape[0], 0)
ax.add_collection(PatchCollection([Circle(xy[::-1], radius=r/5) for xy, r in stip
ax.set_aspect('equal')
```




```
sorted_stipples = sorted(stipples, key=lambda element: (element[0], element[1]))
```

```
import svgwrite
from svgwrite.shapes import Circle as svg_circle
```

Start coding or generate with AI.

```
rescale = img.shape[1]/9

dwg = svgwrite.Drawing(size=(11*90,14*90))
shapes = dwg.add(dwg.g(id='shapes', stroke="black", stroke_width="1", fill='none'
```

```
svg_circles = []

for xy, r in sorted_stipples:
    circ = dwg.circle(center=((str(90*(xy[1]/rescale + 1)), str(90*xy[0]/rescale)
    shapes.add(circ)


save = True
if save:
    dwg.saveas("marie_stipple_svg.svg")




"""

rescale = img.shape[1]/9

picture = canvas.Canvas(paper_size=(11,14), margin_percent=0.05, origin='corner')
picture_bbox = Box(picture.bbox[0,0], picture.bbox[0,1], picture.bbox[1,0], pictu
dp = {
    "alpha":0.7,
    "linewidth":0.25*0.0393701*72,
    "clear":False,
}

#stips = [affinity.translate(affinity.scale(Point(xy[::-1]).buffer(r/3), 1/rescal
stips = [affinity.translate(affinity.scale(Point(xy[::-1]).buffer(0.25*0.0393701*

picture.make_canvas()
picture.add_grid(11,14)
picture.plot_shapes(stips, **dp)
picture.fig

save = False

picture.make_canvas()
picture.plot_shapes(stips, **dp)
picture.display_overlays(False)
if save:
    picture.fig.savefig("marie_stipple_test.svg")
picture.fig
"""
```

```
'\n\nrescale = img.shape[1]/9\n\npicture = canvas.Canvas(paper_size=(11,14),
margin_percent=0.05, origin=\'corner\')\npicture_bbox =
Box(picture.bbox[0,0], picture.bbox[0,1], picture.bbox[1,0],
picture.bbox[1,1])\ndp = {\n    "alpha":0.7,\n
"linewidth":0.25*0.0393701*72,\n    "clear":False,\n}\n\n#stips =
[affinity.translate(affinity.scale(Point(xy[::-1]).buffer(r/3), 1/rescale, 1/
rescale, origin=(0,0)), 1,0) for xy, r in stipples[0:15000]]\nstips =
```

```
[affinity.translate(affinity.scale(Point(xy[::-1]).buffer(0.25*0.0393701*72/2
1/rescale, 1/rescale, origin=(0,0)), 1,0) for xy, r in
stipples[0:15000]]\n\npicture.make_canvas()\npicture.add_grid(11,14)\npicture
**dp)\npicture.fig\n\nsave =
False\n\npicture.make_canvas()\npicture.plot_shapes(stips,
**dp)\npicture.display_overlays(False)\nif save:\n
picture.fig.savefig("marie_stipple_test.svg")\npicture.fig\n'
```

```
"""
#stips = [affinity.translate(affinity.scale(Point(xy[::-1]).buffer(r/3), 1/rescal
#stips = [affinity.translate(affinity.scale(Point(xy[::-1]).buffer(0.25*0.0393701
stips = [Point(11 - (xy[1]/rescale + 1), xy[0]/rescale).buffer(0.25*0.0393701/2)


picture.make_canvas()
picture.add_grid(11,14)
picture.plot_shapes(stips, **dp)
picture.fig

save = True



picture.make_canvas()
picture.plot_shapes(stips, **dp)
picture.display_overlays(False)
if save:
    picture.fig.savefig("marie_stipple_full.svg")
picture.fig




mpl_circles = [Circle((11 - (xy[1]/rescale + 1), xy[0]/rescale), 0.25*0.0393701/2
picture.make_canvas()
picture.add_grid(11,14)
picture.ax.add_collection(PatchCollection(mpl_circles))
picture.fig


save = False

picture.make_canvas()
picture.ax.add_collection(PatchCollection(mpl_circles))
picture.display_overlays(False)
if save:
    picture.fig.savefig("marie_stipple_full.svg")
picture.fig
"""
```

```
'\n#stips = [affinity.translate(affinity.scale(Point(xy[::-1]).buffer(r/3),
```

```
1/rescale, 1/rescale, origin=(0,0)), 1,0) for xy, r in
stipples[0:15000]]\n#stips =
[affinity.translate(affinity.scale(Point(xy[::-1]).buffer(0.25*0.0393701*72/2
1/rescale, 1/rescale, origin=(0,0)), 1,0) for xy, r in stipples]\nstips =
[Point(11 - (xy[1]/rescale + 1), xy[0]/rescale).buffer(0.25*0.0393701/2) for
xy,r in
stipples]\n\n\npicture.make_canvas()\npicture.add_grid(11,14)\npicture.plot_sl
**dp)\npicture.fig\n\nsave =
True\n\n\n\npicture.make_canvas()\npicture.plot_shapes(stips,
**dp)\npicture.display_overlays(False)\nif save:\n
picture.fig.savefig("marie_stipple_full.svg")\npicture.fig\n\n\n\n\nmpl_circle
= [Circle((11 - (xy[1]/rescale + 1), xy[0]/rescale), 0.25*0.0393701/2) for
xy,r in
stipples]\npicture.make_canvas()\npicture.add_grid(11,14)\npicture.ax.add_col
=
False\n\npicture.make_canvas()\npicture.ax.add_collection(PatchCollection(mpl_
save:\n    picture.fig.savefig("marie_stipple_full.svg")\npicture.fig\n'
```

```
from pyaxidraw import axidraw
ad = axidraw.AxiDraw()
ad.interactive()
ad.connect()
```

```
ad.options.model = 2
ad.options.pen_pos_up = 50
ad.options.pen_pos_down = 43
ad.update()
```

```
ad.plot_setup("marie_stipple_svg.svg")
ad.plot_run()
```

```
Plot paused by button press.
Use the resume feature to continue.
```

```
ad.options.mode = "res_plot"
ad.options.pen_pos_up = 50
ad.options.pen_pos_down = 43
ad.update()

