

Prozedurale Generierung von Wirbeltierskeletten

Masterarbeit von

Nina Zimbel

An der Fakultät für Informatik
Institut für Visualisierung und Datenanalyse,
Lehrstuhl für Computergrafik

4. Juni 2020

Erstgutachter: Prof. Dr.-Ing. Carsten Dachsbacher
Zweitgutachter: Prof. Dr. Hartmut Prautzsch
Betreuernder Mitarbeiter: Dr. Johannes Schudeiske

Inhaltsverzeichnis

1. Einleitung	1
2. Biologie der Wirbeltiere	3
2.1. Das Skelett	4
2.2. Von Groß und Klein	5
2.3. Mythologie	6
3. Grundlagen	8
3.1. Prozedurale Generierung	8
3.2. Animation von Figuren	9
3.3. Inverse Kinematik	9
3.4. Grammatiken	9
3.5. Principal Component Analysis	10
3.6. Quantil-Quantil-Diagramme	11
3.7. Genetische Algorithmen	12
4. Analyse von Skelettdaten mit Hilfe von PCA	13
4.1. Haltung des Skeletts	13
4.2. Datenerhebung	14
4.3. Analyse der Eingabedaten	16
4.4. Analyse der Ergebnisse	21
4.5. Bedingte Verteilungen	25
5. Generierung von Skeletten	27
5.1. Überblick über den Ablauf der Generierung	27
5.2. Bestandteile eines Skeletts	28
5.3. Wirbel und Rippen	29
5.4. Aufbau als Grammatik	30
5.5. Extremitäten	33
5.5.1. Berechnung der Bodenhöhe	37
5.5.2. Algorithmus zur Ausrichtung der Beine	37
5.5.3. Zusätzliche Ansatzpunkte für Extremitäten	40
5.6. Knochenmodelle	42
6. Zusätzliche Funktionen	46
6.1. Benutzeroberfläche	46
6.2. Speichern und Laden von Skeletten	48
6.3. Erzeugung von Variationen	49
7. Implementierungsdetails	51
7.1. Aufbau der Software	51
7.2. Dateiformate für 3D-Modelle	51
7.3. Transformationsmatrizen	52

7.4. PCA	53
7.4.1. Annotation der Bilder	53
7.4.2. Anpassung des Verlaufs der Wirbelsäule	55
7.5. Probleme der Positionierung von kurzen Beinen	56
8. Fazit und Ausblick	58
Literatur	60
Anhang A. PCA Daten	64
A.1. Eingabedaten für PCA	64
A.1.1. Bilder	64
A.1.2. Gewichte	65

1. Einleitung

Wirbeltiere sind allgegenwärtig. Sie sind nicht nur im alltäglichen Leben anzutreffen, sondern auch in Computerspielen und Filmen. In der Produktion dieser Medien besteht also ein stetiger Bedarf an 3D-Modellen von Wirbeltieren. Trotzdem gibt es nur wenige Algorithmen, die die KünstlerInnen bei dieser zeitintensiven Arbeit unterstützen [9].

Hier wird ein Algorithmus zur Generierung abstrakter, aber dennoch wirklichkeitsnäher, 3D-Modelle von Skeletten vorgestellt, die die Grundlage für die Modellierung eines kompletten Tieres bilden können. Viele andere Ansätze generieren hingegen zunächst 3D-Modelle von der „Außenhaut“ der Tiere und passen dann ein sehr abstraktes Skelett (Rig) für die Animation ein (siehe Abschnitt 3.1). Dieses Rig hat nicht viel mit einem echten Wirbeltierskelett zu tun und ist nur für die Animation gedacht.

Diese Arbeit konzentriert sich auf die Generierung von relativ wirklichkeitsnahen Skeletten, die dann unterschiedlich weiterverwendet werden können. Einerseits können die Skelette als Inspiration dienen. Sie können einen Anhaltspunkt zu Körperform, Proportionen und zu möglichen Bewegungsabläufen bieten und es kann schnell eine große Auswahl von ihnen generiert werden. Andererseits können die Skelette eine Grundlage für sehr realistische Modelle sein. Diese Modelle simulieren zusätzlich zum Skelett auch Muskeln und Haut und benötigen deshalb relativ detaillierte Skelettmodelle.

In beiden Fällen darf das Modell nicht zu abstrakt sein, da es sonst zu wenig Informationen zum Aufbau des konkreten Tiers liefert. Ziel ist es also ein Skelett zu generieren, welches genug Knochen enthält um realistisch zu wirken, aber auch nicht zu viele um den Aufwand für die Generierung und die Programmierung des Algorithmus im Rahmen zu halten.

Der hier vorgestellte Algorithmus beschränkt sich auf die Generierung von Wirbeltierskeletten, da deren Aufbau nicht sehr variiert. Natürlich treten trotzdem mehr Unterschiede hervor, je detaillierter die Skelette betrachtet werden. Deshalb werden abstrahierte Skelette mit vereinfachten und in der Anzahl reduzierter Knochen generiert. Die Grundlagen zur Biologie der Wirbeltiere, die für diese Arbeit notwendig sind, werden in Kapitel 2 vorgestellt, die technischen Grundlagen in Kapitel 3.

Im Folgenden werden die wichtigsten Schritte des Algorithmus kurz angerissen, um den Leser durch den Aufbau der Arbeit zu führen. Ein Überblick über den Ablauf des Algorithmus ist in Abschnitt 5.1 zu finden.

Die Datengrundlage für den Algorithmus schafft eine *Principal Component Analysis* (Hauptkomponentenanalyse) auf annotierten 2D-Skelettbildern (Kapitel 4). Ohne sie wäre es schwer Skelette in einer natürlichen und stabilen Haltung zu generieren. Die Knochen des

Skeletts werden dann mit Hilfe einer kontextfreien Grammatik generiert und gleichzeitig angeordnet (Kapitel 5). Zum Schluss wird aus existierenden 3D-Modellen von einzelnen Knochen ein 3D-Modell des generierten Skeletts erstellt (Abschnitt 5.6)).

Prinzipiell generiert der Algorithmus zufällige Skelette. Es können aber auch Benutzereingaben, z. B. zur Anzahl der Extremitäten, berücksichtigt oder Variationen zu schon bestehenden Skeletten generiert werden (Kapitel 6).

Zusätzliche Informationen zu Implementierungsdetails sind in Kapitel 7 zu finden und abgerundet wird die Arbeit mit Fazit und Ausblick in Kapitel 8.

2. Biologie der Wirbeltiere

„Nichts anderes in der Natur hat eine herrlichere Struktur als der Körper der Wirbeltiere.“

— „Vergleichende und funktionelle Anatomie der Wirbeltiere“ [22, S. 1]

Dieses Kapitel soll einen Überblick über die biologische Beschaffenheit der Wirbeltiere geben. Da dies keine biologische Arbeit ist, werden nur diejenigen Merkmale und Eigenschaften weiter ausgeführt, die im Folgenden zum Verständnis nötig sind.

Wirbeltiere, oder auch Vertebrata, sind Tiere mit einer skelettartigen Schädelkapsel, einem Cranium. Deshalb werden sie auch Schädeltiere oder Craniota genannt.

Auf den ersten Blick ist diese Definition etwas überraschend, da als Haupteigenschaft nicht die Wirbelsäule, sondern der Schädel genannt wird. Das liegt daran, dass zu den Wirbeltieren auch einige Tiere gezählt werden, die gar keine Wirbelsäule sondern „nur“ eine Chorda dorsalis besitzen. Die Chorda dorsalis besteht aus weichem Gewebe, nicht aus Knochen. Sie ist das Achsenskelett der Chordatiere, einem Tierstamm, zu dem auch die Wirbeltiere gehören. [22, S. 27 f.]

Hier in dieser Arbeit soll es aber nur um Wirbeltiere mit Wirbelsäule gehen.

Es gibt fünf große Gruppen von Wirbeltieren. Die historisch ersten Wirbeltiere sind die Fische. Danach entwickelten sich die Tetrapoden, Wirbeltiere mit vier Gliedmaßen, zu denen alle anderen Gruppen gehören. Zunächst entstanden die Amphibien, die noch nicht vollkommen terrestrisch leben. Daraus entwickelten sich die Reptilien, die erste Klasse der Wirbeltiere, die alle Strukturen besitzt um vollkommen an Land zu leben. (Es gibt aber auch Reptilien die trotzdem wieder im Wasser leben.) Außerdem gibt es noch die Vögel, die sich auf das Fliegen spezialisiert haben, und die Säugetiere, die ihre Jungen säugen. [22, Kapitel 4]

Auch der Mensch ist ein Säugetier und besitzt eine Wirbelsäule. Da sich Wirbeltiere, vor allem in ihrem Skelett, sehr ähneln, lassen sich viele der im folgenden Abschnitt aufgeführten Eigenschaften des Skeletts, vermutlich leicht „am eigenen Leib“ nachvollziehen.

2.1. Das Skelett

„Das innere, gelenkige Skelettsystem der Vertebraten ist einzigartig im Tierreich. Es ist das wichtigste aller Organsysteme für das Studium der Wirbeltiermorphologie.“ — [22, S. 131]

Wirbeltiere haben ein inneres Skelett und ihr Körper ist bilateralsymmetrisch [22, S. 27]. Das bedeutet, dass ihre linke Körperseite symmetrisch zur rechten ist.

Wirbelsäule

Aufbau und Form der Wirbelsäule hängen von der Lebensweise des entsprechenden Wirbeltiers ab. Ein wesentlicher Einflussfaktor sind die Kräfte, die auf die Wirbelsäule wirken. Bei Fischen muss die Wirbelsäule dem Druck der starken Axialmuskeln, die seitlich am Körper entlang führen, entgegenwirken. Bei Tetrapoden sind diese Axialmuskeln zurückgebildet, dafür muss die Wirbelsäule aber der Schwerkraft widerstehen. Und bei Vögeln ist die Wirbelsäule speziell an den Flug angepasst und deshalb u. a. besonders leicht. [22, Abschnitt 9.2, S. 168 ff.]

Auch die Anzahl der Wirbel unterscheidet sich teilweise erheblich. Die Halswirbelsäule ist z. B. bei den Fischen noch gar nicht herausgebildet. Amphibien haben einen Wirbel, der für die Beweglichkeit des Kopfes zuständig ist. Bei Reptilien ist die Halsregion meist schon stärker abgesetzt. Säugetiere haben, bis auf wenige Ausnahmen, genau 7 Halswirbel, egal wie lang der Hals ist. Vögel haben die meisten Halswirbel, nämlich 10 bis maximal 31 beim Trauerschwan [48], meistens aber zwischen 15 und 20. [22, Abschnitt 9.2]

Ebenso unterscheidet sich die Anzahl der Wirbel in anderen Teilen der Wirbelsäule. Teilweise sind Wirbel sogar miteinander zu einem größeren Knochen verwachsen. Betrachtet man nur die Gruppe der Säugetiere, so kommt man auf etwa 12 bis 14 Brustwirbel, 5 bis 7 Lendenwirbel, 3 bis 5 Kreuzwirbel und 4 bis 22 Schwanzwirbel [15].

Auch die Form der einzelnen Wirbel unterscheidet sich sowohl zwischen den verschiedenen Tieren als auch entlang der Wirbelsäule einer einzelnen Art [22, Abschnitt 9.1 und Abbildung 9.2]. Um diese Vielfalt zu abstrahieren wird hier mit Platzhaltern gearbeitet, die für alle Arten von Wirbeln stehen. Falls später eine genauere Unterscheidung nötig sein sollte, können diese leicht spezialisiert werden.

Extremitäten

Die Extremitäten der Tetrapoden sind nach einem „Grundbauplan“ aufgebaut (siehe Abbildung 2.1a). Dieser wird jedoch vielfach abgewandelt. Einige Beispiele sind in Abbildung 2.1b zu sehen. [18, S. 487]

Bei Fischen haben die Extremitäten keine Verbindung zur Wirbelsäule. Erst Amphibien bilden Schulter- und Beckengürtel aus, da sie zur Fortbewegung an Land nötig sind. [22, Abschnitt 9.2.3] Der Einfachheit halber wird hier aber nicht weiter beachtet ob eine Verbindung zur Wirbelsäule besteht, oder nicht.

Schulter- und Beckengürtel bestehen jeweils aus mehreren Knochen, die je nach Art mehr oder weniger ausgebildet, unterschiedlich geformt oder sogar komplett zurückgebildet sein können [22, Absatz 9.7]. Um das im Folgenden zu vereinfachen, wird jeder Extremitätengürtel durch einen Knochen repräsentiert. Der Schultergürtel wird durch ein Schulterblatt ersetzt und der Beckengürtel durch einen „Beckenknochen“.