ad.plot_run()
```

```
Failed after command: SC,4,18843
Failed after command: SC,5,17585
Failed after command: SC,11,1350
Failed after command: SC,12,900
Failed after command: EM,1,1
Warning: AxiDraw movement was limited by its physical range of motion. If eve
```

Start coding or generate with AI

Start coding or generate with AI.

Start coding or generate with AI.

Chapter 6

The Dinosaur Chapter

6.1 Farmer Labs

PHOTOGRAMMETRY TECHNIQUES IN COMPUTATIONAL PALEOPHYSIOLOGY

Geoff Bradway *Mathematics/Computer Science*

Colleen Farmer *Biology*

BACKGROUND

The purpose of this project is to increase the participation and impact of undergraduate student contributions to the development of a new, project-based course, Computational Paleophysiology, in order to enhance teaching methodologies and student learning.

They will begin by studying the methods scientists use to reconstruct body mass, volume, and surface area. This is particularly important because these traits are inextricably linked with many other life-history characters (posture, gait, growth rates, daily energy expenditures, rates of reproduction, rates of heat loss or gain, foraging ranges, etc.).

Basic Process

1. Acquisition of the photographs of the specimen.
2. Production of a spare then dense point cloud.
3. Post processing

Acquisition of photographs

The three type of cameras that were used were the Pentex Optio S5i, Nikon D3000, and the Canon EOS Digital Rebel XT. The camera choice was based off of the availability and not off of any technical specifics.

The number of photographs to generate a viable model ranged greatly based upon camera choice and model complexity. In order to generate a good model, every point photographed has to be in at least three photographs from three different positions (Falkingham, 2012). For the best results, photographs every 15 degrees around the specimen is recommended (at least 24) (Falkingham, 2012). Also ideally, there should be a ~50% overlap in photos (Falkingham, 2012).

Production of point clouds and 3d models

Generating The Point Cloud

To generate the point clouds we used a free software called Bundler.

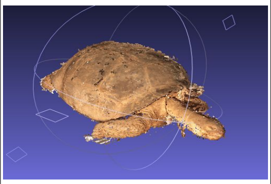
The Bundler package that is available is all inclusive. The basic Bundler package generates a sparse point cloud, and the PMVS and CMVS (Patch-based Multi-view Stero system and Clustering Views for Multi-view Stero) are two methods to generate a point cloud from from the sparse point cloud.

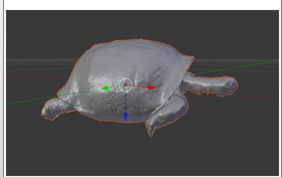
3d Model Generation

To generate the models we used an array of modern techniques.

The simplest technique is to use use one iteration of a surface reconstruction. This is done by first calculating the normal of the point set and then preforming a Poisson reconstruction. The advantage of this way is that it is simple and produces models that you can 3d print. However this method prefers convex objects and will tend to overfit the points.

The second technique is to use two iterations of a surface technique. To do this we first generate a simplistic model from Ball Pivot surface reconstruction. Then using some variant of random sampling we can generate a larger more complete point cloud. Then we run the Poisson reconstruction on this new point cloud. The advantage of this is that the Ball Pivoting fits the data very closely while still filling the wholes in the model. Then resampling and the Poisson reconstruction makes a good 3d model that fits the data closely and can be 3d printed.

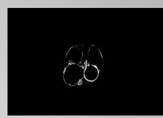


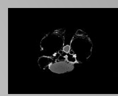


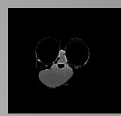
Lab Advantages

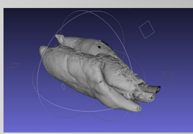
- Extremely portable and can easily be taken into the field to get laser scanner quality models, while only requiring a camera
- Extremely cheap and affordable
- Can be automated
- Can make 3d printed models

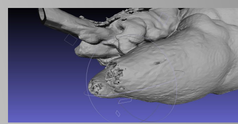
Using CT scans

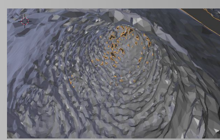


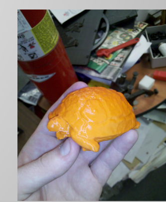


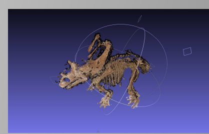












ACKNOLWEDGEMENTS & REFERENCES

Bates, Karl T., Falkingham, Peter L., Breithaupt, Brent H., Hodgetts, David, Sellers, William I., and Manning, Phillip L., 2009. How Big Was 'Big Al'? Quantifying the effect of soft tissue and osteological unknowns on mass predictions for Allosaurus (Dinosauria:Theropoda). Palaeontologia Electronica Vol. 12, Issue 3; 14A: 33p;http://palaeo electronica.org/2009_3/186/index.html

Falkingham, Peter L. 2012. Acquisition of high resolution 3D models using free, open-source, photogrammetric software.
Palaeontologia Electronica Vol. 15, Issue 1; 1T:15p;

Geoff Bradway

Collaborators:
Ben Ferguson
Leah Moelling

Introduction

Falcarius utahensis was a therizinosaur dinosaur that was discovered in the Yellow Cat formation, in Utah. Therizinosaurs were part of a much broader class of dinosaurs, theropods. Falcarius and other therizinosaurs were primarily herbivorous, which challenged the notion that theropods were primarily hypercarnivorous (Zanno, 2009).

Falcarius was excavated in the Crystal Geyser Quarry in Grand County Utah, which would date it to having lived about 126 million years ago in the early Cretaceous period (Kirkland, 2005). *Falcarius* helped shed new insight on the shift from carnivory to herbivory in Therizinosauroidea (Zanno, 2009). *Falcarius* is the earliest known therizinosaur, so it represents a good transitional fossil between hyper carnivorous theropods and the only known branch of primarily herbivorous theropods (Zanno, 2010). As a result, the physiological structure, osteological structure , and lifestyle of *Falcarius* are of great interest because it represents the drastic transition from a well adapted carnivorous form to the herbivorous form. This can be seen in the mixture of traits that *Falcarius* possesses, such as regression in cursorial adaptations on its hind limbs (Zanno, 2009).

Typically therizinosaurioids had long necks, laterally expanded pelves, small leaf shaped teeth, larges claws on their hands, and hind feet with four toes. Their forelimbs supported a range of motion that is not seen in other theropods (Burch, 2006) . *Falcarius* was a slender smaller theropod, about four meters long and one meter high at the hip (Kirkland, 2005) . It is believed to be a transition to a more herbivorous omnivore because of its teeth and an osteological hind limb structure designed to support large weight (Zanno, 2009). *Falcarius* has an increased number of digits on its hindlimbs believed to help support weight, and a shortening of the tibia relative to the femur, both of which are a reversal of cursorial adaptations. This dinosaur has small leaf shaped teeth likely with a keratin beak, which is seen in other herbivores. In addition, *Falcarius* had a displaced pubis with an expanded pelvis which suggests an increase in intestinal mass (Kirkland 2005). All of these suggest a slower moving dinosaur, which suggests more of a herbivore than a carnivorous predator.

With reconstruction techniques and computational simulations, we can hope to gauge this change in overall form from a well adapted carnivore to a herbivore. We used photogrammetry to first establish a working 3D model of *Falcarius* to use for simulations. Photogrammetry is using a series of photographs to generate a 3D model. Intuitively this process is similar to how our eyes utilize stereoscopic vision to generate a 3D view.

With this 3D model, we can then analyze body features that allude to lifestyle. Since presently there are many allometric mass curves to describe features in extant animals, we started by using volumetric reconstructions from the skeleton to gauge a range of body mass. We used modeling and mathematical software to generate a range of viable volumes, and extant animal physiological densities to deduce a range of masses.