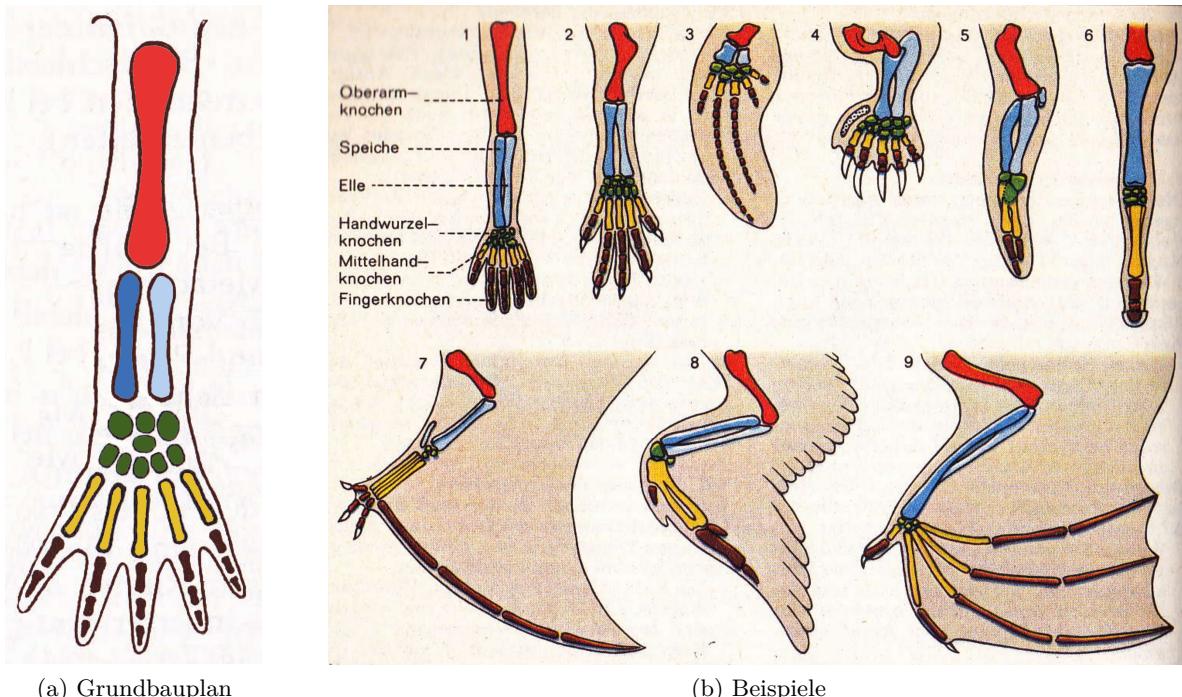


Abbildung 2.1.: Extremitäten von Tetrapoden, Oberarm rot, Speiche dunkelblau, Elle hellblau, Handwurzelknochen grün, Mittelhandknochen gelb, Finderknöchen braun
 (a) Grundbauplan [18, S. 487, vereinfacht und eingefärbt],
 (b) Beispiele für Vordergliedmaßen von Mensch (1), Eidechse (2), Wal (3), Maulwurf (4), Pinguin (5), Pferd (6), Flugsaurier (7), Vogel (8) und Fledermaus (9). [47, S. 474]

Abstraktion

Das Skelett von Wirbeltieren ist im Allgemeinen also ein kompliziertes Konstrukt aus vielen Einzelteilen, die zwischen verschiedenen Arten stark variieren. Um einen Algorithmus zu entwerfen, der Skelette nachempfundene Gebilde konstruiert, ist es also notwendig ein erheblich vereinfachtes Modell zu erstellen.

In Abbildung 2.2 ist ein verallgemeinerter und abstrahierter Bauplan für Wirbeltiere zu sehen. Er ist reduziert auf die Wirbelsäule, Rippen, Schädel, Extremitäten, Schulterblatt und Beckenknochen. Die Extremitäten bestehen jeweils aus Oberarm/-schenkel, Unterarm/-schenkel und Hand/Fuß. Hände und Füße werden hier jeweils als ein „Knochen“ betrachtet, obwohl sie natürlich aus vielen Einzelteilen bestehen. Es kann maximal zwei Extremitätenpaare geben, jeweils einer an jedem Extremitätengürtel. Rippen, Hals- und Schwanzwirbelsäule sind optional und auch die Anzahl der Wirbel und Rippen variiert je nach Tier.

2.2. Von Groß und Klein

Dieser Abschnitt widmet sich der Frage nach Unterschieden und Gemeinsamkeiten von großen und kleinen Wirbeltieren. Kann man Wirbeltiere und ihre Skelette einfach vergrößern und verkleinern? Oder sind sie besonders an ihre Größe angepasst? Unterscheidet sich das Skelett einer Maus wesentlich von dem eines Elefanten?

Zunächst ein kurzer Überblick über die Dimensionen, in denen man sich hier bewegt: Das größte und schwerste Wirbeltier ist der Blauwal, er kann bis zu 120 Tonnen wiegen (siehe A.1.2). Das aktuell kleinste bekannte Wirbeltier, mit einer Länge von 7 bis 8mm, ist

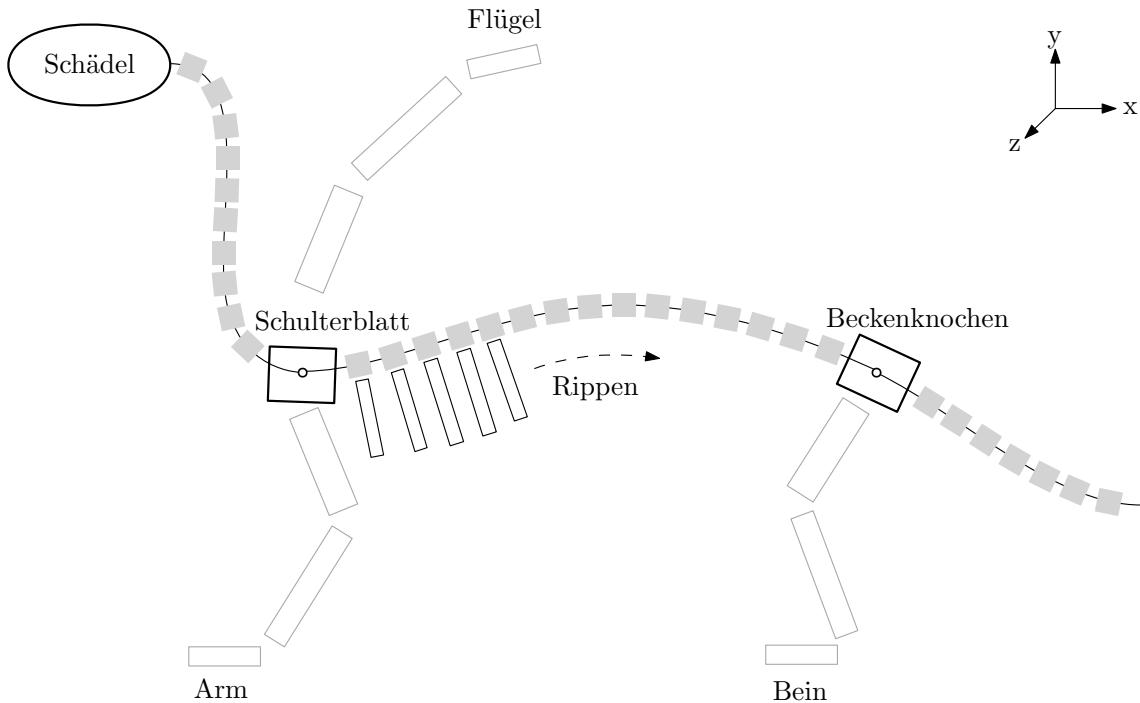


Abbildung 2.2.: Verallgemeinerter und abstrahierter Bauplan eines Wirbeltierskeletts. Es sind maximal zwei Extremitätenpaare erlaubt, jeweils eines an jedem Extremitätengürtel. Rippen und Hals- und Schwanzwirbelsäule sind optional. Die Anzahl der Wirbel und Rippen variiert je nach Tier. Oben rechts im Eck ist die Orientierung der Koordinatenachsen angegeben.

der Frosch *Paedophryne amauensis* [40]. Größe und Gewicht von Wirbeltieren kann also erheblich variieren.

Wenn eine Tierart im Laufe der Evolution wächst, so ändert sich die Größe gleichmäßig. Dieses Wachstum ist aber begrenzt. Der Grund dafür ist, dass die Oberfläche nicht proportional mit dem Volumen mitwächst. Beispielsweise produziert ein Säugetier Wärme proportional zu seinem Volumen. Es kann aber nur Wärme proportional zu seiner Oberfläche abgeben. Dadurch haben große Säugetiere eher Probleme Wärme abzugeben und kleine warm zu bleiben. Ein ähnliches Problem tritt bei Knochen auf. Die Last, die ein Knochen tragen kann, ist ungefähr proportional zu seiner Querschnittsfläche. Das Skelett muss aber das komplette Gewicht des Körpers tragen, was wiederum proportional zum Volumen ist. Aus diesem Grund müssten die Knochen großer Tiere eigentlich überproportional vergrößert sein. Das ist jedoch nicht unbedingt der Fall. Das Problem an dicken Knochen ist, dass sie den Bewegungsradius, z. B. der Extremitäten, stark einschränken. Deshalb passen viele Tiere stattdessen ihre Körperhaltung und Aktivität an, um Spitzenkräfte, wie Stöße oder Oszillationen, auf ihre Knochen zu vermeiden. Deshalb wirken große Tiere auch eher „behäbiger“ als kleine. [22, Kapitel 23]

Sehr große und sehr kleine Wirbeltiere können also nicht über einfaches Skalieren ineinander überführt werden. Dennoch ähneln sich ihre Skelette sehr.

2.3. Mythologie

In diesem Abschnitt wird ein kurzer Blick über die Biologie hinaus, auf fantastische Tiere aus der Mythologie geworfen. Ein Algorithmus, der wirbeltierähnliche Skelette erzeugt, bewegt sich ja schon über die Grenzen der Biologie hinaus. Da ist es nur ein kleiner Schritt auch fiktive Wirbeltiere mit zusätzlichen Gliedmaßen zu erzeugen. Einen guten Überblick

über solche Tiere, nicht nur aus der Mythologie sondern auch aus aktuellerer Literatur, Filmen oder Spielen, gibt die Webseite [46].

Hier sollen drei Beispiele hervorgehoben werden, die jeweils charakteristische Merkmale besitzen. Sie stammen aus der Mythologie und bilden oft auch die Grundlage für andere fiktive Wirbeltiere.

Zentauren haben einen Pferdekörper, aber statt eines Halses setzt ein menschlicher Oberkörper auf dem Schultergürtel auf [35]. Sie haben also zwei Schultergürtel.

Der *Pegasus* hat ebenfalls den Körper eines Pferdes, aber auch Flügel, die zusätzlich zu den Vorderbeinen am Schultergürtel ansetzen [36]. Es ist also „möglich“ an einem Extremitätengürtel zwei Paare von Extremitäten ansetzen zu lassen. Ein anderes Tier mit sechs Extremitäten an zwei Extremitätengürteln ist das Schreckenspferd (oder auch Direhorse) aus dem Film „Avatar“ [6]. Es besitzt vier Beine am Schultergürtel.

Asiatische Drachen haben einen langen, schlangenartigen Körper mit oft mehr als vier Beinen. Sie können als Inspiration für Tiere dienen, die mehr als nur zwei Extremitätengürtel entlang des Rückens besitzen.

3. Grundlagen

3.1. Prozedurale Generierung

Werden Inhalte, wie Texturen oder 3D-Objekte, generiert, ohne dass diese vor der Ausführung des Algorithmus festgelegt wurden, so wird dies als *prozedurale Generierung* (PG) bezeichnet. Ursprünglich wurde PG verwendet, weil der Speicherplatz auf Computern sehr begrenzt war. Große 3D-Landschaften oder andere vorgefertigte künstlerische Werke konnten nicht gespeichert werden.

Die Demoszene entstand in den 1980er Jahren und treibt PG ins Extreme. Sie zeigt mit künstlerischen Inhalten, dass beeindruckende Ergebnisse trotz stark limitiertem Speicherplatz möglich sind bzw. wie man das volle Potential von Computerhardware ausschöpft [12].

Auf heutigen Rechnern ist der Speicherplatz nicht mehr so begrenzt und es kann viel Inhalt, z. B. für PC-Spiele, von Künstlern vorgefertigt werden. Trotzdem behält PG ihre Daseinsberechtigung, da der Entwurf komplexer Inhalte, wie beispielsweise Landschaften mit viel Vegetation, aufwändig ist und viel Zeit benötigt. Hier können wieder Algorithmen aus der PG eingesetzt werden um Zeit und Arbeit zu sparen. Sie können sowohl fertige 3D-Modelle generieren, als auch Inspiration und Hilfe für den Künstler sein. *SpeedTree* [43] ist ein Beispiel für Software, die für die Generierung von Vegetation verwendet wird. Sie unterstützt und beschleunigt die Erstellung von Szenen mit viel Vegetation. [9]

Dinge, die prozedural generiert werden, sind Landschaften, Straßennetze, Gebäude, Menschen, Tiere oder auch Geschichten [16]. Humanoide Charaktere für Computerspiele werden z. B. in der Masterarbeit [29] erzeugt. Die Zusammenstellung [9] untersucht PG von Wesen, die Videospiele bevölkern. Obwohl einige Arbeiten genannt werden, die sich mit diesem Thema beschäftigen, wird auch festgestellt, dass dieser Themenbereich noch wenig erforscht ist.

Ein Beispiel, das erst nach Erscheinen dieser Zusammenstellung auf den Markt gekommen ist, ist das Computerspiel „No Man’s Sky“ [33]. Darin gibt es viele prozedural generierte fantastische Tiere. Die Generierung dieser Tiere funktioniert so, dass zunächst prozedural ein 3D-Modell generiert wird, in welches anschließend ein Rig eingepasst wird [13]. Die dort verwendeten Rigs sind keine wirklichkeitsgetreuen Skelette, sondern werden nur für die Animation verwendet (siehe Abschnitt 3.2).

ToDo

(Beispiele für Methoden der prozeduralen Generierung angeben, z. B. perlin noise)

3.2. Animation von Figuren

Um Figuren für Computerspiele oder Filme zu animieren, ist außer dem 3D-Modell ein Skelett bzw. Rig nötig. Dieses Rig ist oft sehr abstrakt und hat wenig mit lebensechten Skeletten zu tun. Es gibt eine Baumstruktur auf den Einzelteilen der Figur vor. Diese Baumstruktur ist für viele Algorithmen, die zur Animation verwendet werden essentiell, z. B. für inverse Kinematik (siehe Abschnitt 3.3). Als Wurzel für das Rig wird meist ein Knochen nahe des Schwerpunkts der Figur verwendet. Das ist bei humanoiden Figuren die Hüfte.

Weit verbreitete Editoren für Animation, Modellierung, Simulation und Rendering sind Maya [31] und 3ds Max [1]. Auch hier werden solche Rigs verwendet [32]. 3ds Max stellt z. B. in einer Komponente namens Biped vorgefertigte Rigs für humanoide Figuren zur Verfügung, die einfach in schon existierende 3D-Modelle eingefügt werden können [2].

Animationen werden dann mit Hilfe sogenannte Keyframes erzeugt. Keyframes sind Positionen, die die Figur im Laufe der Bewegung einnehmen soll. Interpolation zwischen den einzelnen Keyframes ergibt dann eine flüssige Bewegung. [21, Kapitel 3]

Im Computerspiel Spore [44] werden sogar zur Laufzeit Animationen für Figuren erzeugt, deren Aufbau vorher noch nicht bekannt war. Dazu werden vorher abstrakte Animationen, unabhängig vom Aufbau der speziellen Figur, erstellt. Diese werden dann zur Laufzeit auf die konkreten Figuren angewandt. [21]

Um besonders lebensecht wirkende Tiere zu erzeugen, ist es nötig den Aufbau des entsprechenden Tieres möglichst wirklichkeitsgetreu zu modellieren. Dazu sind Modelle für Skelett, Muskeln und Haut nötig. Solche Modelle lassen sich beispielsweise mit dem Maya-Plugin Ziva [53] gestalten. [54]

Der Algorithmus, der hier erarbeitet wird, erzeugt Skelette mit lebensechten Knochenmodellen. Er kann als Inspiration zur Erstellung neuer Charaktere dienen oder auch als Grundlage für ein Tier, das ein realitätsnahe Skelett haben soll.

3.3. Inverse Kinematik

Inverse Kinematik wird beispielsweise in der Robotik zur Bewegung von Roboterarmen verwendet. Gegeben ist eine Kette von Gelenken, die über starre Verbindungsteile miteinander verbunden sind. Für ein oder mehrere Punkte dieser Kette können Zielpunkte im Raum angegeben werden. Das Ziel ist es dann eine Konfiguration der Gelenke zu berechnen, so dass alle Punkte ihr Ziel erreichen.

Es gibt viele verschiedene Methoden Probleme aus der inversen Kinematik zu lösen. Diese finden jeweils in Abhängigkeit von der konkreten Problemstellung ihre Anwendung. Beispielsweise geht es in [17] um die Animation von Beinen. Ein Überblick über die verschiedenen Techniken ist in der Zusammenstellung [4] zu finden.

3.4. Grammatiken

Eine *Grammatik* ist ein Tupel $G = (\Sigma, N, S, P)$ mit

- dem endlichen Alphabet Σ der Terminalsymbole
- dem endlichen Alphabet N der Nichtterminalsymbole
- dem Startsymbol $S \in N$ und
- der Menge von Produktionen $P \subseteq (N \cup \Sigma)^* N (N \cup \Sigma)^* \rightarrow (N \cup \Sigma)^*$

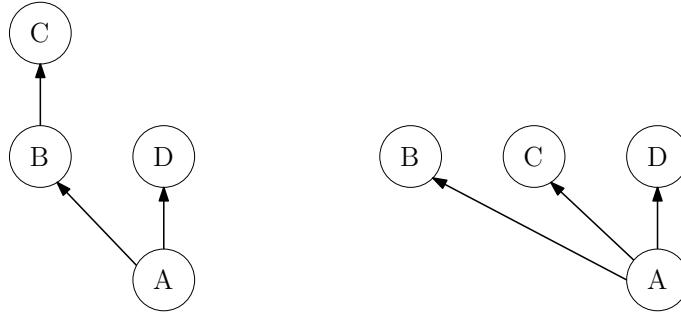


Abbildung 3.1.: Visualisierung der Beispiele für Klammerausdrücke aus dem Text, links der Ausdruck $A(BC)D$, rechts $A(B)(C)D$.

Eine Teilmenge der Grammatiken bilden die *kontextsensitiven* Grammatiken. Ihre Produktionen haben die Form

$$\alpha A \beta \rightarrow \alpha \gamma \beta \text{ oder } S \rightarrow \epsilon$$

mit $A \in N$, $\alpha, \beta, \gamma \in ((N \setminus \{S\}) \cup \Sigma)^*$, $\gamma \neq \epsilon$ und dem leeren Wort ϵ .

Wiederum eine Teilmenge davon bilden die *kontextfreien* Grammatiken mit Produktionen der Form

$$A \rightarrow w \text{ mit } A \in N \text{ und } w \in (\Sigma \cup N)^*.$$

[19, Abschnitte 1.4 und 1.5]

L-Systeme oder Lindenmayer-Systeme sind spezielle Grammatiken. Sie wurden 1968 von dem Biologen Aristid Lindenmayer vorgestellt, mit dem Zweck das Wachstum vielzelliger Lebewesen zu modellieren. Im Unterschied zu Grammatiken, bei denen Produktionen einzeln nacheinander angewendet werden, werden bei L-Systemen alle Symbole gleichzeitig ersetzt. Dies soll modellieren, dass viele Zellteilungen zur gleichen Zeit stattfinden können. Es gibt auch hier wieder verschiedene Teilmengen wie z. B. kontextfreie und kontextsensitive L-Systeme. Kontextfreie L-Systeme sind aber beispielsweise keine Teilmenge kontextfreier Grammatiken. Es gibt Sprachen, die von kontextfreien L-Systemen, aber nicht von kontextfreien Grammatiken erzeugt werden können und anders herum [39, Abbildung 1.2]. Oft werden L-Systeme verwendet um das Wachstum von Pflanzen zu modellieren. Es gibt aber auch andere Anwendungsbeispiele, wie z. B. die Generierung von Städten [37].

Bei Pflanzen treten oft Verzweigungen auf. Um diese zu modellieren, werden in L-Systemen Klammerstrukturen verwendet. Das Teilwort innerhalb einer Klammer repräsentiert einen „Ast“, der wiederum in sich verzweigt sein kann. Wenn zwei Symbole direkt aufeinander folgen, so bedeutet das, dass das linke Element das Elternelement des rechten ist. Kommt in einem Wort $A(BC)D$ mit $A, B, C, D \in \Sigma \cup N$ ein Klammerausdruck vor, so bedeutet dies, dass A sowohl das Elternelement von B als auch von D ist. Das Elternelement von C ist B . Es können auch Mehrfachverzweigungen auftreten. In $A(B)(C)D$, was gleichbedeutend ist zu $A(B)(C)(D)$, ist A das Elternelement von B, C und D . Eine Visualisierung der beiden Beispiele ist in Abbildung 3.1 zu finden. [39, Kapitel 1]

3.5. Principal Component Analysis

Principal Component Analysis (PCA) oder auch Hauptkomponentenanalyse [28] wird meist mit dem Ziel angewendet die Dimensionalität einer Menge von Datenpunkten zu verringern, dabei aber möglichst wenig Information zu verlieren. Beispielsweise wird dies in [11] bei 3D-Modellen von Gesichtern oder in [41] bei 3D-Modellen von menschlichen Körpern gemacht. Auch im Zusammenhang mit prozeduraler Generierung wird PCA oft

verwendet, z. B. zur Generierung von humanoiden Charakteren für Videospiele [29] oder Texturen für Gesichter [7].

Als Ausgangspunkt für eine PCA dient eine Menge von Datenpunkten (oder Beispielen) D im n -dimensionalen Raum. Voraussetzung ist, dass die Punkte in jeder Dimension normalverteilt sind. Die Datenpunkte D sind dann nach einer n -dimensionalen Normalverteilung verteilt. Betrachtet man nun die Wahrscheinlichkeitsdichte dieser Normalverteilung, so bilden alle Punkte, bei denen die Dichte einen Wert größer als ϵ annimmt das Innere eines $(n - 1)$ -dimensionalen Hyperellipsoids. In diesem Volumen liegen die meisten der Datenpunkte D . Den Mittelpunkt bildet ihr Mittelwert.

Nun ist das Ziel herauszufinden wo die Achsen des Hyperellipsoids liegen. Betrachtet man die Komponenten der Datenpunkte D in den Richtungen der Achsen, so sind diese wiederum normalverteilt und die Verteilungen sind jeweils unabhängig voneinander. Um die Achsen des Hyperellipsoids zu berechnen, wird zunächst die Kovarianzmatrix der Datenpunkte D aufgestellt und dann diagonalisiert. In den Spalten der damit berechneten Basiswechselmatrix stehen dann die Eigenvektoren und auf der Diagonalen der diagonalisierten Kovarianzmatrix die Eigenwerte. Die Eigenvektoren sind die Achsen des Hyperellipsoids und die Eigenwerte geben die Varianz der Normalverteilung entlang der Achsen an.

Jetzt können die Datenpunkte D vom Eingabekoordinatensystem in dasjenige Koordinatensystem transformiert werden, das von den Achsen des Hyperellipsoids aufgespannt wird. Dazu stellt man die Punkte als Linearkombinationen der Eigenvektoren dar. Um nun die Dimensionalität der Eingabedaten zu reduzieren, werden alle Dimensionen weggelassen, die nicht zu den Hauptkomponenten gehören. Je weniger Streuung die Datenpunkte auf denjenigen Achsen aufweisen, die weggelassen werden, desto weniger Information wird verworfen.

Will man einen zufälligen Datenpunkt erzeugen, der die gleichen Verteilungen aufweist, wie die Eingabebeispiele, so funktioniert das im transformierten Koordinatensystem sehr gut. Da die einzelnen Dimensionen unabhängig voneinander sind, kann man einfach für jede Dimension eine normalverteilte Zufallszahl mit entsprechender Varianz generieren.

3.6. Quantil-Quantil-Diagramme

Quantil-Quantil-Diagramme sind eine grafische Methode um zu testen, ob eine Stichprobe einer bestimmten Verteilung unterliegt.

Seien x_1, x_2, \dots, x_n die Elemente einer Stichprobe und F die Verteilungsfunktion derjenigen Verteilung, gegen die getestet werden soll. Die Verteilungsfunktion gibt für einen Wert x an, wie groß die Wahrscheinlichkeit ist, dass die Zufallsvariable einen Wert $\leq x$ annimmt.

Nun werden die empirischen Quantile der Stichprobe mit den entsprechenden Quantilen von F verglichen. Dazu wird die Stichprobe aufsteigend sortiert, was $x_{(1)}, x_{(2)}, \dots, x_{(n)}$ ergibt. Das empirische $\frac{i}{n}$ -Quantil der Stichprobe ist $x_{(i)}$, da i Werte $\leq x_{(i)}$ beobachtet wurden. Die inverse Verteilungsfunktion gibt für einen Wert y den kleinsten Wert x an, für den gilt, dass $F(x) > y$, also das y -Quantil. Das $\frac{i}{n}$ -Quantil von F ist also $F^{-1}\left(\frac{i}{n}\right)$.

Trägt man nun die sortierte Stichprobe gegen die entsprechenden Quantile in ein Diagramm ab, und die angenommene Verteilung ist korrekt, so liegen die Punkte annähernd auf einer Geraden mit Steigung 1 durch den Ursprung. [30, Kapitel 1 und 2]

Den Werten $x_{(i)}$ einfach das $\frac{i}{n}$ -Quantil zuzuordnen ist aber insofern schwierig, als dass es suggeriert, dass alle Werte, die mit der angenommenen Verteilung generiert werden, kleiner als $x_{(n)}$ sind. Deshalb gibt es verschiedene andere Möglichkeiten die Quantile zuzuordnen.

Sie liefern aber für $n \rightarrow \infty$ alle das gleiche Ergebnis. Sogenannte „Rankit Plots“ verwenden folgende Zuordnung:

$$x_{(i)} \approx F^{-1} \left(\frac{i - 0,5}{n} \right).$$

Diese Zuordnung wird auch in dieser Arbeit verwendet.