To test the viability of *Falcarius* as a predator we analyzed a wide range of aspects. We reconstructed a hindlimb to examine the structure for cursorial adaptations. If *Falcarius* is this transition between these predators and the slow moving therizinosaurs, we should be able to see a decrease in cursorial aptitude. Furthermore, to analyze what the likely speed of *Falcarius* would have been, we reconstructed M. caudofemoralis in *Falcarius*. This muscle in an integral part of the movement in many extant reptiles, such as alligators, and size helps determine maximum movement. Such a reconstruction then can help us determine agility and athleticism in *Falcarius*, and while the presence of such features can not help us determine carnivorous verses herbivorous behavior, the lack of such features would imply herbivorous or scavenger behavior, since it would unlikely that this dinosaur could have efficiently caught prey .

Next we used computational models to simulate bite force in extant alligators to try and predict what the likely bite force in *Falcarius* would have been. Clearly a high bite force can be linked with predatory behavior, whereas a low bite force might allude to either herbivorous or possibly scavenger behavior. We used juvenile alligators due to the similarity in skull size.

Next we used extant animal data and our previous estimate to model the cardiac and respiratory system of *Falcarius*. This allows use to assess if the cardiac output would have been sufficient to support a high activity lifestyle. The cardiac output cannot imply much about predatory or herbatory behavior since there are extant animals of both classes with high and low cardiac output. However, the cardiac capabilities does shed light on the lifestyle this dinosaur would have lived given either behavior.

Finally we analyzed the inner ear structure of this dinosaur. This allows us to infer about the auditory capabilities of this dinosaur, which then imply information about its lifestyle. A well developed auditory system could be easily utilized in any life style, but it also suggests a high degree of sociability (Walsh 2008), which would have implications about the predatory or herbatory structure used.

Photogrammetry

Materials and Methods:

In order to fully calibrate the software system for classroom use, the technique was applied to variety of objects, ranging from drastically different sizes, and utilizing different cameras. As the technique is virtually the same for all of these, the process will be described in general. The process of producing a 3D model from a specimen can be given by the following:

1. Acquisition of the photographs of the specimen.
2. Production of a sparse point cloud.
3. Production of a dense point cloud.
4. Post processing.

1. Acquisition of photographs

The three type of cameras that were used were the Pentex Optio S5i, Nikon D3000, and the Canon EOS Digital Rebel XT. The camera choice was based off of the availability and not off of any technical specifics.

The number of photographs to generate a viable model ranged greatly based upon camera choice and model complexity. In order to generate a good model, every point photographed has to be in at least three photographs from three different positions (Falkingham, 2012). For the best results, photographs every 15 degrees around the specimen is recommended (at least 24) (Falkingham, 2012). Also ideally, there should be a ~50% overlap in photos (Falkingham, 2012).

2. Production of a sparse point cloud / 3. Production of a dense point cloud

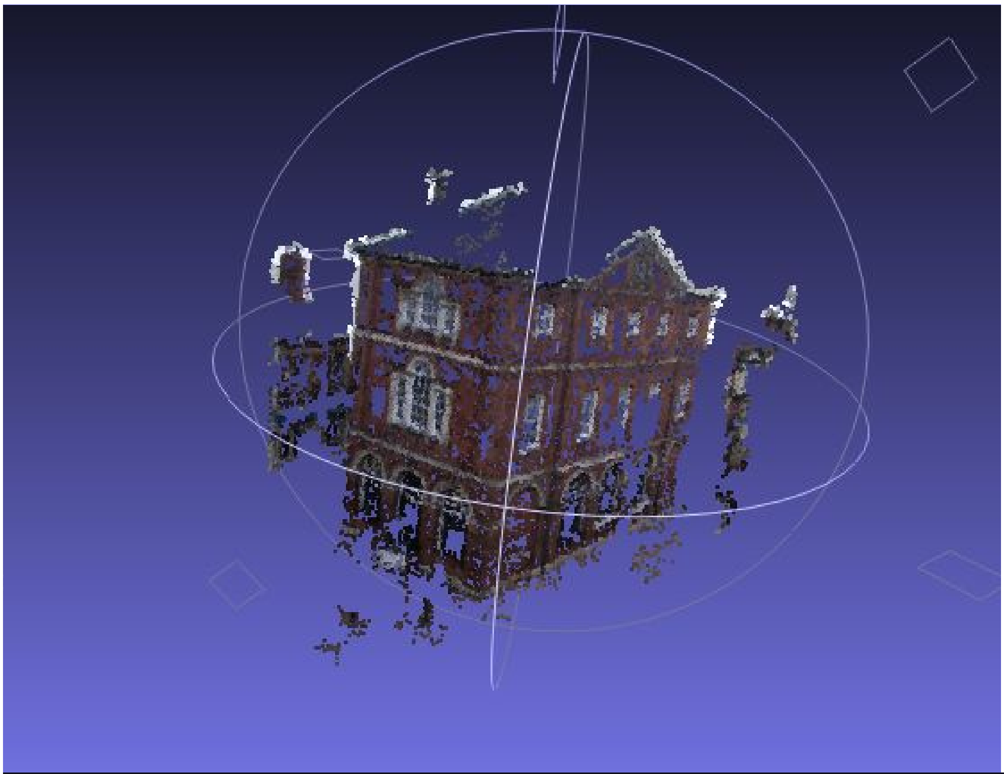
The sparse point cloud was produced with a freeware software called Bundler. For installation specifics see the installation appendix (after references). The Bundler package that is available is all inclusive. The basic Bundler package generates a sparse point cloud, and the PMVS and CMVS (Patch-based Multi-view Stero system and Clustering Views for Multi-view Stero) are two methods to generate a point cloud from from the sparse point cloud. PMVS is used for smaller photo sets while CMVS is used on larger photo sets and calls PMVS first. For this paper we used PMVS.

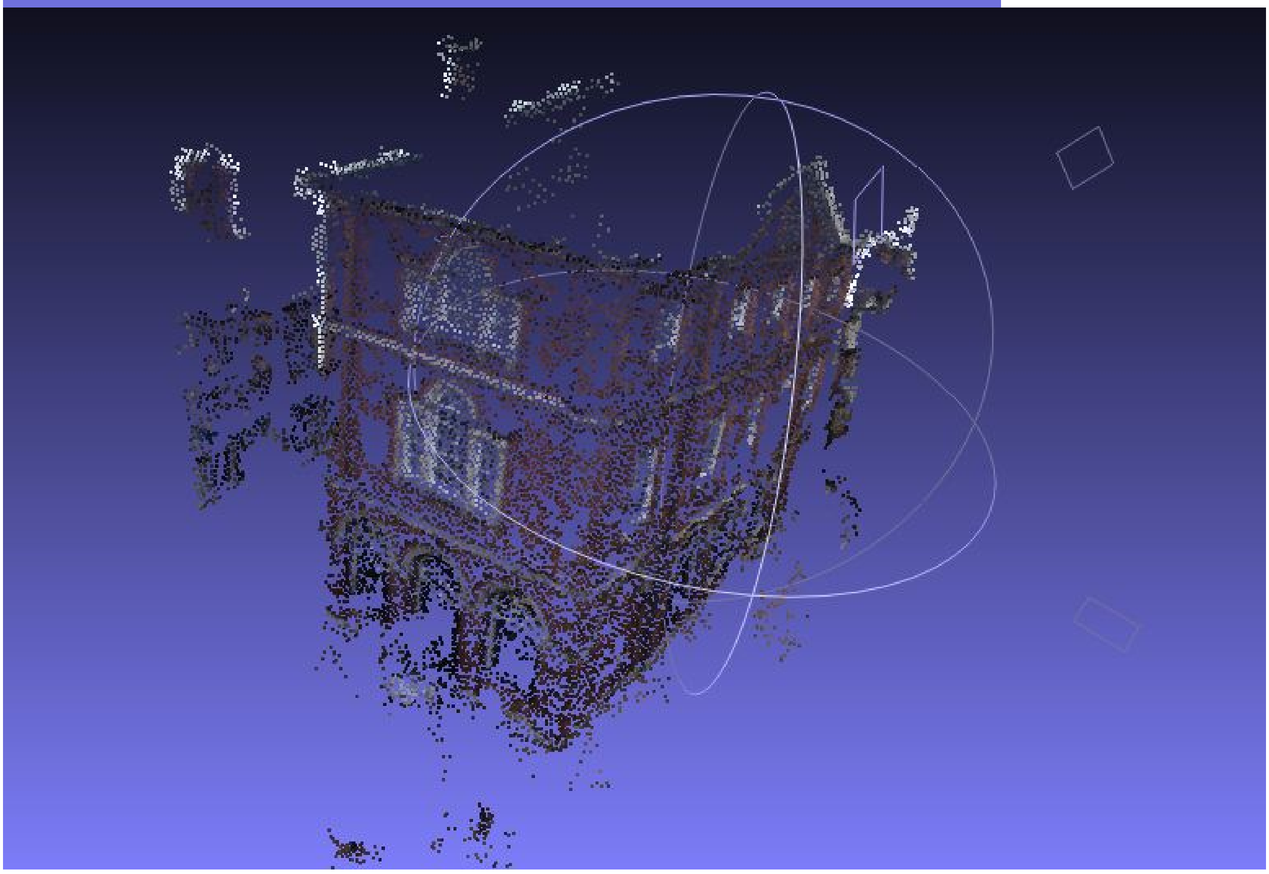
4. Post processing

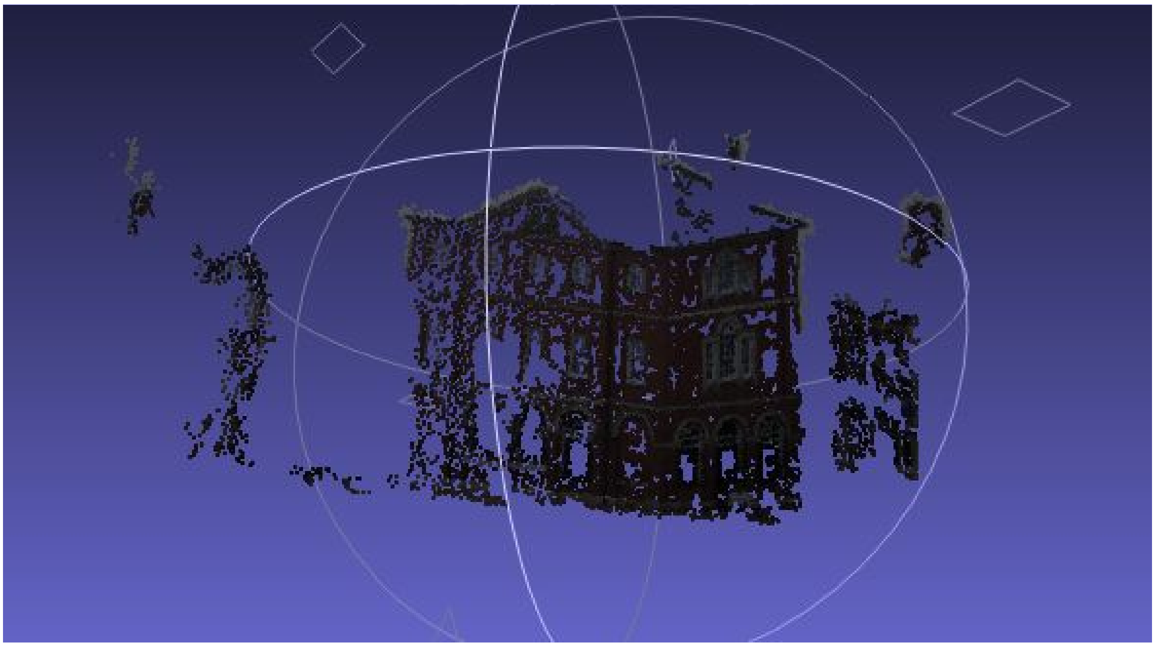
The file that is outputted by the PMVS script is a .PLY which is a CAD file that can be opened and read by many computer programs. For this test, we used the free software MeshLab. Since the 3D model was taken with photographs there are a couple of negative results. First off, the result is scaleless, so some sort of scale has to be known beforehand, or included while sampling the specimen. Also the software will reconstruct extraneous objects in the background, but these can be readily deleted. However, we have found that it useful to have objects in the background as it readily provides key points to help the software calibrate the photos. After this, the model can be treated as if it were a laser scanned model.

Results (figures included):

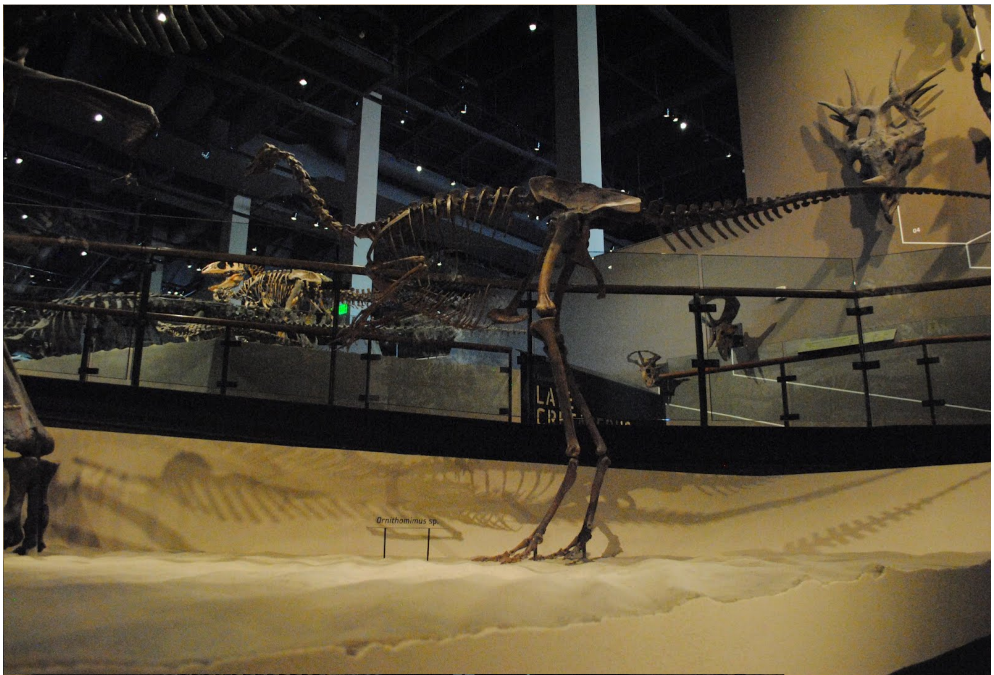
For the first set, we tried the sample photos included in the software. There were four photos and I got the following model. This was with the Pentex Optio S5i.

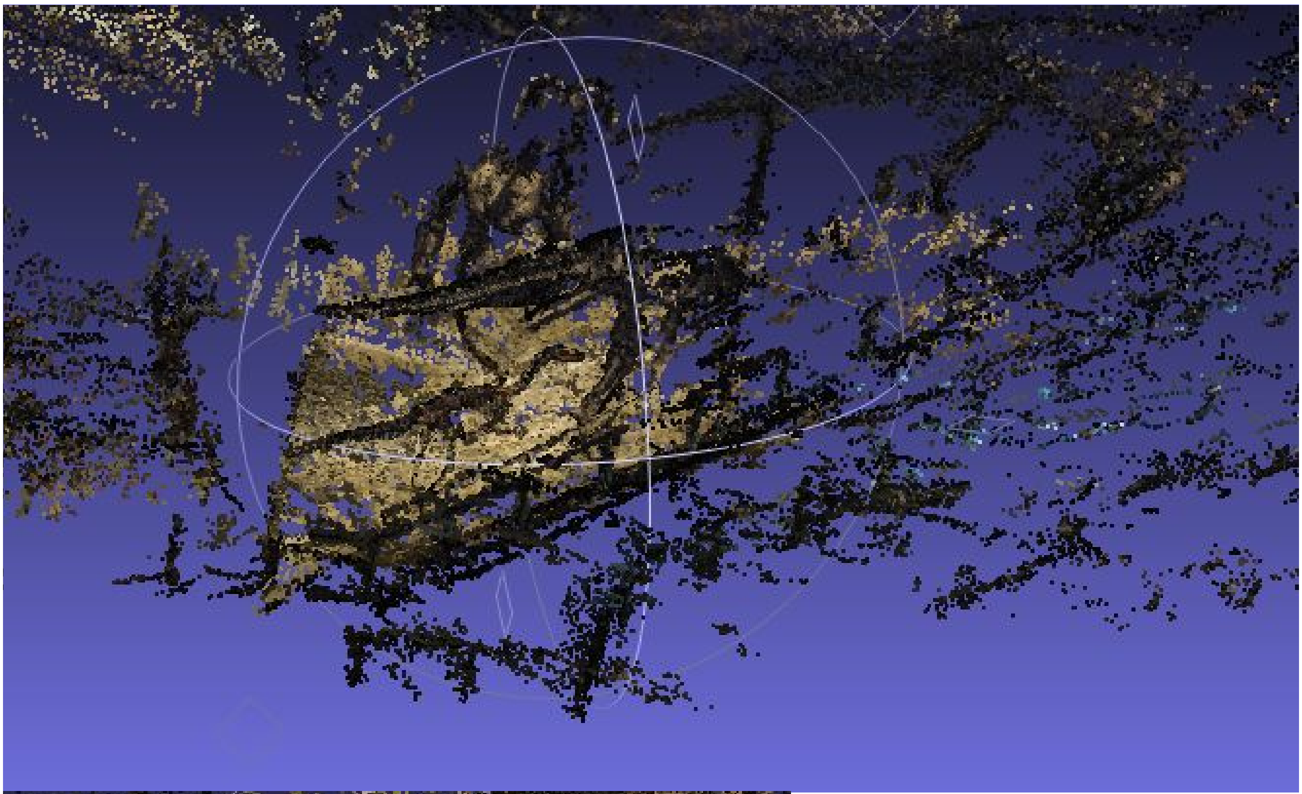


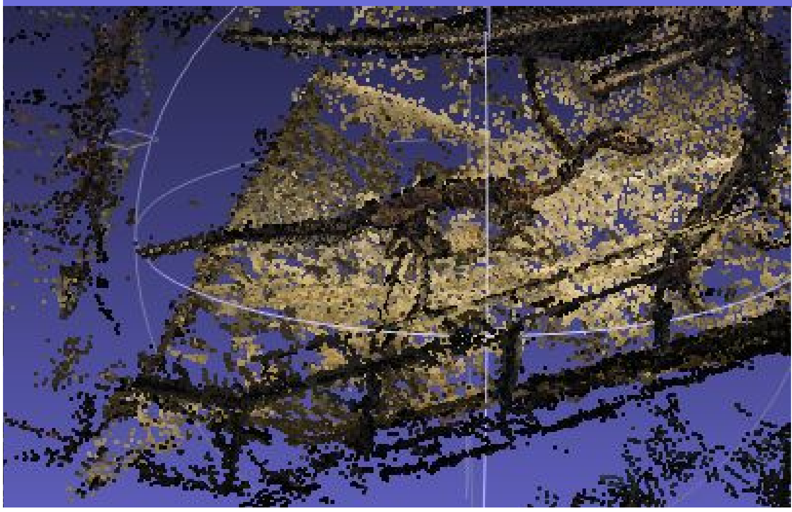


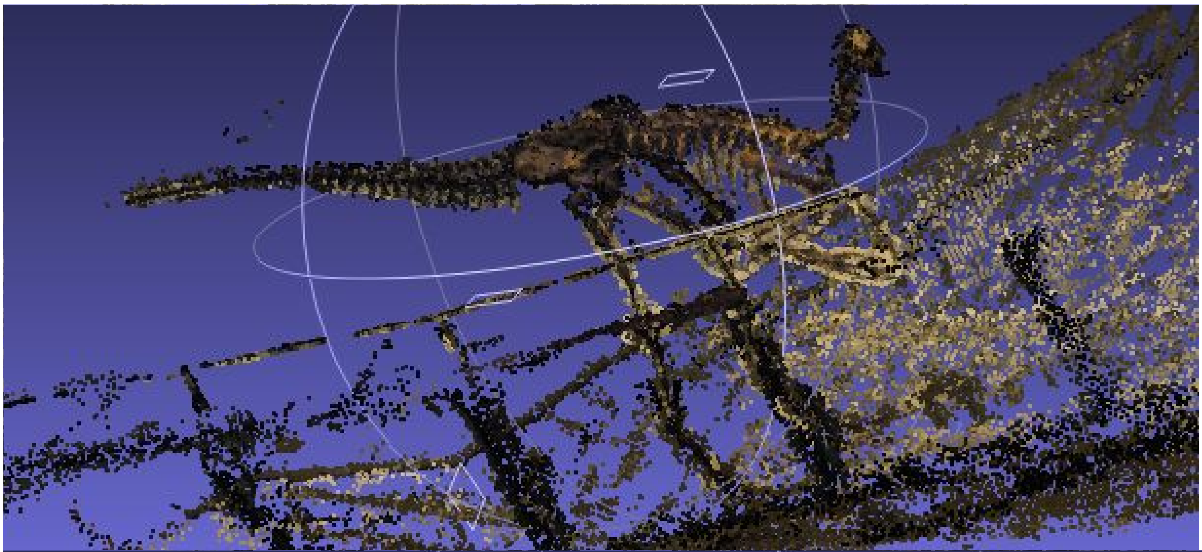
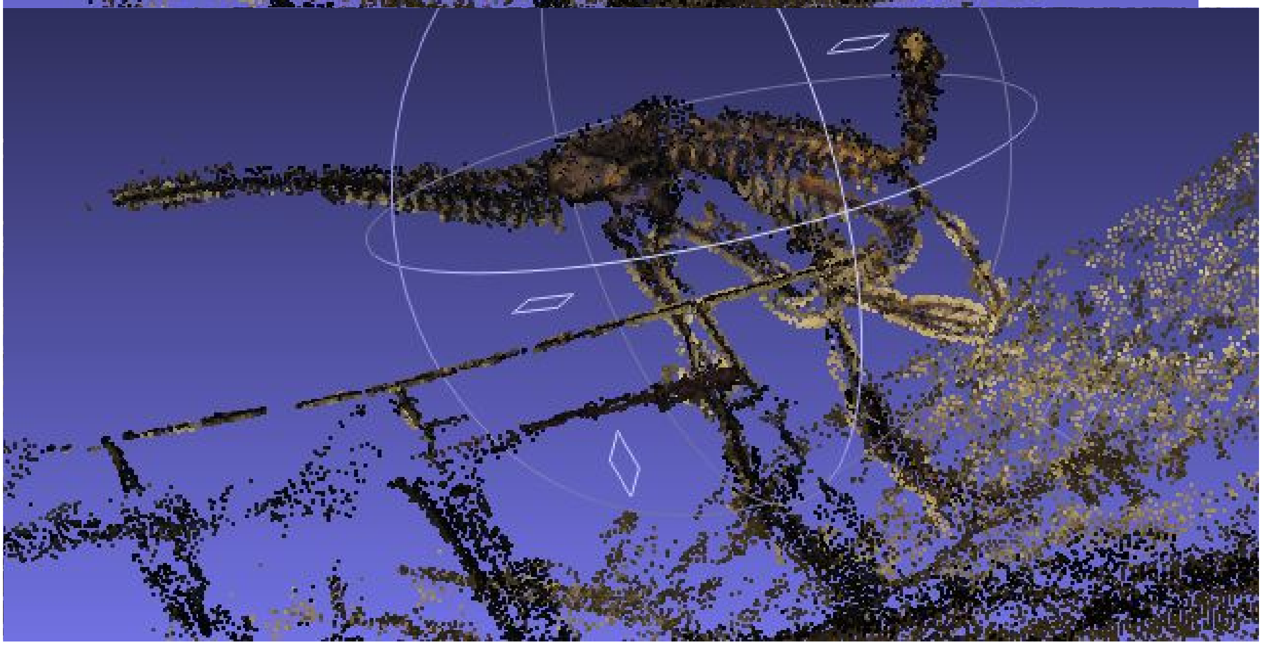


For my second set we used about 200 photos we had of an ornithopod taken in the museum. These were taken with a Nikon D3000.

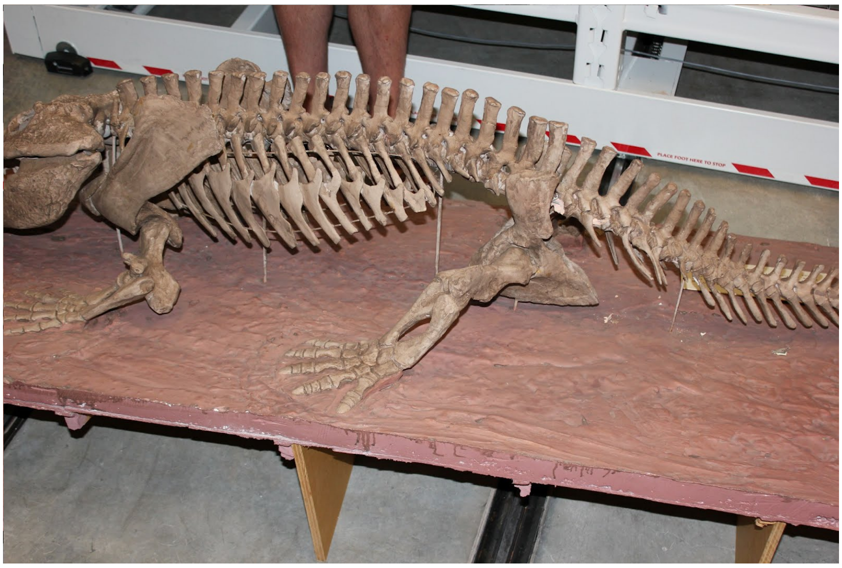

Ornithomimus sp.

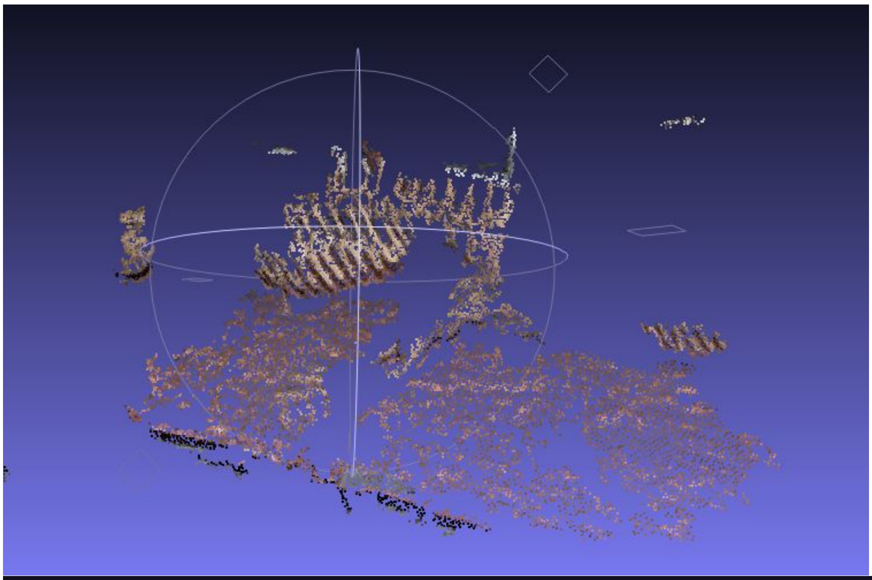


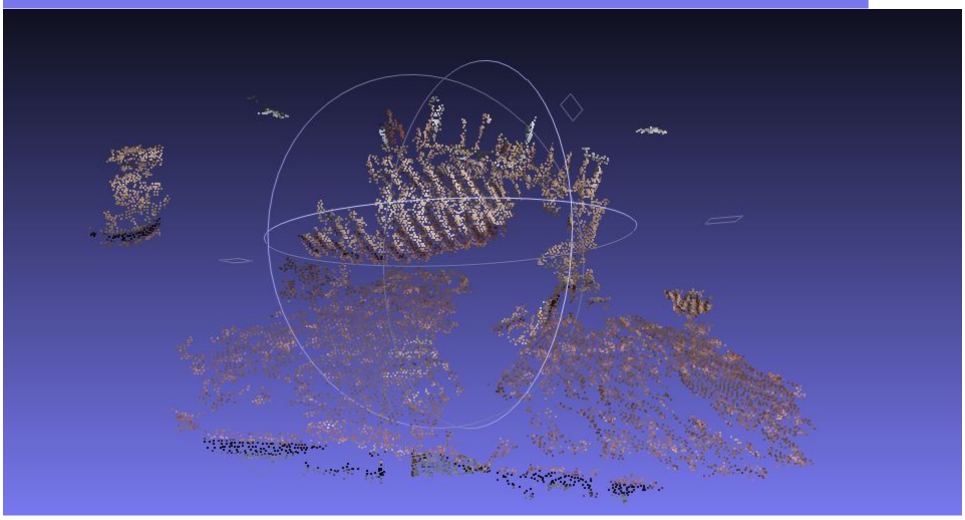


Next with a small photoset (about 40 photos) of an amphibian, taken in the UMNH, we got the following results. This was taken with a Canon EOS Digital Rebel XT. Unfortunately we did not get optimal results. This is due to the limited exposure range, few misc background objects (helps determine positions and scale) and not enough overlap in photos.

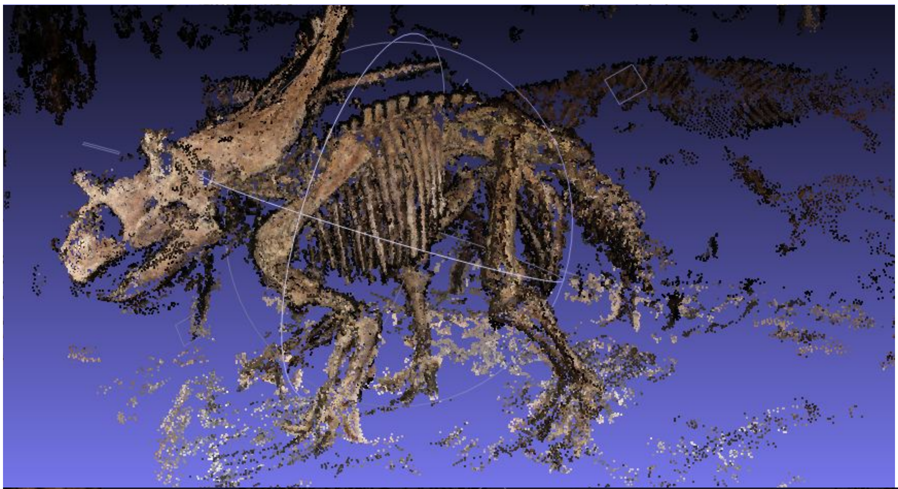


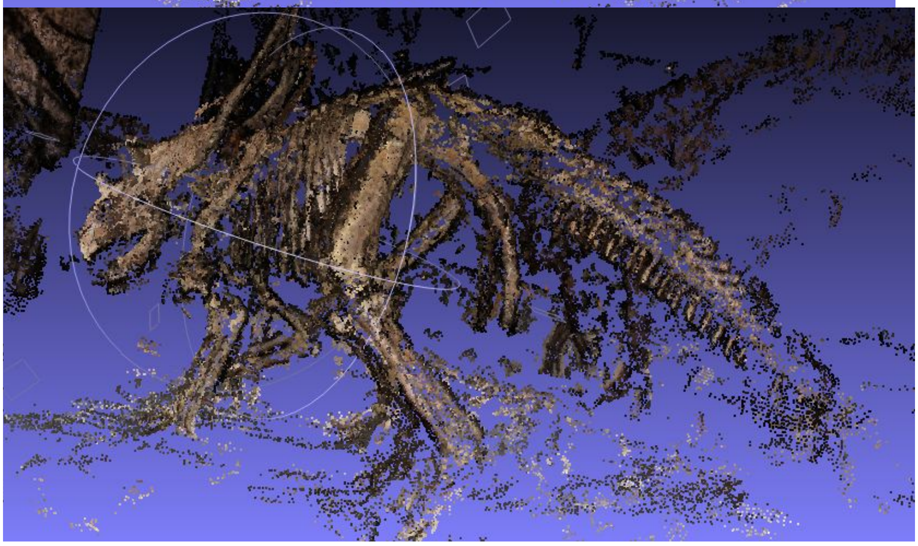




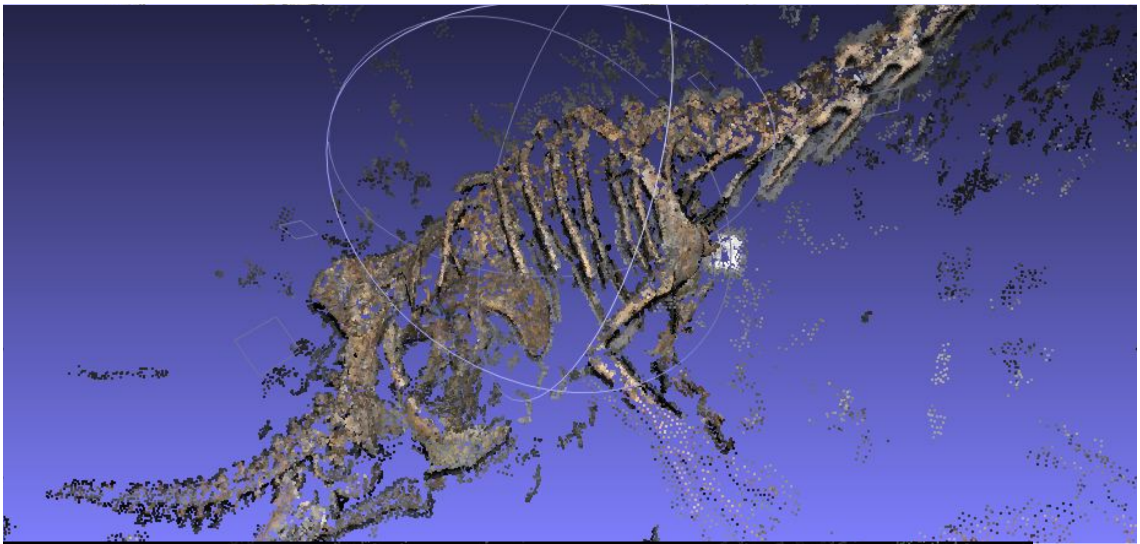
Next with a photoset (about 100 photos) of an ceratopsian, taken in the UMNH, we got the following results. This was taken with a Canon EOS Digital Rebel XT.

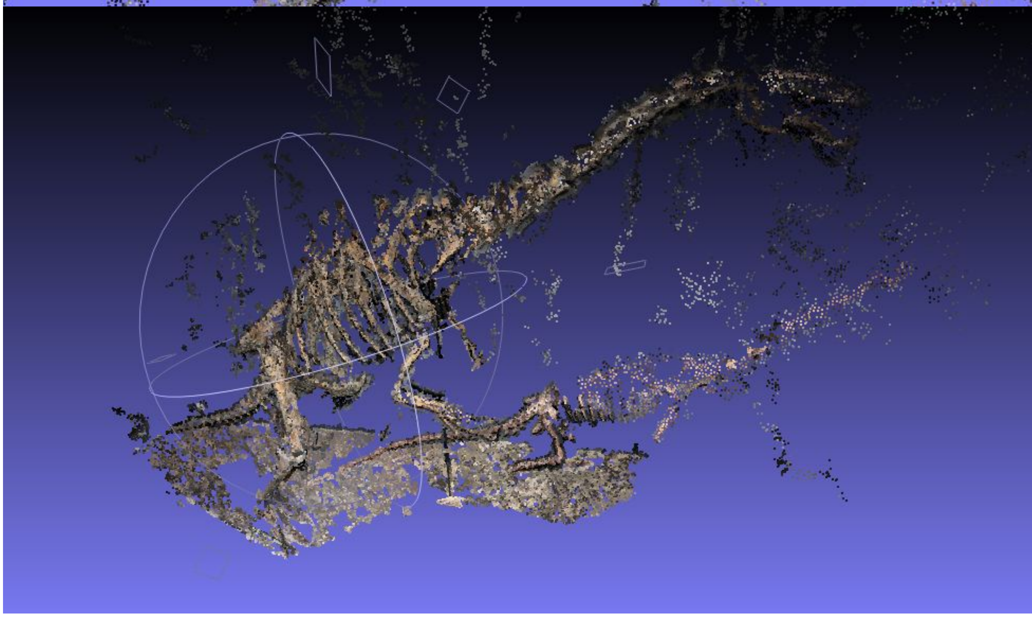


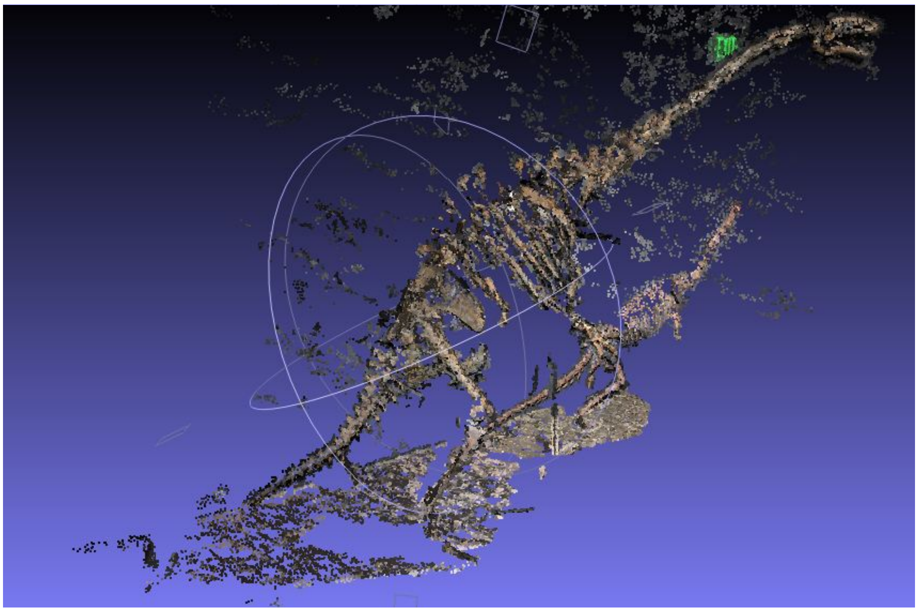





Finally with a photoset (about 150 photos) we were able to generate *Falcarius*, taken in the UMNH, we got the following results. This was taken with a Canon EOS Digital Rebel XT.







Volumetric Reconstruction

Materials and methods:

For the skeletal models for the volumetric reconstruction we used models obtained by RIEGL LMS-Z420i 3D terrestrial Light Detection and Range (LiDAR) scanner (Hutchinson, 2011). For a more detailed insight on how the models were obtained, see (Bates,2009) or (Hutchinson, 2011). The reason that we used a scanner and not photogrammetry was due to time constraints coupled with improper use of the photogrammetry software, and we had scanned models on similar dinosaurs.

For the model handling we used Maya 2012 (Bates, 2009). Then we used a MatLab script to determine the smallest convex volume around a given set of bones. A convex set is a set in which if X and Y are points in the set, then the line from X to Y is contained in the set. Then we would partition the skeleton into sets where a convex volume surrounding it seemed physiologically likely based off of extant animal proportions. We did this for the entire skeleton and used this to derive the minimum possible mass. We then adjusted the volume of these minimized portions to derive reasonable estimates for volume, and estimates that were believed to be overestimates. Finally we would estimate the average volume density in a section, ie a jaw is mostly bone, but some muscle, so an approximation of 80% bone and 20% muscle was used. The densities used for muscle were 1006 kg/m^3 and for bone was 1500 kg/m^3. Finally we also adjusted each proportion to try a create a caudal and cranial centered animal.