3.7. Genetische Algorithmen

Ein sehr bekannter Zweig der evolutionären Algorithmen sind die genetischen Algorithmen. Die Grundprinzipien, die von genetischen Algorithmen verwendet werden, sind von Evolution und natürlicher Auslese abgeschaut. Im einfachsten Fall ist eine Menge von Individuen (oder eine Population) gegeben, die durch „Gene“ kodiert sind, und eine Fitnessfunktion. Die Fitnessfunktion bewertet die Individuen. Das Ziel ist Individuen zu entwickeln, die möglichst gut bewertet werden. Dazu werden aus der aktuellen Population zufällig, aber nach Fitness priorisiert, Individuen ausgewählt. Diese produzieren dann „Nachkommen“ indem sie ihre Gene nach einer bestimmten Vorschrift vermischen. Diese „Nachkommen“ bilden dann eine neue Population, die wiederum anhand der Fitnessfunktion bewertet wird. Damit startet eine neue Iteration. Dies wird so lange gemacht bis ein Individuum gefunden wurde, das die Fitnessfunktion gut genug erfüllt. [23] [10, Abschnitt 1.3.1]
In interaktiven Versionen evolutionärer Algorithmen können sogar Benutzer die Fitnessfunktion ersetzen, indem sie selbst die erzeugten Individuen bewerten. [45]

In der Masterarbeit [24] werden 3D-Modelle von Lebewesen und ein passendes Rig (siehe Abschnitt 3.2) mit Hilfe von genetischen Algorithmen generiert. Das Augenmerk liegt dort aber nicht auf realistischen Tieren oder realistischen Skeletten, im Gegensatz zu dieser Arbeit.

4. Analyse von Skelettdaten mit Hilfe von PCA

Das Ziel dieser Arbeit ist es einen Algorithmus zu entwerfen, der in der Lage ist Skelette zu generieren, die denjenigen von Wirbeltieren ähnlich sehen. Ein Skelett ist nicht nur eine Ansammlung von Knochen, sondern wirkt auch wesentlich durch seine Haltung. Die Skelette sollen also in einer natürlich wirkenden Haltung dargestellt werden, einer Haltung, die das dargestellte Tier auch einnehmen würde.

Wie solch eine Haltung generiert werden könnte, wird in Abschnitt 4.1 erörtert. Das Hilfsmittel, das schließlich für die Generierung verwendet wird, ist die PCA (siehe Abschnitt 3.5). Wie sie in diesem Zusammenhang verwendet wird, wird in den folgenden Abschnitten genauer erklärt. Abschnitt 4.2 beleuchtet die Datenerhebung. In Abschnitt 4.3 werden die Eingabedaten genauer analysiert, z. B. darauf ob sie normalverteilt sind. Die Ergebnisse der PCA werden dann in Abschnitt 4.4 vorgestellt. Im letzten Abschnitt wird gezeigt wie die Eingabe für die PCA auf Skelette mit gewissen Eigenschaften eingeschränkt werden kann.

4.1. Haltung des Skeletts

Gleichgewicht

Ein erster Ansatz dafür ein Skelett in einer natürlichen Haltung zu erzeugen, könnte sein, in jedem Schritt auszurechnen, ob das Skelett ausbalanciert ist. Dazu benötigt man den Schwerpunkt des Körpers und Position und Gewicht der Knochen. Damit können dann die Drehmomente der einzelnen Knochen um den Schwerpunkt berechnet werden. Addieren sie sich alle zu null auf, ist das Skelett im Gleichgewicht.

Hierbei tauchen gleich mehrere Fragen auf:

Gewicht Wie wird das Gewicht der Knochen bestimmt? Hängt es von der Größe der Knochen ab? Außerdem müsste eigentlich das ganze Gewicht des Tieres, nicht nur das der Knochen berücksichtigt werden. Wie kann bestimmt werden wieviel anderes Gewebe an einem Knochen hängt?

Schwerpunkt Wo liegt der Schwerpunkt? Wird am Anfang ein Quader festgelegt, in dem das Skelett liegen muss und dessen Mittelpunkt der Schwerpunkt des Skeletts ist? Verändert der Schwerpunkt seine Position je nachdem welche Knochen schon generiert wurden?

Gleichgewicht Wann wird überprüft ob sich das Skelett im Gleichgewicht befindet? Ist das Gleichgewicht eine Invariante, die während der Generierung aufrecht erhalten wird? Oder wird das erst am Ende überprüft? Was passiert, wenn sich das Skelett nicht im Gleichgewicht befindet?

Das Hauptproblem ist hier, dass nicht klar ist wie das Gewicht der einzelnen Körperteile bestimmt werden soll ohne zusätzlich auch noch anderes Gewebe, wie Muskeln oder Ein geweide, zu betrachten. Da sich diese Arbeit aber nur auf das Skelett beschränken soll, wurde ein anderer Ansatz erarbeitet.

Verlauf der Wirbelsäule

Die Wirbelsäule ist ein zentraler Teil des Wirbeltierkörpers und bestimmt wesentlich das Aussehen des Skeletts. Ihre Form variiert von relativ gerade, z. B. bei Fischen oder Schlangen, bis hin zu stark geschwungenen Hälsen, v. a. bei Vögeln, und langen Schwänzen, z. B. bei Mäusen. Außerdem zeigt sie wie aufrecht ein Tier ist. Große Unterschiede sind hier zwischen Fischen und Vierbeinern gegenüber Vögeln zu beobachten, da Vögel nur auf ihren Hinterbeinen stehen und deshalb ihr Schwerpunkt nach hinten verschoben ist. Es gibt aber auch aufrechtere Exemplare unter den „Vierbeinern“ wie beispielsweise das Känguru oder den Tyrannosaurus Rex. Der Mensch ist natürlich auch ein Beispiel für ein sehr aufrechtes Wirbeltier. Seine Haltung unterscheidet sich aber so stark von der anderer Wirbeltiere, dass er hier außen vor gelassen werden soll.

Viele Knochen setzen direkt an der Wirbelsäule an, wie z. B. der Kopf, die Rippen oder die Hüfte. Durch die Wirbelsäule wird also schon sehr viel vorgegeben. Deshalb eignet sie sich sehr gut als Startpunkt für die Generierung eines Skeletts. Ausgehend von ihr kann dann der Rest des Skeletts generiert werden.

Wie soll aber nun die Lage der Wirbelsäule bestimmt werden?

Hierfür schien es sinnvoll viele Beispiele zu betrachten und Zusammenhänge zwischen verschiedenen Eigenschaften der Tiere und dem Verlauf der Wirbelsäule zu suchen. Ein geeignetes Werkzeug hierfür ist die *Principal Component Analysis* (siehe Abschnitt 3.5 des Grundlagenkapitels).

4.2. Datenerhebung

Die Beispiele, die erhoben werden konnten, sind vor allem durch die Datenlage bzw. die zugänglichen Quellen eingeschränkt. Trotzdem wurde darauf geachtet, möglichst viele unterschiedliche Tierarten mit viel Variation in den erhobenen Merkmalen zu finden.

Viele Beispiele entstammen Zoologiebüchern, in denen sie als Beispiele für Erklärungen angegeben waren (Bildquellen siehe Anhang A.1.1). Dem ist auch geschuldet, dass recht viele Dinosaurierskelette dabei sind, denn von anderen Tieren gibt es als alternative Darstellungsmöglichkeit eine Außenansicht des lebenden Tieres. Das geht bei ausgestorbenen Tieren im Allgemeinen nicht.

Erhobene Merkmale

Die Merkmale, die zur Datenerhebung ausgesucht wurden, sind charakteristisch für ein Skelett, tragen also viel zum Gesamteindruck bei. Das sind vor allem der Verlauf der Wirbelsäule und der Aufbau der Extremitäten.

Eingeschränkt wurde die Erhebung natürlich auch durch die begrenzte Datenlage. 2D-Bilder mit Seitenansichten von Skeletten waren gut verfügbar. Merkmale, die Tiefeninformationen benötigen, z. B. den Abstand der Füße, sind durch sie aber nicht zu bekommen.

Auch Informationen zu sehr kleinen Knochen, wie Handwurzelknochen oder die unterschiedlichen Fingerknochen, sind schwierig zu erheben, da sie teilweise schwer zu erkennen und zu markieren sind. Deshalb ist die Erhebung auf folgende Daten eingeschränkt:

- Ein Bild mit der Seitenansicht des Skeletts. Darin wurde die Lage der *Wirbelsäule* und die Länge der Knochen der *Vorder- und Hinterextremitäten* markiert, falls vorhanden. Konkret wird für die Extremitäten die Länge des Oberschenkels/-arms, des Unterschenkels/-arms, des Fußes und der Hand erhoben.
Hier hätte man zusätzlich auch die Positionen der Knochen und damit auch die Winkel an den Gelenken erheben können. Da die Position der Extremitäten auf den Abbildungen aber wenig vergleichbar und relativ beliebig ist, ist das nicht sinnvoll.
- Die *Tierklasse*, also ob das Tier ein Fisch, eine Amphibie, ein Reptil, ein Säugetier oder ein Vogel ist. Dieses Merkmal lässt sich nicht auf einer kontinuierlichen Skala abbilden und ist deshalb nicht als Eingabedimension für die PCA geeignet. Es wurde trotzdem erhoben, da es für eine anderweitige Auswertung hilfreich sein könnte.
Auch andere Merkmale wie etwa der Lebensraum könnten für weitere Analysen interessant sein. Man könnte sie, je nach Ziel, in zukünftigen Analysen mit aufnehmen.
- Ob *Flügel* vorhanden sind oder nicht. Dieses Merkmal kann in den erhobenen Beispielen nur zwei Werte annehmen. Flügel sind entweder vorhanden (1) oder nicht (0). Betrachtet man die Werte aber auf einer kontinuierlichen Skala, so lassen sie sich als Wahrscheinlichkeit dafür betrachten, dass das Tier Flügel hat.
Auch hier können, wie bei den *Flügeln*, nur diskrete Werte in den Beispielen angenommen werden. Werte in $[0, 1]$ bzw. in $[1, 2]$ lassen sich aber wiederum als Wahrscheinlichkeiten dafür interpretieren, ob das Tier ein bzw. zwei Beinpaare hat.
- Die Anzahl der Paare von *Beinen mit Bodenkontakt*. Hier werden sowohl Vorder- als auch Hinterbeine berücksichtigt. Außerdem werden nur Paare berücksichtigt, da ein Tier keine ungerade Anzahl von Beinen haben kann.
Hier wurden oft Angaben zum maximalen Gewicht der Tiere verwendet, da keine Angaben zum Durchschnittsgewicht zu finden waren. Teilweise gibt es auch verschiedene Tierarten einer Gattung, die unterschiedlich schwer werden können, deren Skelette aber, bei der Auflösung der hier erhobenen Daten, gleich aussehen. In diesem Fall wurde ein beliebiger Wert gewählt, der zwischen dem Gewicht der leichtesten und dem der schwersten Art liegt. (Die Quellen zu den Gewichten sind im Anhang A.1.2 zu finden.)

Wie genau die Datenerhebung auf den Bildern der Skelette durchgeführt wurde ist im Kapitel „Implementierungsdetails“ im Abschnitt 7.4 nachzulesen. Das Ergebnis der Erhebung sind annotierte Bilddateien und eine Textdatei, in der Tierklasse, Flügel, Beine und Gewicht erfasst werden.

Die Lage der Wirbelsäule wird durch drei kubische Bézierkurven erfasst, jeweils eine für Hals, Rücken und Schwanz. Hals und Rücken gehen an der Schulter ineinander über und Rücken und Schwanz an der Hüfte. Diese Bézierkurven benötigen 20 Eingabedimensionen der PCA (10 zweidimensionale Punkte). Bei manchen Tieren ist kein Hals oder kein Schwanz vorhanden. In diesen Fällen werden die 3 fehlenden Punkte jeweils mit dem ersten bzw. letzten Punkt des Rückens ersetzt.

Bézierkurven wurden aufgrund ihrer einfachen Handhabung verwendet. Sie werden von Programmen unterstützt, mit denen Vektorgrafiken erstellt werden können, sind leicht in

die Beispielbilder einzutragen und leicht zu interpretieren. Es könnten aber natürlich auch andere Repräsentationen, wie B-Splines, verwendet werden. B-Splines und Bézierkurven können sogar ineinander umgewandelt werden [38].

Insgesamt gehen also folgende Merkmale als Eingabedimensionen in die PCA ein: die Lage der *Wirbelsäule* (20 Dimensionen), die Länge der Knochen der *Vorder- und Hintergliedmaßen* (6 Dimensionen), die Anzahl der Paare von *Flügeln* (1 Dimension), die Anzahl der Paare von *Beinen mit Bodenkontakt* (1 Dimension) und das *Gewicht* (1 Dimension). Die *Tierklasse* wird zwar erhoben, geht aber nicht in die PCA ein.