We did this volumetric analysis on *Tyrannosaurus Rex* to compare our methodology with the that of current literature, namely (Hutchinson, 2011). Then, we reran the method on *Plateosaurus* due to the resemblance of *Falcarius* in body form.

Results:

For *Tyrannosaurus Rex* the predicted range was from ~6150- 12550 kg. Similar conducted computational estimates that a typical Trex was around 6000-8000 kg, while “Sue”, the largest found instance of Tyrannosaurus Rex, would have weighed around 9500 kg (Hutchinson, 2011). As we can see, the lower end of our ranges are similar, while upper range is vastly different, however if we include Hutchinson's value for a maximal value, we get that ranges are similar.

For the normalization we got the following results as compared to Hutchinsons the following results. (For the comparison we used the the Trex with similar mass from Hutchinson, as he tested it with multiple models). As we can see the results are somewhat close in the cranial and and max model. However for the caudal and minimizing model have off with their COM X. Since we reused a lot of parts in the caudal model from the minimizing model it's probable that we made a mistake on a more cranial body part that is present in both. Note that the COM X,Y are relative to the hip joint and the normal X, and Y have been normalized with the femur length.

	Mass	COM X	COM Y	normal X	norm Y
Caudal	6830	-0.346	-0.381	-27.1	-29.9
Hutchinson Stan (most ventral)	7746	0.378	-0.63	29.5	-49.9
Cranial	9100	0.556	-0.351	43.6	-27.5
Hutchinson Carnegie (most ventral)	8888	0.447	-0.388	35.3	-30.7
Min	6158	-0.029	-0.410	-2.2	-32.1
Hutchinson Stan (min)	5934	0.524	-0.49	41.0	-38.7
Max	12564	0.236	-0.519	18.5	-40.7
Hutchinson Sue(most caudal)	13691	0.116	-0.35	13.6	-24.7

For *Plateosaurus* we conclude that the most reasonable mass estimate was about ~735 kg. Since *Plateosaurus* was roughly twice the size of *Falcarius* that means that the volume would have been off by a factor of eight. This means that *Falcarius* would have weighed ~91.875 kg, and this is within 10% of the literature value of 100 kg (Paul, 2010). This means that our method provides data that is consistent with the previously done work, and so *Falcarius* weighed around 100 kg.

Hindlimb Reconstruction

Materials and Methods:

We used the skeletal reconstruction that Hartman did for the Utah Museum of Natural History, combined with the figure he designed which included a lateral view of *Falcarius* that had likely body proportions (see figure HR.1). With this diagram we divided the limb into 23 sections perpendicular to the long axis of the limb(see figure HR.2). Next we estimated an aspect ratio [anterior-posterior to medio-lateral diameter] to seven limb intermediate sections (see figure HR.3). To determine the intermediate aspect ratios we simply used a line connecting the two values, ie if section one had an aspect ratio of 2:1.5 and section 3 had 1.5:1 then we assumed section 2 had 1.75:1.25. Now assuming that thickness is constant in each section, we can determine the volume of bone and flesh. Assuming bone has a density of 1500 kg/m^3 and muscle has a density of 1006 kg/m^3 we can then approximate mass of each section. Finally we can estimate the center of mass for each section (see fig HR.2) and use this to derive moment of inertia for the total limb.

Results:
For the derived measurements we got the following results:

Section	Aspect Ratio(A/B)	Bone diameter (unscaled)	Leg Diameter (unscaled)	Distance (unscaled)	Scaling factor (scales to meters)	BD(scaled)	LD(scaled)	Distance(scaled)	A	B
1	1.53846153846154	1.3	4	0.6	0.04273504273504	0.05555555555555	0.17094017094017	0.02564102564102	0.08547008547008	0.05555555555555
2	2.1	1	4.5	1.5		0.04273504273504	0.19230769230769	0.06410256410256	0.09615384615384	0.04578754578754
3	2.66	0.9	4.5	2.4	Thickness	0.03846153846153	0.19230769230769	0.10256410256410	0.09615384615384	0.03614806246385
4	3.22222222222222	0.8	4.3	3.4	1	0.03418803418803	0.18376068376068	0.14529914529914	0.09188034188034	0.02851458885941
5	2.75	0.8	4.1	4.4		0.03418803418803	0.17521367521367	0.18803418803418	0.08760683760683	0.03185703185703
6	2.28	0.8	3.6	5.4	Scaled thickness	0.03418803418803	0.15384615384615	0.23076923076923	0.07692307692307	0.03373819163292
7	1.81818181818182	0.8	2.7	6.3	0.04273504273504	0.03418803418803	0.11538461538461	0.26923076923076	0.05769230769230	0.03173076923076
8	1.77	1.2	3	7.5		0.05128205128205	0.12820512820512	0.32051282051282	0.06410256410256	0.03621613791105
9	1.73	0.9	3.2	8.2		0.03846153846153	0.13675213675213	0.35042735042735	0.06837606837606	0.03952373894570
10	1.687	0.7	2.8	8.8		0.02991452991453	0.11965811965812	0.37606837606837	0.05982905982906	0.03546476575522
11	1.64	0.6	2.1	9.6		0.02564102564102	0.08974358974359	0.41025641025641	0.04487179487179	0.02736085053158
12	1.6	0.6	1.9	10.4		0.02564102564102	0.08119658119658	0.44444444444444	0.04059829059829	0.02537393162393
13	1.35555555555556	0.5	1.2	11.3		0.02136752136752	0.05128205128205	0.48290598290598	0.02564102564102	0.01891551071878
14	1.11111111111111	0.5	1.2	12		0.02136752136752	0.05128205128205	0.51282051282051	0.02564102564102	0.02307692307692
15	1	0.6	1.4	12.9		0.02564102564102	0.05982905982906	0.55128205128205	0.02991452991453	0.02991452991453
16	1	0.6	1.3	14		0.02564102564102	0.05555555555555	0.59829059829059	0.02777777777777	0.02777777777777
17	1	0.6	2	15.8		0.02564102564102	0.08547008547008	0.67521367521367	0.04273504273504	0.04273504273504
18	1	0.5	1.2	16.3		0.02136752136752	0.05128205128205	0.69658119658119	0.02564102564102	0.02564102564102
19	1	0.5	1.2	17.3		0.02136752136752	0.05128205128205	0.73931623931623	0.02564102564102	0.02564102564102
20	1	0.5	1.3	18.3		0.02136752136752	0.05555555555555	0.78205128205128	0.02777777777777	0.02777777777777
21	1	0.3	0.7	19.3		0.01282051282051	0.02991452991453	0.82478632478632	0.01495726495726	0.01495726495726
22	1	0.3	0.8	19.8		0.01282051282051	0.03418803418803	0.84615384615384	0.01709401709401	0.01709401709401
23	1	0.1	0.6	19.5		0.00427350427350	0.02564102564102	0.83333333333333	0.01282051282051	0.01282051282051

Then we can figure out the inertia with a cursorial sensitivity (+- 5% mass on the foot):

Section	Bone cross section pi*radius^2	Muscle cross section pi*a*b-BCS	Mass Scaled thickness*(MCS*1006+BCS*1500)	Inertia mass*distance^2
1	0.00242406840554	0.01491734403414	0.796707295156664	0.00052380492778
2	0.00143436000328	0.01383132860308	0.68657506750535	0.00282123219717
3	0.00116183160265	0.01091946994980	0.543920263824332	0.00572171217698
4	0.00091799040210	0.00823075338435	0.412697585803909	0.00871280607037
5	0.00091799040210	0.00876785151097	0.435788214666335	0.01540810109566
6	0.00091799040210	0.00815320422918	0.40936363494496	0.02180043026334
7	0.00091799040210	0.00575106643316	0.30609224080821	0.02218715946686
8	0.00206547840472	0.00729335594889	0.445954431268322	0.04581221557243
9	0.00116183160265	0.00849008464370	0.439477459639247	0.05396753668299
10	0.00070283640160	0.00666590540944	0.331630574543409	0.04690165770443
11	0.00051636960118	0.00385702903321	0.19891989782603	0.03349027208686
12	0.00051636960118	0.00323627475740	0.172232769560836	0.03402128781449
13	0.00035859000082	0.00152371357725	0.088493199143255	0.02063645371941
14	0.00035859000082	0.00185893056425	0.10290466448166	0.02706237067236
15	0.00051636960118	0.00281134560643	0.153964447942104	0.04679162791665
16	0.00051636960118	0.00242406840554	0.137314838365538	0.04915207158968
17	0.00051636960118	0.00573744001313	0.279761498076155	0.12754704576618
18	0.00035859000082	0.00206547840472	0.111784456255828	0.05424065341261
19	0.00035859000082	0.00206547840472	0.111784456255828	0.06110009846009
20	0.00035859000082	0.00242406840554	0.127200761419315	0.07779652091408
21	0.00012909240029	0.00070283640160	0.038491111985526	0.02618444426818
22	0.00012909240029	0.00091799040210	0.047740895083618	0.03418135091785
23	0.00001434360003	0.00051636960118	0.023118940975986	0.01605482012221
	Average BCS	Average MCS	Total mass	Total Inertia
Thigh	0.00124174594569	0.01008157402067	3.59114430270976	0.07717524619818
Shank	0.00058367418595	0.00353566356646	2.46243823710496	0.53961319293813
Foot	0.00009084280020	0.00071239880163	0.348336165720273	0.21531723468242
	Total BCS	Total MCS	Total mass	Total Inertia
Complete Leg	0.01726969443952	0.12316138772389	6.40191870553499	0.83210567381874
Leg(+5% foot)	0.01728332085955	0.12326824754413	6.40738625293724	0.83592670458416
Leg(-5% foot)	0.01725606801949	0.12305452790364	6.39645115813273	0.83592670458416

As we can see the +-5% foot mass has an insignificant effect on the total mass and total inertia. An explanation of this is that the leg is not well adapted to cursory, so what might have a larger significance on a agile fast animal does not have the same impact on *Falcarius.* In addition Pontzer outlined two methods for estimating the cost of transport (COT) in extant animals, one based off of total mass, and the second based off of effective limb length (the length of the limb in the y direction when the animal is standing) (Pontzer, 2007). The regression is derived over a wide variety of extant animals to average the results. However since one is based off of the geometry of the animal, and the other is simply based off of mass, then the relationship between these two results can give us insight to cursorial adaptations of *Falcarius.* When we use the mass COT y=11.259x^-0.26 (Pontzer, 2007) with *Falcarius*' mass of 100 kg (Paul, 2010) we get that COT= 3.40, and if we use the effective limb length, y=90.284x^-0.77 we get that COT= 3.42. Now since when we take into account the geometry of *Falcarius* the COT is higher than if we just consider mass, we can infer that *Falcarius* did not have cursorial adaptations, in fact, it probably had about average movement capabilities. This means that it probably had cursorial regression from the formerly well adapted theropod group which it evolved from.

Figures:

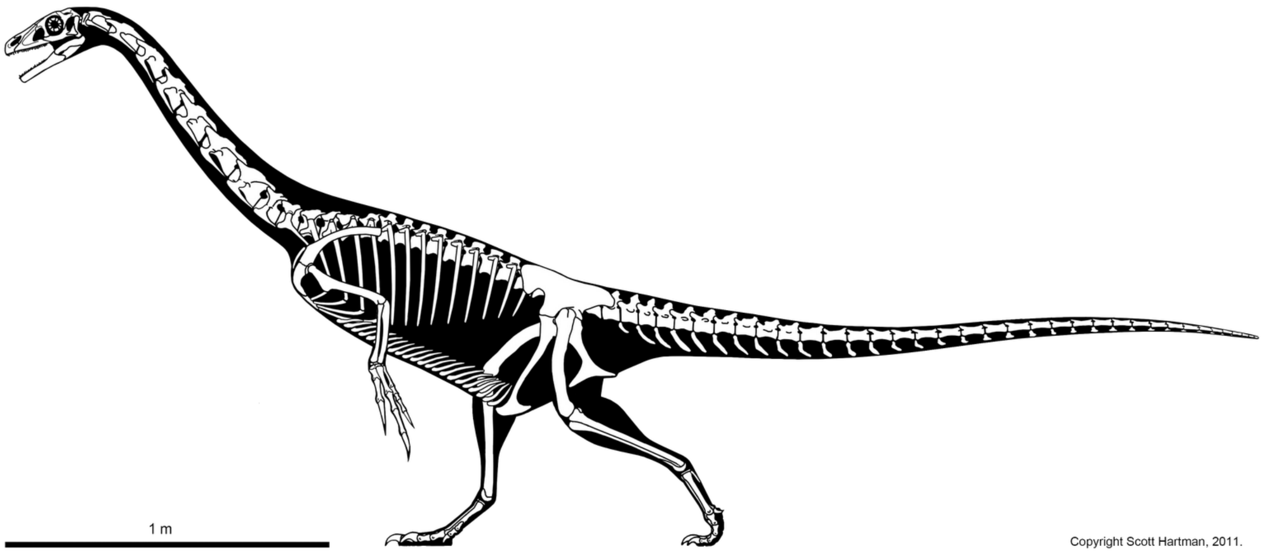


Fig HR.1, a reconstruction of *Falcarius* by Hartman done for the UMNH, available on his blog at http://skeletaldrawing.blogspot.com/2011/11/falcarius-bizarre-sickle-cutter.html

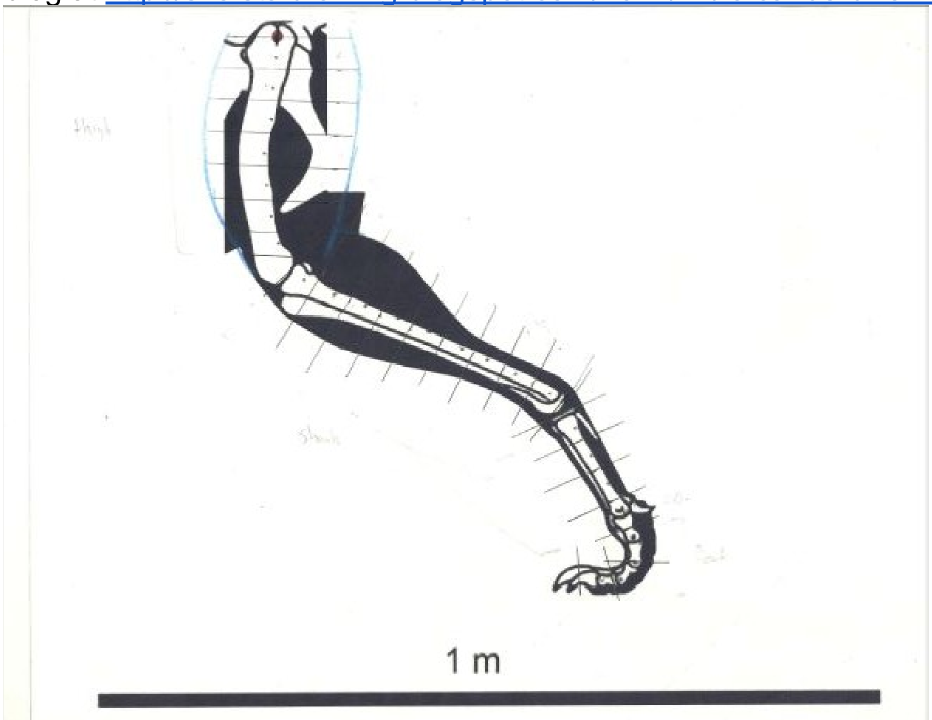


Fig HR.2, the sectioned limb. The center of mass was assumed to be at the dot.

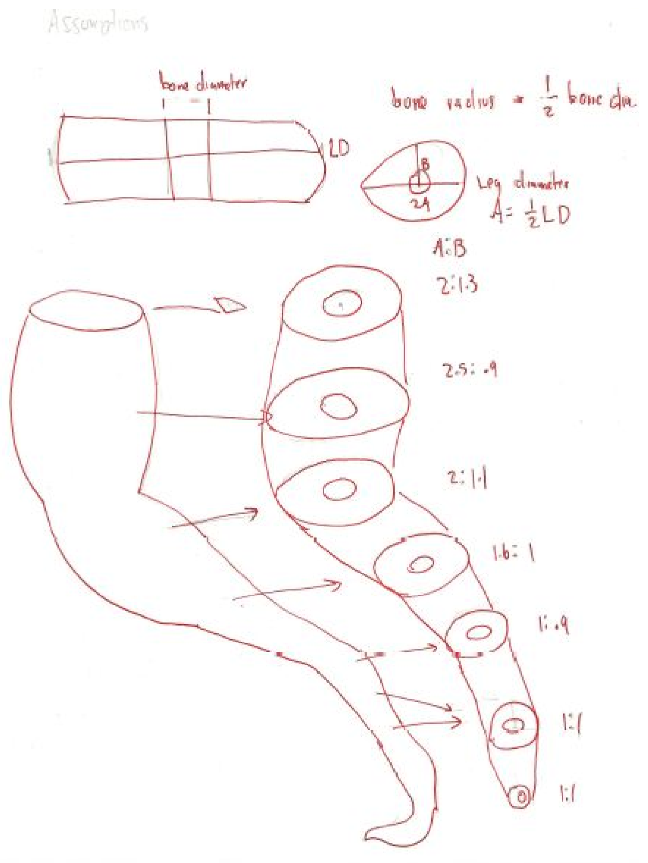


Fig HR.3, the proportions assumed for the hindlimb.

Muscle Reconstruction

Methods and Material:

To reconstruct the muscle M. caudofemoralis in *Falcarius* we note that since *Falcarius* was a non-avian dinosaur it is likely that the muscle M. caudofemoralis originated from a transverse process as well as the vertebral central, and inserted on the fourth trochanter of the femur and probably inserted on the proximal tibia. This is based off of the anatomy of extant crocodilians. (Schachner, 2011).

Based off of muscle scarring on the transverse process, we can tell that M. caudofemoralis originated from the 10th vertebra behind the pelvis (Zanno , 2010) (see figure MR.1).

Results:

As we can see, M. caudofemoralis took up less than a third of the total length of his tail (see figure MR.2) . Typically in extant reptiles the muscle extends to about half of the tail length (Persons, 2011). Considering the fact that the primary role of this muscle is as hindlimb retractor, it is likely that *Falcarius* did not have the muscle mass needed for high move speeds (Persons, 2011).

Figures:

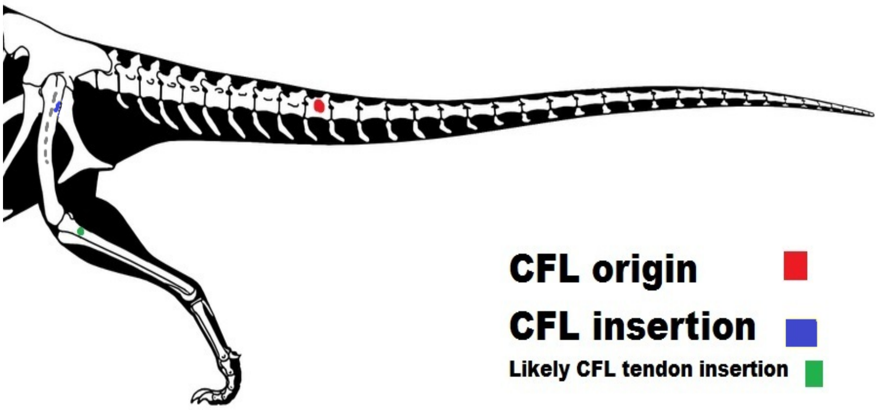


Fig MR.1. Note since the reconstruction is done from a lateral perspective, and it inserts onto the medial surface, I have added a dashed line to indicate a shift from the latter surface to the former just on the muscle map, not the reconstruction.

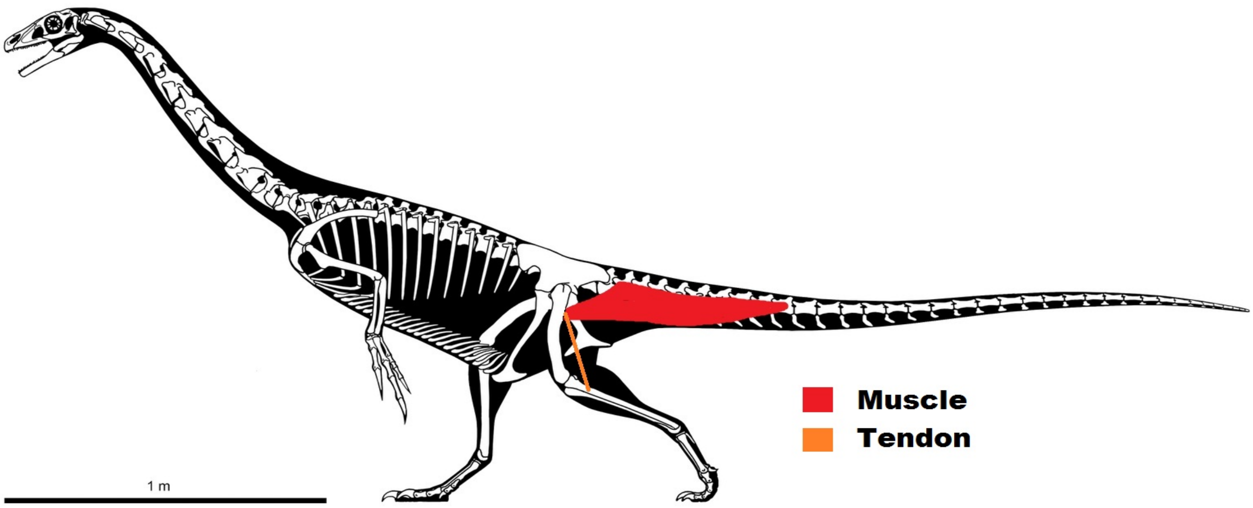


Fig MR.2 The full muscle reconstruction.

Auditory Reconstruction

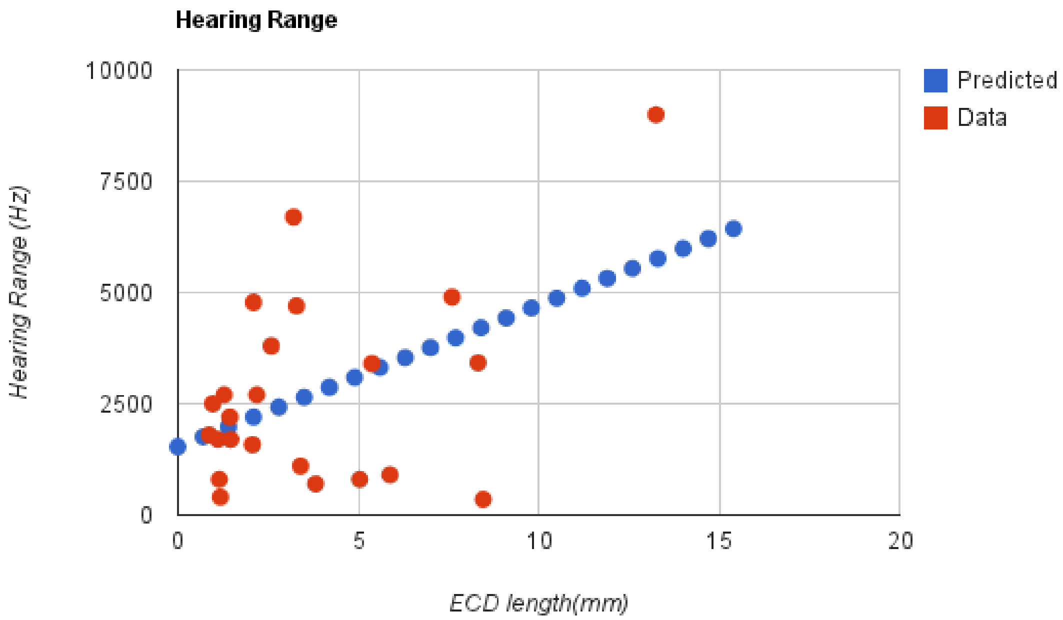
Methods and Materials:

An inner ear reconstruction was available via CT scan and using Surf Driver 8 (Smith, 2011). For exact methodology see (Smith, 2011). This reconstruction allowed us to compare the cochlear duct (see fig AR.1) in *Falcarius* to that of extant birds and reptiles to predict the mean auditory frequency range (Walsh, 2008). While Walsh outlines a method for determining auditory range based off of endosseous cochlear duct(ECD) length, he uses a normalization factor that is unclear and ill defined in that paper. However he published his original, unnormalized data. So we used a small sample of his data to create a simple linear regression, based off of ECD length and hearing range in extant reptiles and birds. For the exact materials and methods used to gather the hearing sensitivity data see (Walsh, 2008).

Results:
The data from Walsh that we used was the following:

Species	Length(mm)	Overall range (Hz)
acutus	1.28	2700
mississipiensis	5.88	900
crocodylus	1.11	1700
mydas	8.46	350
serpenta	5.04	800
punctatus	3.82	700
gecko	2.1	4780
hasselquistii	1.44	2200
caudicinctus	2.07	1580
scicula	0.97	2500
rugosa	2.59	3800
occelatus	1.46	1700
niloticus	3.4	1100
wislizenii	1.18	400
sagrei	0.87	1800
hardwickii	2.19	2700
zarudnyi	1.15	800
alba	13.25	9000
undulatus	3.21	6700
novaehollandiae	8.32	3420
demersus	5.38	3400
sp.	7.6	4900
guttata	3.29	4700

Our regression looks like the following:



When we use the ECD length of 10.1 for Falcarius (Fig AR.1) (Smith, 2011) we get a predicted hearing range around 4750 Hz. This ECD length is a lot higher than in other similar theropods, such as Allosaurus and Ceratosaurus (Smith,2011). This suggests that *Falcarius* could have heard a large range of frequency which may have included sounds made by juveniles, prey, other small bodies, and adults (Smith, 2011). In addition, there is a positive correlation between hearing range frequency and sociability, so there a chance that *Falcarius* lived in larger aggregates (Walsh, 2008). While this data could imply either predatory behavior, since a larger frequency range could aid hunting behaviors, or herbivorous behavior, since it would enable hearing as a defence mechanism, it implies more of a specific mechanism on how either would work. Moreover, it could likely imply either a pack or herd structure.

Figures:

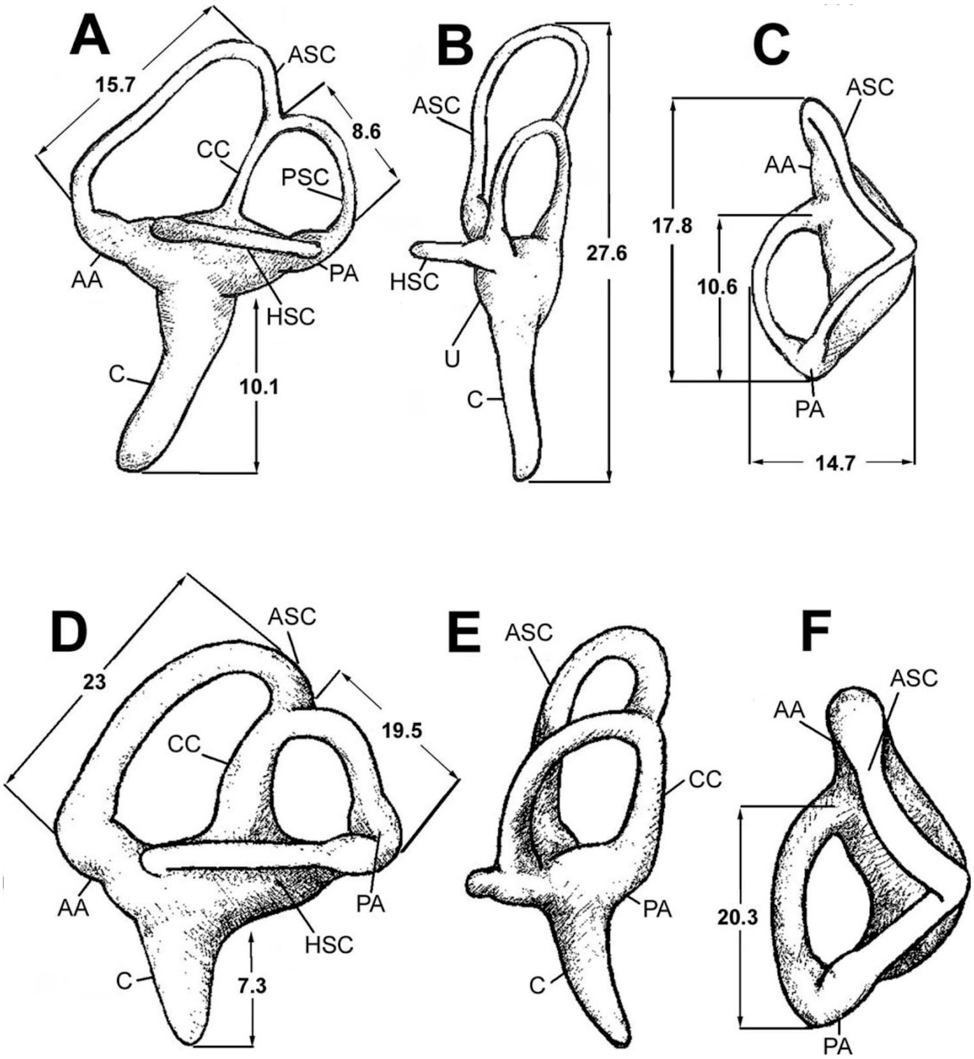


Fig AR.1. *Falcarius* courtesy of (Smith, 2011)
Anatomical Abbreviations—AA, anterior ampulla; **ASC**, anterior semicircular canal;**C**, cochlear duct; **CC,** common crus;**HSC,** horizontal semicircular canal;**PA,** posterior ampulla;**PSC,** posterior semicircular canal;**U,** utricle;

Cardiovascular and Respiratory reconstruction

Materials and Methods:

(Blaylock, 2000) outlines many allometric curves for various cardiovascular and respiratory quantities. These include the stroke volume of the heart (SV), amount of oxygen used (VO2), heart rate (HR), and heart mass (using the method of Laplace for stress) (Blaylock, 2000). For the exact methods used see (Blaylock,2000). We used these quantities, along with some endothermic and endothermic assumptions to model the mean arterial pressure (MAP), the oxygen extraction rate, and the heart mass(assuming a spherical heart with thin walls). In addition, we ran a sensitivity analysis by changing the mass +-15%.

Our assumptions were that endotherms have an intracranial pressure of 90 mmHg and ectotherms have an intracranial pressure of 40 mmHg. With this information, the distance between the center of the heart and the center of the head (determined to be 1.17m for *Falcarius*), and the density of blood (assumed to be 1055 kg/m^3), we can determine MAP. To determine this, we used conservation of energy applied to pressure, and the assumption that MAP is uniform throughout the body. Then we get that MAP= ICP+p*h*g, where ICP is the inter-cranial pressure, p is the density of blood, h is the distance from the heart to the head, and g is the acceleration due to gravity.