Bei der Annotation der Bilder sind folgende Schwierigkeiten aufgetreten:

- Bei Fischen ist nicht klar wo der Rücken in den Schwanz übergeht, da der Beckengürtel sich teilweise am Kopf befindet oder gar nicht vorhanden ist (siehe auch Abschnitt 2.1). Bei der Datenerhebung wurde der Übergang ungefähr bei der Rücken- oder der Afterflosse festgelegt. Alternativ hätte man auch die komplette Wirbelsäule als Rücken klassifizieren können.
- Manche Hälse und Schwänze sind mit einer kubischen Bézierkurve nicht darstellbar. Das ist unter den verwendeten Beispielen der Hals von Ichthyornis und Schwan und der Schwanz von Ichthyosaurus und Koboldmaki (siehe Abbildungen A.1 p, ag, q und w). In diesen Fällen wurde versucht die Form möglichst gut anzunähern oder Fortsätze (wie am Schwanz vom Ichthyosaurus) einfach wegzulassen.
- Die Schwanzposition bei Tieren mit sehr langen Schwänzen ist auf den Bildern relativ beliebig. Hier wurde versucht, den Schwanz möglichst gerade nach hinten fortzusetzen, auch wenn er auf dem Bild eine andere Position hat.

4.3. Analyse der Eingabedaten

Insgesamt wurden 44 Datenpunkte erhoben. Das entspricht bei 29 Dimensionen $44 \cdot 29 = 1276$ Zahlen als Eingabe. Das Ergebnis der PCA sind im Wesentlichen 29 Eigenwerte und -vektoren. Das sind $29^2 + 29 = 870$ Zahlen als Ausgabe. Die Datengrundlage ist also immerhin größer als die Ausgabe, da $1276 > 870$ und $44 > 29$. Dennoch wäre es besser mehr Datenpunkte zu haben, um die Ergebnisse aussagekräftiger zu machen.

Untersuchung auf Normalverteilung

Eine Voraussetzung dafür, dass die PCA korrekt funktioniert, ist, dass die Eingabedaten in jeder Dimension normalverteilt sind. Das soll im Folgenden genauer untersucht werden. Visualisierungen der Eingabedaten in den einzelnen Dimensionen sind in den Abbildungen A.2 und A.3 im Anhang zu finden.

Es gibt die beiden Merkmale *Flügel* und *Beine mit Bodenkontakt*, die offensichtlich nicht normalverteilt sind, da sie diskret sind und nur zwei bzw. drei verschiedene Werte annehmen können. Sie können jedoch hilfreiche Informationen zur Weiterverarbeitung durch den Algorithmus liefern, weshalb sie nicht ganz außen vor gelassen werden sollten. Da das Hauptaugenmerk der PCA aber auf der Position der Wirbelsäule und dem Aufbau der Extremitäten liegt, sollen diese beiden Merkmale keinen großen Einfluss auf die Hauptkomponenten haben. Deshalb wurde der Einfluss dieser Merkmale verringert. Dazu weiter unten mehr.

Die anderen Merkmale wurden mithilfe eines Quantil-Quantil-Diagramms (siehe Abschnitt 3.6) mit der Normalverteilung verglichen. Die Diagramme zeigen für alle Merkmale, außer dem Gewicht, dass sie recht gut normalverteilt sind (Beispiele in Abbildung 4.1 und eine vollständige Aufzählung im Anhang in den Abbildungen A.4 und A.5).

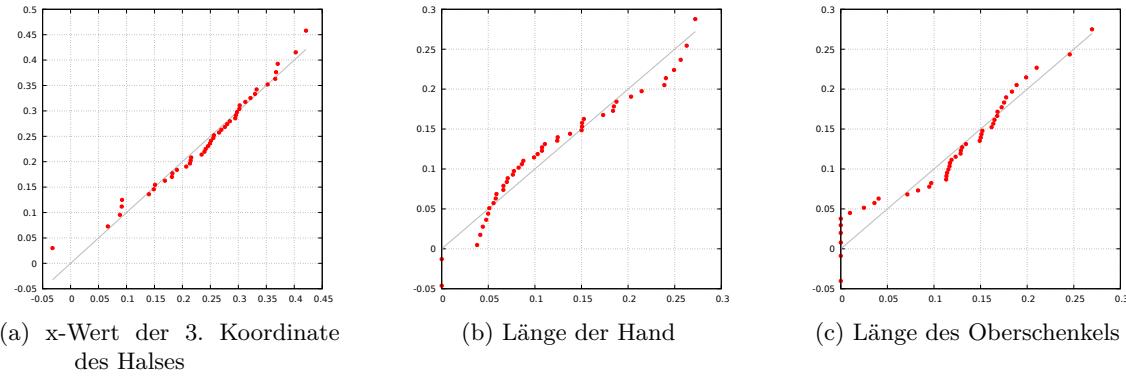


Abbildung 4.1.: Beispielhaft ausgewählte Quantil-Quantil-Diagramme von drei Eingabedimensionen. (a) weicht nicht stark von der Normalverteilung ab, (b) und (c) hingegen schon mehr, sind aber trotzdem noch akzeptabel verteilt.

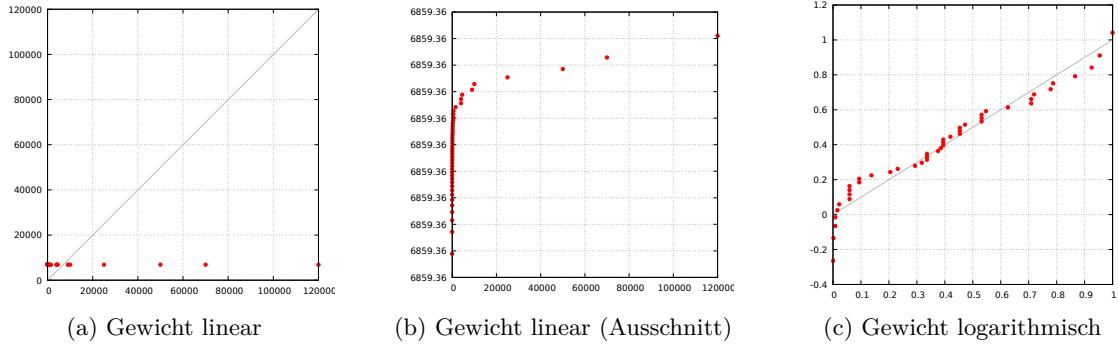


Abbildung 4.2.: Quantil-Quantil-Diagramme des Gewichts, einmal linear (a,b) und einmal mit logarithmischer Skala (c)

Das Merkmal *Gewicht* ist überhaupt nicht normalverteilt (siehe Abbildung 4.2 a und b). Verwendet man es als Eingabe für die PCA und generiert dann zufällig normalverteilte Punkte im Ergebnisraum, treten schnell Gewichte kleiner null auf. Betrachtet man das Gewicht jedoch auf einer logarithmischen Skala, so ist es normalverteilt (siehe Abbildung 4.2 c). Deshalb ist es sinnvoll das Gewicht zunächst zu logarithmieren, bevor es in die PCA eingeht. Welcher Logarithmus hier zur Umrechnung der Daten verwendet wird schlägt sich nur als linearer Faktor nieder und ist eine Frage der Skalierung. Es wurde der Zehnerlogarithmus verwendet. Da das Gewicht, genauso wie die Merkmale *Flügel* und *Beine mit Bodenkontakt*, kein Hauptmerkmal für die Untersuchung sein sollte, wurde es ebenfalls kleiner skaliert.

Skalierung der Merkmale

Generell bewirkt die Skalierung eines Merkmals eine Gewichtung, denn durch eine Skalierung ändert sich die (Ko-)Varianz und somit auch die Kovarianzmatrix. Seien beispielsweise $s, t \in \mathbb{R}$, dann bewirkt eine Skalierung des Merkmals x mit s und eine Skalierung des Merkmals y mit t eine Skalierung von $s \cdot t$ der Kovarianz $\text{Cov}(x, y)$ von x mit y , da $\text{Cov}(sx, ty) = (sx - s\mu_x)(ty - t\mu_y) = st \cdot \text{Cov}(x, y)$, mit Erwartungswert μ_i für Merkmal i .

Wie genau werden die einzelnen Merkmale nun skaliert?

Zunächst werden alle Merkmale auf das Intervall $[0, 1]$ skaliert, damit alle den gleichen Einfluss haben. Bei Koordinaten oder Längen im Bild bedeutet das, dass sie durch 1000

geteilt werden, da sie in Pixeln dargestellt werden und das Bild eine Größe von 1000×1000 Pixeln hat. Bei Längen von Strecken im Bild wären dabei theoretisch auch Werte größer 1000px möglich. Solche Längen wären aber unrealistisch und werden deshalb ignoriert.

Koordinaten und Längen im Bild sind diejenigen Merkmale, die hier am interessantesten sind. Sie stellen den Verlauf der Wirbelsäule und die Längen der Knochen der Extremitäten dar. Deshalb sollten sie den größten Einfluss auf die Hauptkomponenten der PCA haben. Alle anderen Merkmale werden kleiner skaliert.

Man könnte statt einer Skalierung mit 0.001 auch für jedes einzelne Merkmal den maximal und minimal angenommenen Wert ermitteln und sie dann so skalieren, dass sie Intervalle gleicher Länge abdecken. Das würde ausgleichen, dass z. B. kleine Längen eine kleinere Varianz und damit auch einen kleineren Einfluss haben. Da aber kleine Elemente im Bild auch weniger zum Gesamteindruck beitragen, wirkt es natürlich, dass sie auch weniger Einfluss auf die Hauptkomponenten haben. Deshalb wird die oben beschriebene Variante verwendet. Falls es in Zukunft Gründe für eine andere Gewichtung gäbe, ließe sich das aber leicht anpassen.

Die diskreten Merkmale *Flügel* und *Beine mit Bodenkontakt* und das logarithmische *Gewicht* werden zunächst ebenfalls auf das Intervall $[0, 1]$ skaliert. Das bedeutet für das angepasste Gewicht $\bar{w} = \frac{\log(w+1)}{\log(\max+1)}$. Das schwerste Wirbeltier ist der Blauwal mit 120 Tonnen (siehe Abschnitt 2.2). Deshalb ist hier $\max = 120.000\text{kg}$.

Danach werden die Werte noch einmal durch 100 geteilt, um ihren Einfluss zu verringern. Das Ziel ist, dass diese Merkmale nicht als große Einträge in den größten Eigenvektoren (den Hauptkomponenten) auftauchen. Ohne diese Skalierung sind diese Merkmale recht dominant. Mit der Skalierung hingegen sind sie in den größten Eigenvektoren unter den kleinsten Werten zu finden.

Obwohl die Merkmale nun sehr kleine Werte annehmen, sind ihre Korrelationen mit anderen Merkmalen trotzdem noch vorhanden. Wird ein zufälliger n -dimensionaler Punkt p mit der gegebenen Normalverteilung generiert, so enthält er also trotzdem Informationen zu *Flügeln* und *Beinen mit Bodenkontakt*. Erst wenn p auf seine Hauptkomponenten reduziert wird, fallen die meisten Informationen dazu weg.

Interessant ist die Projektion der Eingabedaten auf die größten beiden Eigenvektoren. In Abbildung 4.3 ist gut zu vergleichen was die Effekte der Skalierung der Eingabedaten sind. Ganz links sind die Ergebnisse zu sehen, die entstehen, wenn alle Merkmale nur auf das Intervall $[0, 1]$ skaliert werden. In der Mitte geht das Gewicht nicht mehr linear, sondern logarithmisch ein und ganz rechts sind *Flügel*, *Anzahl Beine* und *Gewicht* zusätzlich klein skaliert. Gut zu sehen ist, wie sich die Clusterbildung durch die Skalierung verringert. Gibt es weniger Cluster, ist die Verteilung der Daten näher an einer Normalverteilung.

In Abbildung 4.4 ist noch einmal jeweils die Projektion der skalierten Daten auf die ersten beiden Eigenvektoren zu sehen. Diesmal sind die Daten anhand der verschiedenen diskret erhobenen Merkmale markiert. Es ist z. B. schön zu sehen, dass alle Tiere mit Flügeln auch Vögel sind und dass fast alle Tiere, die zwei Beine haben, Vögel sind. Vier Tiere haben zwei Beine, sind aber keine Vögel: Ohrenrobbe, Seehund, Tyrannosaurus Rex und Känguru.

Auch im Koordinatensystem der Eigenvektoren sollten die Daten wieder normalverteilt sein. Dies wurde ebenfalls mit Quantil-Quantil-Diagrammen untersucht. In Abbildung 4.5 sind die Ergebnisse für die ersten drei Eigenvektoren zu sehen. Es gibt, wie bei den Eingabedaten, keine allzu großen Abweichungen.