To determine oxygen extraction (Ca-Cv, where Ca is the concentration of oxygen in arterial blood, and Cv for venous blood) we used Fick's principle, which is (Ca-Cv)*SV*HR=VO2, and the allometric data for HR, SV and VO2 (Blaylock,2000).

Finally, to estimate heart volume we assumed that the the heart was a sphere with the inner radius, r, and thickness, t. Then we assumed that the maximum volume was 1.5*SV. Using this we can determine radius. Then using a thin wall assumption (stress= pressure*radius/ (2*thickness)) , and assuming exterior stress is 20kPa, we can determine thickness. Then with heart volume we can determine the mass, assuming the density of muscle= 1006 kg/m^3.

Results:

We got the following results for VO2 (mL/min)

Mass(kg)	Bird	Mammal	Reptile
85	276.86	339.48	38.4
100	311.23	384.11	43.3
115	344.17	427.16	48.0

For endotherms we got that MAP= 180.74 mmHg, and for ectotherms MAP= 130.74

For oxygen extraction (unitless) we got the following ranges over 85-100 kg

Bird	Mammal	Reptile
.04	.0487-.0488	.06496-.0654

Then interesting results was that when we substituted the allometric curves into Fick's equation we get that the oxygen extraction for birds does not scale with mass, for mammals its proportional for m^-0.01 and for reptiles m^-0.02. This could be a result of range of masses of the tested animals, or different physiologies (difference in avians lungs against mammalian lungs).

For Heart Rate (beats/minute)

Mass(kg)	Birds	Mammal	Reptile
85	57.6	81.7	6.9
100	55.1	77.7	6.64
115	53.0	76.2	6.41

Stroke Volume (mL)

Mass (kg)	Birds	Mammal	Reptile
85	119	85	85

100	140	100	100
115	160	115	115

Heart mass/ Total Mass (as a %) using thin wall assumption

Mass (kg)	Bird(endo)	Bird(ecto)	Mammal(Endo)	Mammal(ecto)	Reptile(endo)	Reptile (ecto)
85	0.66	0.41	0.47	0.29	0.47	0.29
100	0.66	0.42	0.47	0.30	0.47	0.30
115	0.64	0.42	0.47	0.30	0.47	0.30

Heart mass/ Total mass (as a %) with the method of Laplace (Blaylock,2000).

Mass(kg)	Bird	Mammal	Reptile
85	0.542	0.527	-
100	0.534	0.53	-

115	0.527	0.537	-

Oxygen levels present in the atmosphere 125 million years ago would have been about 2-3% higher than today's level (Berner,1999). This means that there is a high likelihood that the lungs needed at that period would have been comparable to that in extant animals, in terms of membrane thickness and volume, possibly marginally less since oxygen extraction would have been easier. Since these level are fairly similar, the data generated by the allometric curves in extant animals should be viable for *Falcarius.*

The data generated seems to support two possible variations of cardiovascular and respiratory physiologies. The first kind would support high amount of activity (modeling *Falcarius* with either mammalian or avian mechanics). Either assumption would make make the heart mass approximately equal (using the method of Laplace). To provide adequate oxygen supply using mammalian mechanics the internal lung surface area would have had to been ~25% larger, or had ~25% thinner membranes, or some combination of the two. In addition to support a higher VO2 max, the heart rate would need to be ~33% larger in mammalian mechanics than in avian mechanics. The avian hearts would take up a significantly more volume than mammalian heart, but it would have a larger stroke volume.

It’s likely that both hearts would have had contained a divided ventricle, since it is present in both extant birds and mammals, both of which have the highest VO2 outputs of extant animals.

The second variation would support low levels of activity (modeling *Falcarius* with reptilian mechanism). The lungs would have need to be able to supply ~50% more than the avian lung, which would probably require both thinner membranes and an increase in internal surface area. The high levels of oxygen would have helped facilitate this, and dampened the needed volume. However the heart would have needed to be significantly smaller to supply the required amount of oxygen and pressure. This means that there is a good chance that the heart would have only needed one ventricle.

However there is little evidence that *Falcarius,* and theropods in general, would have had avian lungs (Quick, 2007). In addition, it was likely that theropods had a cardiovascular system similar to crocodilians (Quick, 2007). This means that *Falcarius* would have had a four-chambered heart or a five-chambered heart, which would have allowed oxygenated blood and deoxygenated blood to mix (Quick, 2007). This implies that *Falcarius* would have had the latter discussed variation, that supported low level of activity, as seen in extant reptiles.

Bite force Estimation

Materials and methods:
A A Z&F 5600i laser scanner was used to digitize a juvenile skull of *Alligator mississippiensis.* To investigate bite force we used the MDA package GaitSym. Upper and lower jaw segments were connected with one hinge. We reconstructed the basic muscle system, with the physical cross sectional area outlined in (Porro,2011). Finally

we tested the bite force generated at the end of the snout in our model, as well as where the teeth start to generate a range of maximal bite force.

Results:

Our results were that the bite force of a juvenile *Alligator mississippiensis* could range from 150- 250N , based on the position where bite force is measured. This is consistent with results done in similar experiments, see (Bates,2012). *Falcarius*' snout length was approximately ~0.25 m , while the snout length of our skull model was ~0.2 m. Since they are approximately the length, we can assume that the muscle cross sectional area is roughly the same, and that *Falcarius* would have a bite force that is a 25% increase of that of a juvenile alligator. This would put the range of *Falcarius'* bite force to be 190-310 N. For comparison, the human bite force range is 700-1020N (Bates, 2012) .

While this does not rule out *Falcarius* as primarily a predator, it does eliminate a biting mechanism as a potential predatory mechanism.

Conclusion

Many of the traits that were analyzed in *Falcarius* allude to it being herbivorous. While no single trait that was analyzed excludes predatory behavior, none of them suggest it. The hindlimb reconstruction along with the M. caudofemoralis reconstruction both support the hypothesis that this dinosaur was not capable of high running speeds. This fact in symphony with the cardiovascular and respiratory reconstruction, suggests a slow moving, lethargic dinosaur. Furthermore, the low bite force suggests no predatory behavior short of scavenging, since there would be no viable way for *Falcarius* to hunt or injure prey. Finally it was highly unlikely that it scavenged, due to the extremely well adapted inner ear structure. While a large hearing range could facilitate in the survival of a slow scavenger, based off of the other traits it was likely that the structure supported a social structure. Ultimately, it seems likely that *Falcarius* was primarily herbivorous and lived in a large social aggregate.

References

Bates, Karl T., Falkingham, Peter L., Breithaupt, Brent H., Hodgetts, David, Sellers, William I., and Manning, Phillip L., 2009. How Big Was 'Big Al'? Quantifying the effect of soft tissue and osteological unknowns on mass predictions for Allosaurus (Dinosauria:Theropoda). Palaeontologia Electronica Vol. 12, Issue 3; 14A: 33p;
http://palaeo-electronica.org/2009_3/186/index.html

K. T. Bates and P. L. Falkingham. Estimating maximum bite performance in *Tyrannosaurus rex* using multi-body dynamics Biol Lett 2012 : rsbl.2012.0056v1-rsbl20120056.

Robert A. Berner .Atmospheric oxygen over Phanerozoic time. PNAS 1999 96: 10955-10957.

Brøndum, E. Jugular venous pooling during lowering of the head affects blood pressure of the anesthetized giraffe. American journal of physiology. Regulatory, integrative and comparative physiology 297.4 (2009):R1058.

Hutchinson JR, Bates KT, Molnar J, Allen V, Makovicky PJ (2011) A Computational Analysis of Limb and Body Dimensions in Tyrannosaurus rex with
Implications for Locomotion, Ontogeny, and Growth. PLoS ONE 6(10): e26037. doi:10.1371/journal.pone.0026037

Falkingham, Peter L. 2012. Acquisition of high resolution 3D models using free, open-source, photogrammetric software.
Palaeontologia Electronica Vol. 15, Issue 1; 1T:15p;

Kirkland, James I. A primitive therizinosauroid dinosaur from the Early Cretaceous of Utah. Nature 435.7038 (2005):84.

Paul, G.S., 2010, *The Princeton Field Guide to Dinosaurs*, Princeton University Press p. 156

Persons, W S. "The tail of Tyrannosaurus: reassessing the size and locomotive importance of the M. caudofemoralis in non-avian theropods." The anatomical record 294.1 (2011):119.

Schachner, Emma R. "Pelvic and hindlimb myology of the basal Archosaur Poposaurus gracilis (Archosauria: Poposauroidea)." Journal of morphology 272.12 (2011):1464.

Seymour R, Blaylock A. The principle of Laplace and Scaling of Ventricular Wall Stress and Blood Pressure in Mammals and Birds. Physiological and Biochemical Zoology, Vol. 73, No. 4 (July/August 2000), pp. 389-405

Smith, David K. "New Information on the Braincase of the North American Therizinosaurian (Theropoda, Maniraptora) Falcarius utahensis." Journal of Vertebrate Paleontology 31.2 (2011):387-404.

Porro, L. B., Holliday, C. M., Anapol, F., Ontiveros, L. C., Ontiveros, L. T. & Ross, C. F. 2011 Free body analysis, beam mechanics and finite element modelling of the mandible of Alligator mississippiensis. J. Morph. 272, 910–937. (doi:10.1002/jmor.10957)

Quick, Devon E. "Cardioa pulmonary anatomy in theropod dinosaurs: Implications from extant archosaurs." Journal of morphology 270.10 (2009):1232-1246.

Walsh, S., P. Barrett, A. Milner, G. Manley, and L. Witmer. 2008. Inner ear anatomy is a proxy for deducing inner ear capability and behaviour in reptiles and birds. *Proceedings of the Royal Society B*1390:1–6.

Zanno.L.E 2010 .Osteology of Falcarius utahensis (Dinosauria: Theropoda): characterizing the anatomy of basal therizinosaurs. Zool. J. Linn. Soc. :196-230.

Zanno, L. E. The pectoral girdle and forelimb of the primitive therizinosauroid *Falcarius utahensis* (Theropoda, Maniraptora): analyzing evolutionary trends within Therizinosauroidea. 2006. *Journal of Vertebrate Paleontology* 26:636-650.

Zanno, Lindsay E. A new North American therizinosaurid and the role of herbivory in 'predatory' dinosaur evolution. Proceedings - Royal Society. Biological sciences 276.1672 (2009):3505.

Installation appendix

Instructions(to get python, but **just** python, dont download the osm bundler):
http://code.google.com/p/osm-bundler/wiki/InstallingRunningApplication
Downloads(the one we want is osm-bundler-pmvs2-full-32-64, and the example photos if you want to test it out on those):
http://code.google.com/p/osm-bundler/downloads/list
Getting python to work in the command promt http://showmedo.com/videotutorials/video?name=960000&fromSeriesID=96

Got meshlab
http://sourceforge.net/projects/meshlab/files/

Things we need first
-python (you need version 2.7)
- the PIL library
- the osm-bundler-pmvs2-full-32-64
- photos to test

First things first
- downloading and install python, then the PIL library, this should be straightforward(provided by the link above)
- unzip the download and put them in a convenient spot, you want the photos in the osm-bundler file under the osm-bundlerWinXX subfolder (for me XX=64, because im on a 64 bit machine, if you dont know this, its probably 32 if its older, 64 if its newer and nice, 32 if its newer and not so good)
- now that folder should have osmbundler, osmcmvs, osmpmvs, software, testPhotos (this is what i called the folder of my photos), and three python scripts RunBundler.py, RunCmvs.py, RunPMVS.py
- goto start menu and type in cmd
- find the directory you unzipped the osm-bundler software
for me its “C:\Users\Geoff\Desktop\photogrammetry\osm-bundler-pmvs2-cmvs-full-32-64\osm-bundlerWin64”
We need to tell the computer where our python scripts and photos are located at
so enter the command
cd C:\Users\Geoff\Desktop\photogrammetry\osm-bundler-pmvs2-cmvs-full-32-64\osm-bundlerWin64

Now we need to tell computer to run our python scripts
First try typing in python and hitting enter, if it doesnt pop up with the version then you need to follow the link above to get python to work in the cmd

If it changes the line entry from C:\ourDirectory to >>>> then just hit control+z, then hit enter and it should return you
now that python works, we need to tell the computer to run out python script in python
so enter the command
python RunBundler.py --photos=testPhotos (or whatever folder you called it)

Let that run, then a folder should pop up, and we need to get the file directory of that folder
now we need to run the second python script
For this example we are going to use the PMVS script, for many photos you change it to the CMVS, but I currently haven't tested how this works

Now we need to run the second script on the output that it just created so we do the following
python RunPMVS.py --bundlerOutputPath=C:\User\Geoff\AppData\Local\Temp\osm-bundler-wdjsxc

Now in MeshLab, go to importMesh, go to that long folder in the temp data, go to pmvs, models, and the file that is in there, and voila!

Note: If it doesnt support your camera type, then you will have to edit it with this
http://sourceforge.net/projects/sqlitebrowser/

Then open that, and open the file osm-bundler-pmvs2-cmvs-32-64/osm-BundlerWin64/osmbundler/cameras

Then change one of the cameras to reflect your camera, here is a list of stats that i found
https://svn.personalrobotics.ri.cmu.edu/public/trunk/src/moped/BundlerPy/bin/extract_focal.pl

Figure 6.1: A placeholder image for [14]

Bibliography

[1] Harold Abelson, Gerald Jay Sussman, and Julie Sussman. *Structure and Interpretation of Computer Programs*. MIT Press, 2nd edition, 1996. Commonly referred to as "The Wizard Book".

[2] Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman. *Compilers: Principles, Techniques, and Tools*. Pearson, 2nd edition, 2006. Commonly referred to as "The Dragon Book".

[3] Geoffrey Bradway and Amanda Ngo. *Principia Mathematica II: Electric Bugaloo*. https://www.archive.org, 2025.

[4] Lee Stemkoski Dominic Klyve. The euler archive, 2025. Accessed: 2025-01-12.

[5] Bhante Henepola G. *Bhavana Society*.

[6] Haleh Liza Gafori. *Gold: Rumi*. New York Review Books, 2022.

[7] John Green. *The Anthropocene Reviewed: Essays on a Human-Centered Planet*. Dutton, 2021. A collection of essays reviewing facets of the human experience in the Anthropocene era.

[8] Bhante Henepola Gunaratana. *Mindfulness in Plain English*. Wisdom Publications, 1991.

[9] Davar Khoshnevisan. *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2007. Hardback Edition.

[10] Isaac Newton. *Philosophiae Naturalis Principia Mathematica*. Royal Society of London, 1687. Translated by I. Bernard Cohen and Anne Whitman in 1999.

[11] Massachusetts Institute of Technology. Mit - massachusetts institute of technology, 2025. Accessed: 2025-01-12.

[12] University of Utah. *Utah Tea Pot*.

[13] University of Utah. University of utah, 2025. Accessed: 2025-01-12.

[14] Neri Oxman. Vespers: Series i-v (the death masks), 2016. Exhibited at the Museum of Modern Art (MoMA), New York, NY.

[15] Tathagata Meditation Center. Tathagata Meditation Center, n.d. Accessed: 2025-01-12.

[16] Yale University. Yale university, 2025. Accessed: 2025-01-12.

[17] Berkeley University of California. Uc berkeley - university of california, berkeley, 2025. Accessed: 2025-01-12.

[18] Vetro Editions. Vetro Editions, n.d. Accessed: 2025-01-12.

Appendix A

Dedication outro.