Spezielle Punkte

Der Mittelwert der skalierten Eingabedaten ist in Abbildung 4.6 visualisiert. Wie auch bei der Datenerhebung ist hier die Position der Wirbelsäule in einem 1000×1000 Pixel Bild

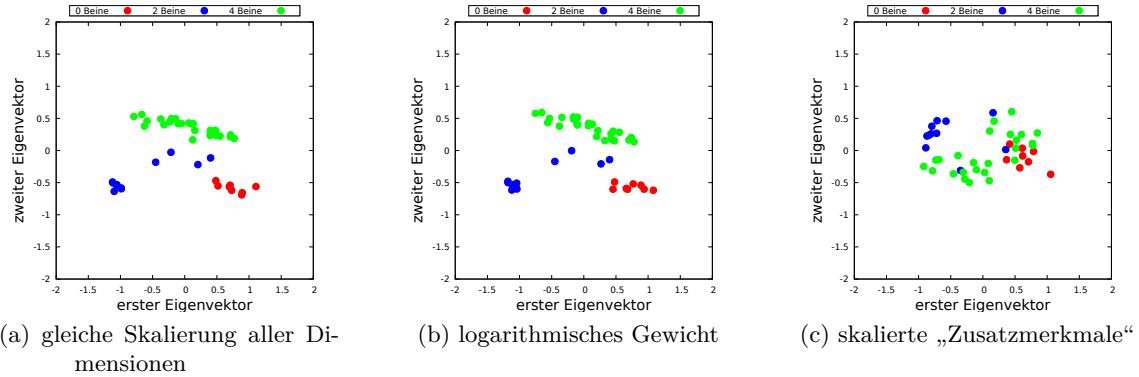


Abbildung 4.3.: Dargestellt sind hier jeweils die Projektionen der Eingabedaten auf die ersten beiden Eigenvektoren. Für jede Version wurden die Eingabedaten unterschiedlich vorverarbeitet. (a) Skalierung aller erhobenen Daten auf das Intervall $[0, 1]$, (b) zusätzlich Verwendung von logarithmischem Gewicht, statt linearem, (c) zusätzliche Skalierung der Merkmale *Flügel*, *Anzahl Beine* und *Gewicht* mit 0,01.

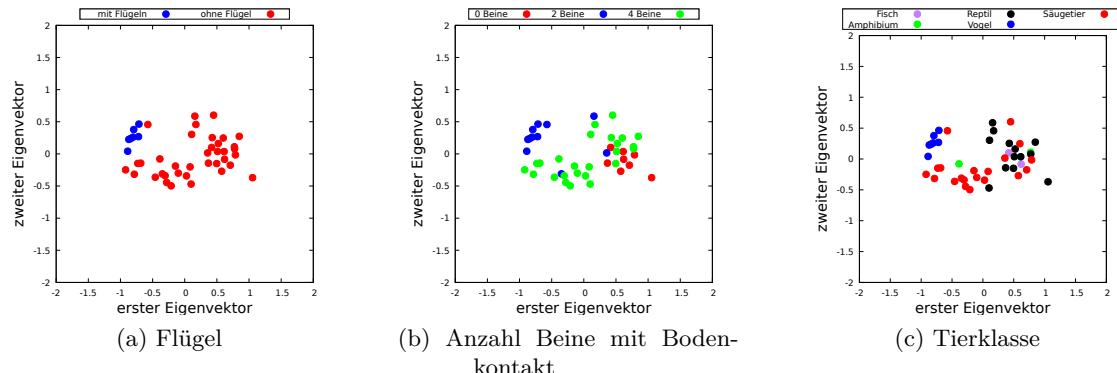


Abbildung 4.4.: Projektion der, wie in Abbildung 4.3c skalierten, Eingabedaten auf die Ebene, die durch den ersten und zweiten Eigenvektor aufgespannt wird. Markiert sind jeweils ob Flügel vorhanden sind (a), die Anzahl der Beinpaare (b) und die Tierklasse (c).

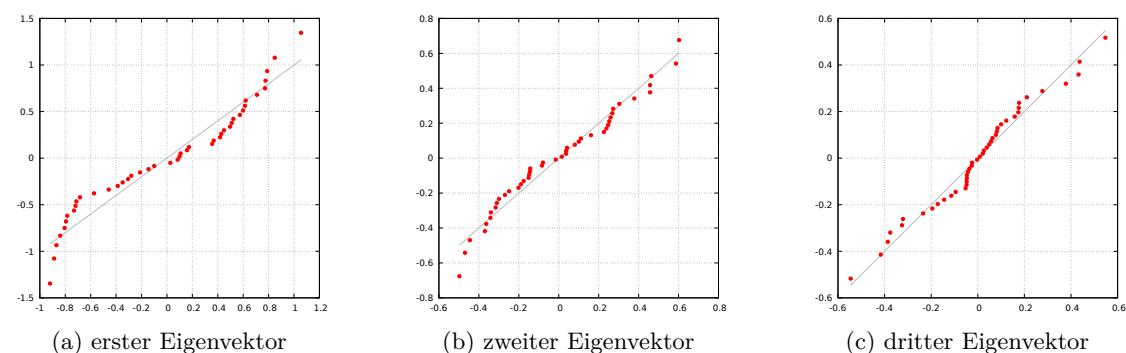


Abbildung 4.5.: Quantil-Quantil-Diagramme der Eingabedaten projiziert auf die größten drei Eigenvektoren.

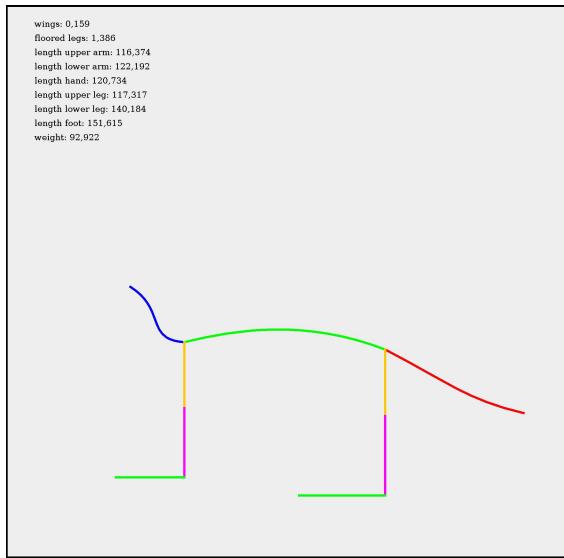


Abbildung 4.6.: Visualisierung des Mittelwerts der skalierten Eingabedaten. Die Werte, die nicht grafisch visualisiert sind, sind folgende: *Flügel* 0,159, *Beine mit Bodenkontakt* 1,39, *Gewicht* 93kg.

gezeigt. Da von den Knochen der Vorder- und Hinterextremitäten nur die Längen erhoben wurden, sind ihre Positionen nicht realistisch. Die restlichen Daten sind nicht visualisiert, sondern nur in Textform angegeben.

Den minimalen Abstand zum Mittelwert hat der Klippschliefer (siehe Abbildung 4.7a). Zusätzlich zum Bild wurde für den Klippschliefer folgende Daten erhoben: *Tierklasse* Säugetier, *Flügel* nein, *Beine mit Bodenkontakt* 2, *Gewicht* 4kg.

Den maximalen Abstand hat die Schlange. Die erhobenen Daten sind hier: *Tierklasse* Reptil, *Flügel* nein, *Beine mit Bodenkontakt* 0, *Gewicht* 50kg. Die Schlange ist allerdings das einzige Tier zu dem es kein „echtes“ Bild des Skeletts gibt. Das liegt daran, dass es keine seitlichen Abbildungen von ausgestreckten Schlangen gibt. Sie werden eigentlich immer gekrümmt dargestellt, da sonst das Bild sehr lang und schmal werden würde. Da aber versucht wurde, eine möglichst große Variation an Skeletten zu erheben, und ein Schlangenskelett, in der hier nötigen Auflösung, sehr einfach darzustellen ist, wurde trotzdem ein Bild erstellt. Dieses Bild enthält nur eine horizontale Linie knapp über dem unteren Bildrand, die den Rücken darstellen soll. Extremitäten und ersichtliche Punkte, an denen der Rücken in Hals oder Schwanz übergeht, gibt es bei Schlangen nicht.

Der Punkt mit dem zweitgrößten Abstand zum Mittelwert ist das Känguru (siehe Abbildung 4.7b). Zusätzlich zum Bild gibt es hier folgende Daten: *Tierklasse* Säugetier, *Flügel* nein, *Beine mit Bodenkontakt* 1, *Gewicht* 50kg

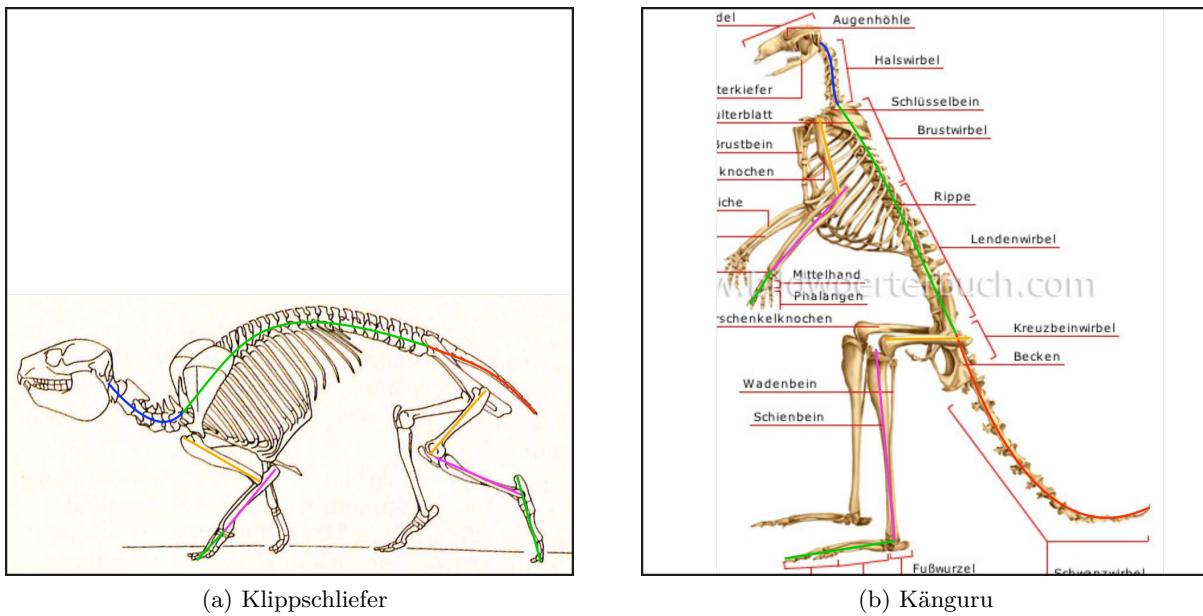


Abbildung 4.7.: Annotierte Bild des Skeletts eines Klippschließers (a) und eines Kängurus (b). Die Teile der Wirbelsäule und die Knochen der Extremitäten sind hier jeweils mit der gleichen Farbe markiert wie in der Visualisierung des Mittelwerts (Abbildung 4.6)

4.4. Analyse der Ergebnisse

Zu 29 Eingabedimensionen gibt es auch 29 Eigenvektoren mit Eigenwerten größer 0. Der kleinste Eigenwert ist 0,000001. Von den Eigenwerten sind 7 größer als 0,01.

Die Eigenwerte λ_i geben die Varianz der Normalverteilung entlang der Achse des zugehörigen Eigenvektors v_i an. Die Summe aller Varianzen $\sum_{i=0}^n \lambda_i$ ist ein Maß für den Informationsgehalt. Werden alle Eigenvektoren verwendet um die Daten darzustellen, so ist 100% der Information vorhanden, bei $m \leq n$ Eigenvektoren jeweils der Anteil $\sum_{i=0}^m \lambda_i / \sum_{i=0}^n \lambda_i$. Die ersten 6 Eigenvektoren enthalten 92,2% der Informationen, die ersten 10 enthalten 96,9% und die ersten 20 enthalten 99,8%.

Versucht man die Eingabedaten durch die Eigenvektoren mit den größten Eigenwerten anzunähern, so funktioniert das bei manchen Tieren ganz gut, wie z. B. beim Archaeopteryx (siehe Abbildung 4.8) schon mit den größten 6 Eigenvektoren, bei anderen aber eher schlechter (siehe Frosch, Abbildung 4.9).

Trotzdem sind die berechneten Eigenvektoren hilfreich um zufällige Punkte mit der Verteilung der Eingabebeispiele zu generieren (wie auch beschrieben in Abschnitt 3.5).

Um zu verstehen in welchem Bereich sich die zufällig generierten Punkte bewegen, ist es sinnvoll sich anzuschauen was die Hauptkomponenten „bedeuten“. In Abbildung 4.10 sind zwei Datenpunkte zu sehen, die auf der Koordinatenachse liegen, die zum Eigenvektor mit dem größten Eigenwert gehört. Ihr Wert in dieser Dimension ist jeweils die positive bzw. negative Standardabweichung. Die Varianz entlang eines Eigenvektors ist gegeben durch den Eigenwert. Die Standardabweichung ist also die Wurzel des entsprechenden Eigenwerts. Es ist zu sehen wie sich die Punkte verändern, wenn sie sich entlang dieser Achse bewegen bzw. welchen Einfluss der Eigenvektor ausübt.

Die Abbildung 4.10 zeigt, dass der Eigenvektor zum größten Eigenwert einen großen Einfluss auf die Halswirbelsäule hat. Bestätigt wird dies dadurch, dass die größten Einträge des Eigenvektors die y -Koordinaten des ersten, zweiten und dritten Kontrollpunkts der

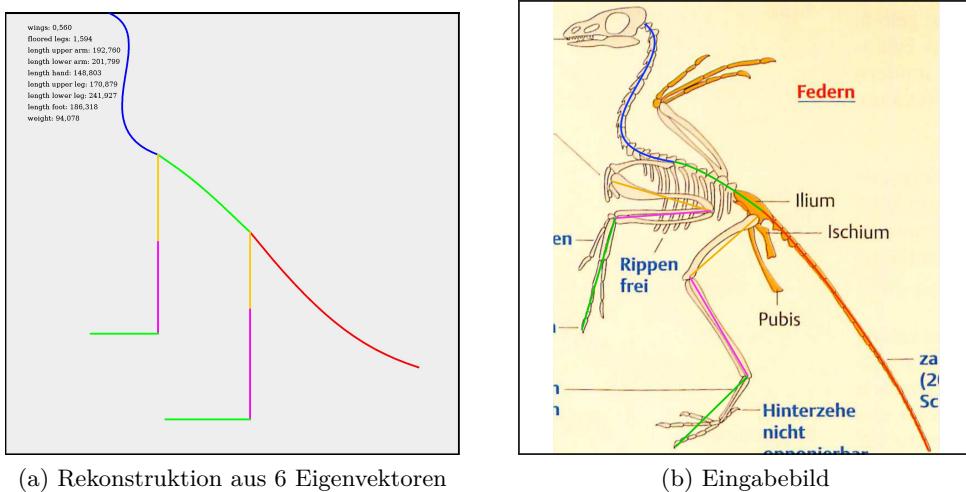


Abbildung 4.8.: Archaeopteryx nicht visualisierte Daten der Rekonstruktion: *Flügel* 0,56, *Beine mit Bodenkontakt* 1,594, *Gewicht* 94,1kg, Originalwert für das *Gewicht*: 1kg

Bézierkurve der Halswirbelsäule sind.

Abbildungen nach dem gleichen Schema für die sechs größten Eigenvektoren sind im Anhang in Abbildung A.6 zu finden.

Auswirkungen von Veränderungen auf den Eingabedaten

Bei allen Eingabedimensionen, außer der Position der Wirbelsäule, kann man die Frage stellen, ob sie nötig sind, oder ob sie die Ergebnisse der PCA eher verschlechtern. Deshalb wurden versuchsweise verschiedene (Kombinationen von) Merkmale weggelassen. Die Ergebnisse unterscheiden sich aber kaum von der PCA mit allen Daten.

Leider gibt es keine gute Möglichkeit die Qualität der Ergebnisse der PCA zu messen. Man könnte den Unterschied zwischen den Eingabedaten und den Rekonstruktionen aus den Linearkombinationen der Eigenvektoren mit den größten Eigenwerten messen. Da aber verschiedene Dimensionen fehlen, ist nicht klar wie dieser Unterschied einheitlich gemessen werden soll. Jedes Merkmal, das nicht verworfen wird, liefert dem Algorithmus, der später Skelette generieren soll, mehr Informationen. Deshalb wurde kein Merkmal verworfen.

Außerdem gibt es die Möglichkeit die Eingabedaten in mehrere Mengen aufzuteilen und diese einzeln zu analysieren. Hierbei gibt es zunächst das Problem, dass sich dann die Anzahl der Datenpunkte noch weiter reduziert, was die Ergebnisse nicht mehr repräsentativ macht.

Merkmale, die sich zur Unterteilung in Mengen anbieten würden, sind die diskreten, also *Flügel* und *Beine mit Bodenkontakt*. Sie sind auch klar als Cluster im Koordinatensystem der PCA ohne angepasste Skalierung zu erkennen (Abbildung 4.3 a). Getestet wurde die Aufteilung anhand der Werte für die Flügel, da sich dadurch nur zwei Gruppen ergeben. Tatsächlich liefert sie bessere Rekonstruktionen aus den größten Eigenvektoren. Das liegt aber natürlich in erster Linie daran, dass die zu untersuchende Datenmenge jeweils verkleinert wurde.

Ein weiteres Problem daran, die Daten in mehrere Mengen aufzuteilen, ist, dass dann keine Skelette mehr erzeugt werden können, die zwischen den beiden Gruppen liegen. Tatsächlich sehen die Datenpunkte, die zwischen den Gruppen erzeugt werden, aber relativ sinnvoll aus (siehe Abbildung 4.11). Auch das ist ein Argument dafür keine Aufteilung vorzunehmen.

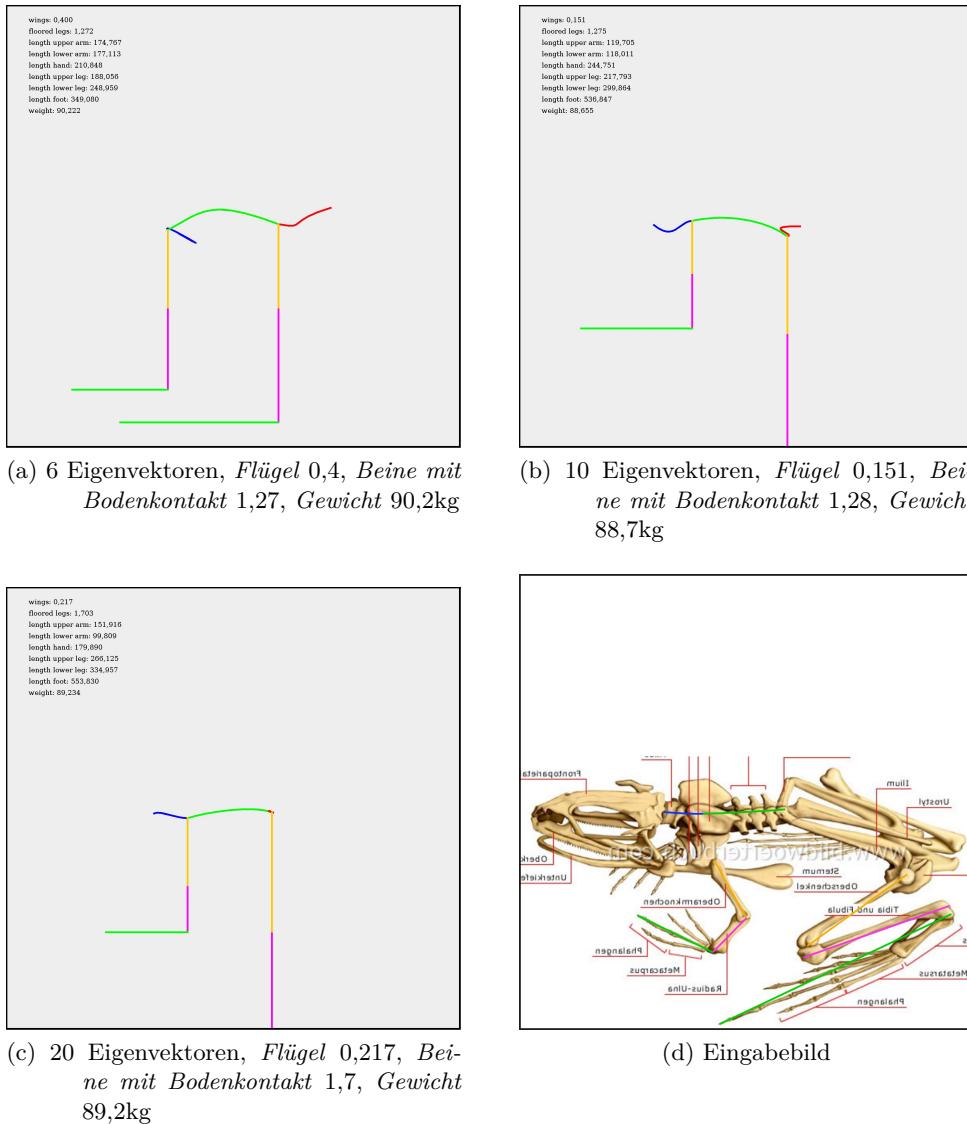


Abbildung 4.9.: Frosch, Rekonstruktionen aus den größten 6, 10 und 20 Eigenvektoren und das Eingabebild (d). Der Originalwert für das *Gewicht* ist 0,01kg.

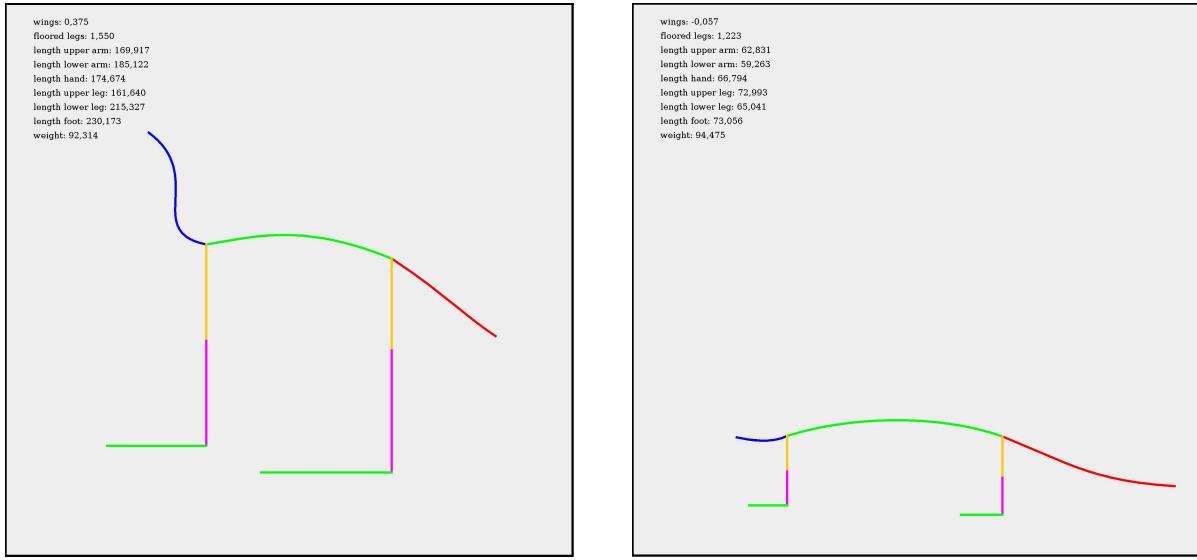


Abbildung 4.10.: Datenpunkte im PCA-Koordinatensystem. Die Koordinate für den Eigenvektor zum größten Eigenwert nimmt den Wert der positiven bzw. negativen Standardabweichung σ an, alle anderen sind null. Für Abbildungen zu den größten sechs Eigenvektoren siehe Abbildung A.6 im Anhang

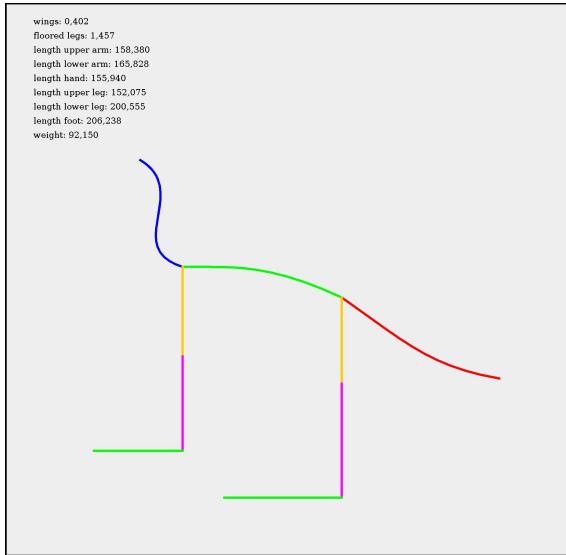


Abbildung 4.11.: Visualisierung eines Punktes, der, projiziert auf die ersten beiden Eigenvektoren, zwischen den Eingabedaten mit Flügeln und jenen ohne Flügel liegt. Die Werte, die auf den jeweiligen Achsen angenommen werden, sind $-0,4$ für den ersten Eigenvektor und $0,2$ für den zweiten. Werte für die ursprünglichen Merkmale, die nicht visualisiert sind, sind folgende: Flügel 0,402, Beine mit Bodenkontakt 1,46 und Gewicht 92,2kg.

4.5. Bedingte Verteilungen

Aus den Ergebnissen der PCA lassen sich sehr gut zufällige Skelette erzeugen. Es ist aber schwer gezielt Eigenschaften festzulegen.

Bedingungen an explizit erhobene Merkmale

Der erste Schritt dies zu erreichen ist, Eigenschaften, die so schon in den erhobenen Daten vorkommen, auf spezifische Werte festzulegen. Das können beispielsweise die Anzahl der Beine sein oder ob das Skelett Flügel haben soll.

Dazu kann man, statt die ursprünglichen Daten und deren Verteilung zu verwenden, die entsprechenden bedingten Verteilungen bilden. Dazu muss der bedingte Mittelwert und die bedingte Kovarianzmatrix, wie in [14, S. 116 f.] beschrieben, bestimmt werden.

Seien

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \text{ und } \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}.$$

Der Zufallsvektor x enthält n normalverteilte Zufallsvariablen, $q \leq n$ Variablen im Vektor x_1 und $n - q$ in x_2 . Die Einträge in μ sind die jeweils zugehörigen Mittelwerte und Σ ist die Kovarianzmatrix. Die Verteilung für x_1 unter der Bedingung, dass $x_2 = b$, hat den Mittelwert $\bar{\mu}$ und Kovarianzmatrix $\bar{\Sigma}$ mit

$$\bar{\mu} = \mu_1 + \Sigma_{12} \Sigma_{22}^{-1} (b - \mu_2), \quad \bar{\Sigma} = \Sigma_{11} - \Sigma_{12} \Sigma_{12}^{-1} \Sigma_{21}.$$

Verwendet man nun $\bar{\Sigma}$ als Eingabe für die PCA, so erhält man nur noch Daten für Skelette unter den vorher festgelegten Bedingungen b . Zu beachten ist hier, dass die Eigenvektoren sich im Vergleich zur PCA ohne Bedingungen verändern. Falls man diese also zur Bestimmung eines Datenpunktes verwendet hat (z. B. in einer Benutzeroberfläche), so müssen die Werte für die neuen Eigenvektoren neu ausgerechnet werden.