```
 By means of our meritorious deeds,

May the Suffering be free from Suffering
May the Fear-Struck be free from Fear
May the Grieving be free from Grief
So too may all Being-Be.
From the highest realms of existence to the lowest,
May all being arisen in these realms,
With form and without form,
With perception and without perception,
Be released from all Suffering,
And attain to Perfect Peace
May all being be free from suffering

Sadu!  Sadu!  Sadu!
```

Figure A.1: A Utah Teapot [12]

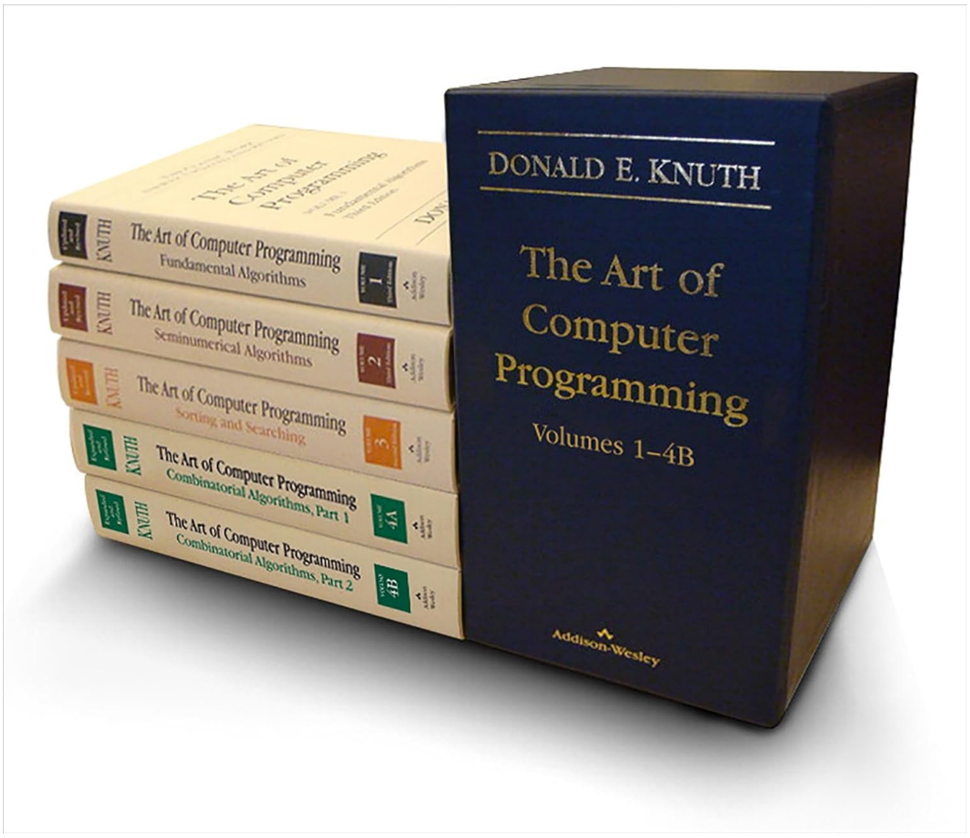


Figure A.2: It can't be that hard, can it? Did you try reading the manuel? [4]

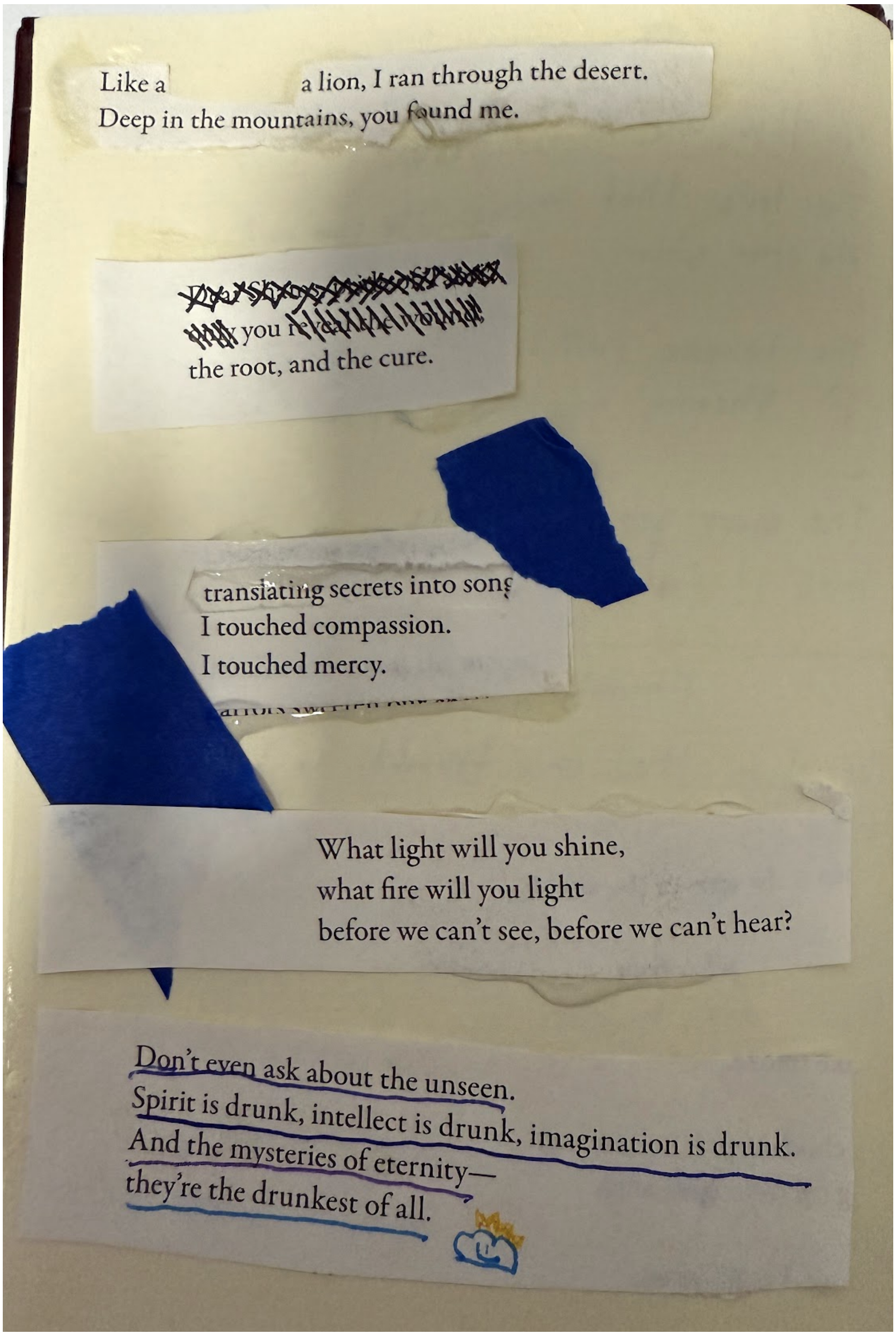

Like a a lion, I ran through the desert.
Deep in the mountains, you found me.
[redacted]
[redacted] you [redacted]
the root, and the cure.
translating secrets into song
I touched compassion.
I touched mercy.
What light will you shine,
what fire will you light
before we can't see, before we can't hear?
Don't even ask about the unseen.
Spirit is drunk, intellect is drunk, imagination is drunk.
And the mysteries of eternity—
they're the drunkest of all.

Figure A.3: [6]