Erzeugt man nun in diesem bedingten Raum zufällige Beispiele, zeigt sich sehr deutlich, dass das Festlegen von nur einer Dimension auch die anderen Dimensionen stark einschränken kann. Legt man z. B. fest, dass das Skelett keine Flügel haben soll, so sind sich die Wirbelsäulen der Skelette, die dann bedingt zufällig generiert werden, sehr ähnlich (siehe Abbildung 4.12a).

Um das zu umgehen, wird die Eingabe als Intervall aufgefasst, aus dem zufällig ein Wert gezogen wird. Legt der Benutzer z. B. einen ganzzahligen Wert fest, so wird ein kleiner zufälliger Wert aufaddiert oder abgezogen. So wird nicht immer genau der gleiche Wert verwendet. Zwei Beispiele sind in den Abbildungen 4.12 b und c gezeigt. Zusätzlich ist dieses Vorgehen hilfreich, weil dann vom Benutzer nicht verlangt wird Bedingungen für *Flügel* oder *Beine* anzugeben, die nicht ganzzahlig sind, um mehr Variation zu bekommen.

Schaut man sich nun die erzeugten Wirbelsäulen mit einem Paar tragender Beine an (Abbildung 4.12c), so liegen sie alle recht nah an der mittleren Wirbelsäule (Abbildung 4.6). Das wirkt zunächst überraschend, da die meisten Tiere in den erhobenen Beispielen, die zwei Beine haben, Vögel sind. Sie haben eine Wirbelsäule, die hoch über dem Boden liegt und relativ aufrecht ist. Dann gibt es noch Känguru und Tyrannosaurus Rex. Bei ihnen liegt die Wirbelsäule ähnlich. Deshalb könnte man eine mittlere Wirbelsäule erwarten, die ebenfalls diese Eigenschaften besitzt.

Es muss aber auch einen Übergang von bodennahen Vierbeinern wie Krokodilen oder Fröschen über Zweibeiner zu Fischen ohne Beine geben, auch wenn es dafür in der Natur eher wenige Beispiele gibt. In den erhobenen Daten sind das nur der Seehund und die Ohrenrobbe. Das ist der Grund, weshalb die mittlere Wirbelsäule für Zweibeiner verhältnismäßig flach verläuft.

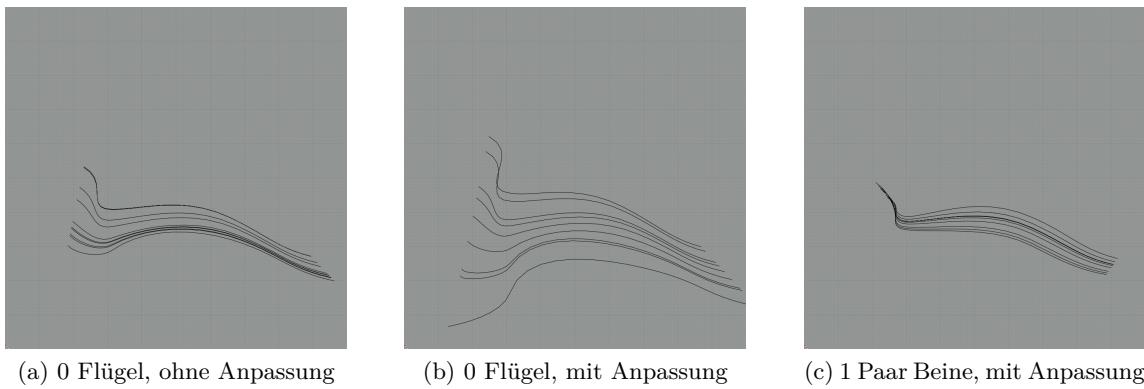


Abbildung 4.12.: Jeweils 10 bedingt zufällig generierte Wirbelsäulen. (a) und (b) ohne Flügel, (c) mit einem Paar Beine. In (b) und (c) wird ein zufälliger Wert aus $[-0,5, 0,5]$ auf die Bedingung (Flügel = 0 bzw. Beine = 1) aufaddiert.

Bedingungen an implizit enthaltene Merkmale

Nun möchte man vielleicht Bedingungen an das zu generierende Skelett stellen, die nicht schon genau so in den erhobenen Daten repräsentiert sind. Bedingungen an Dimensionen, die gar nicht erhoben wurden, können entweder, unabhängig von der PCA, in den Ersetzungsregeln oder den Parametern der Grammatik (siehe Abschnitt 5.4) erzwungen werden, wie z. B. die Anzahl der Flossen, oder sie müssen in die Erhebung eingefügt werden.

Bedingungen, die schon implizit in den erhobenen Daten enthalten sind, sind beispielsweise die Schwanz- oder Halslänge. Um an sie Bedingungen stellen zu können, könnte man ebenfalls eine zusätzliche Dimension einfügen. Sie kann einfach aus den schon bestehenden Dimensionen errechnet werden. Dabei wäre aber das Problem, dass nur die Dimensionalität der Daten erhöht wird, nicht aber die eingegebenen Informationen. Es wären also mehr Eingabebeispiele nötig, nur dafür, dass die PCA eine offensichtliche Korrelation der Dimensionen erkennt.

Die bessere Alternative ist die Eingabedimensionen der PCA umzuparametrisieren. Der Abstand vom Anfangs- zum Endpunkt des Schwanzes in x-Richtung, ist z. B. implizit in der Differenz der x-Koordinaten des ersten und letzten Kontrollpunktes der Bézierkurve des Schwanzes enthalten. Ersetzt man nun den absoluten x-Wert des letzten Kontrollpunktes durch den Abstand in x-Richtung zum ersten Kontrollpunkt, so lässt sich diese Länge ganz einfach als Bedingung an die PCA stellen.

Betrachtet man nun konkret den Wunsch die Schwanzlänge festzulegen, ist das nicht ganz so einfach. Der Abstand vom Anfangs- zum Endpunkt des Schwanzes in x- und y-Richtung lässt sich zwar leicht festlegen, diese Längen sagen aber im Allgemeinen noch nicht viel über die tatsächliche Länge der Bézierkurve aus. Verlangt man z. B. eine Länge von 0px in x-Richtung, so haben die generierten Datenpunkte einen Schwanz der zwar auf gleicher Höhe beginnt und endet, aber trotzdem vorhanden ist und einen kleinen Bogen beschreibt. Um die wirkliche Länge des Schwanzes zu messen, müsste man also noch mehr Aufwand in die Umparametrisierung stecken oder doch eine zusätzliche Dimension für die PCA in Kauf nehmen.

5. Generierung von Skeletten

In diesem Kapitel werden die einzelnen Bestandteile zusammengefügt, die zur Generierung eines Skeletts notwendig sind. Abschnitt 5.1 gibt einen Überblick über den Ablauf des kompletten Algorithmus. Fehlende Details werden in den darauffolgenden Abschnitten beschrieben.

In Abschnitt 5.2 geht es um die Einzelteile aus denen ein fertiges Skelett besteht. Wie die Anzahl der zu generierenden Wirbel und Rippen bestimmt werden, wird in Abschnitt 5.3 beschrieben. Dann wird das Regelwerk zur Generierung der Einzelteile des Skeletts in Abschnitt 5.4 vorgestellt. Die Positionierung der Extremitäten ist mit etwas mehr Aufwand verbunden und lässt viel Spielraum für Variationen. Deshalb widmet sich Abschnitt 5.5 den Extremitäten. Zuletzt geht es in Abschnitt 5.6 um die Erzeugung eines konkreten 3D-Modells.

5.1. Überblick über den Ablauf der Generierung

Generierung eines Skeletts

0. Lies die Benutzereingabe ein.
1. Führe eine PCA auf den gegebenen Beispielskeletten durch.
2. Bestimme einen Punkt q im PCA-Raum und auf dessen Grundlage die Parameter für die Grammatik G .
3. Verwende die Grammatik G um die Bestandteile des Skeletts zu erzeugen.
4. Spiegle alle Elemente, die nicht auf der Wirbelsäule liegen.
5. Generiere ein 3D-Modell.

Bevor der Algorithmus beginnt wird ausgelesen was der Benutzer für Zusatzbedingungen, z. B. an die Anzahl der Flügel, gestellt hat. Welche Eingaben möglich sind und wie diese verarbeitet werden, wird in Abschnitt 6.1 genauer erläutert.

Im ersten Schritt des Algorithmus wird eine PCA auf den gegebenen Beispieldaten ausgeführt (siehe Kapitel 4). Dazu werden zunächst die Beispieldaten eingelesen und dann der

Basiswechsel in das neue Koordinatensystem, im Folgenden als *PCA-Raum* bezeichnet, berechnet. Im PCA-Raum kann nun ein Punkt q bestimmt werden, der als Grundlage für die weitere Generierung verwendet wird. Dieser Punkt kann entweder (bedingt) zufällig oder auch nach anderen Vorgaben ausgewählt werden.

Auf Grundlage von q und Teilen der Benutzereingabe werden die Parameter bestimmt, die im nächsten Schritt von der kontextfreien Grammatik G verwendet werden. Dazu wird q in das Koordinatensystem zurücktransformiert, in dem Merkmale, wie die Lage der Wirbelsäule, einfach abgelesen werden können. Die Grammatik G wird in Abschnitt 5.4 definiert. Mit ihr werden die Einzelteile des Skeletts (siehe Abschnitt 5.2) generiert. Gleichzeitig werden auch Position und Ausdehnung der terminalen Elemente festgelegt.

Danach werden alle Elemente, die nicht auf der Wirbelsäule liegen, gespiegelt. Das sind Extremitäten, Rippen und Schulterblatt. (Technische Details zur Spiegelung der Transformationsmatrizen sind in Abschnitt 7.3 zu finden.)

Zum Schluss wird ein 3D-Modell erzeugt. Dazu wird für jedes terminale Element, also jeden Knochen, ein eigenes Modell erzeugt und dann an der richtigen Position in das 3D-Modell des gesamten Skeletts eingefügt (siehe Abschnitt 5.6).

5.2. Bestandteile eines Skeletts

Ein Skelett besteht aus Knochen und Gelenken. Zwei Knochen sind jeweils durch ein Gelenk miteinander verbunden. Im Folgenden werden sowohl Knochen als auch Gelenke genauer beschrieben.

Knochen

Jeder Knochen hat sein eigenes lokales Koordinatensystem und wird zunächst als Quader dargestellt. Der Ursprung des Koordinatensystems befindet sich in einer Ecke des Quaders. Zur Darstellung eines Knochens sind also seine Ausdehnung in alle drei Raumrichtungen (die Kantenlängen des Quaders) und die Position und Orientierung des Ursprungs im globalen Koordinatensystem erforderlich.

Die Knochen sollten eine Hierarchie (einen Baum) bilden, da das von Algorithmen für Animationen so erwartet wird (siehe Abschnitt 3.2). Da der Wirbeltierbauplan aus Abbildung 2.2 verwendet wird, ist das auch möglich. Im Allgemeinen ist ein Skelett aber weder zusammenhängend noch ohne Kreise.

Für eine Hierarchie ist ein oberstes Element nötig, die Wurzel des Baums. Dafür bietet sich ein Knochen in der Nähe des Schwerpunkts an. Oft wird hierfür die Hüfte verwendet. Da aber nicht jedes Wirbeltier eine Hüfte besitzt und es die Generierung vereinfacht, wird als Wurzelknochen ein Knochen ohne Ausdehnung in der Mitte der Rückenwirbelsäule verwendet. Dieser Knochen wird im Folgenden als *Wurzelknochen* bezeichnet. Liegt Knochen B in der Hierarchie direkt unter Knochen A , so ist A der *Elternknochen* von B und B ein *Kindknochen* von A .

Es ist sinnvoll für jeden Knochen nicht die Position im globalen Koordinatensystem anzugeben, sondern seine Position im Koordinatensystem des Elternknochens. Verfolgt man den Pfad von einem Knochen zurück zum Wurzelknochen, so kann die Position im globalen Koordinatensystem trotzdem ausgerechnet werden.

Für die Darstellung werden Transformationsmatrizen mit homogenen Koordinaten verwendet. Genauere Informationen zu diesen Transformationsmatrizen und wie sie in verschiedenen Situationen berechnet werden können sind in Absatz 7.3 zu finden.

Gelenke

Ein Gelenk ist, wie auch in der Natur, ein Verbindungsstück zwischen zwei Knochen. Es legt fest wie die beiden Knochen sich relativ zueinander bewegen können. Im Gegensatz zu echten Gelenken haben Gelenke hier aber keine Ausdehnung. Sie werden am Ende im 3D-Modell nicht dargestellt.

Ein Gelenk wird im Koordinatensystem des Elternknochens dargestellt. Es wird beschrieben durch seine Position im Elternkoordinatensystems und Bewegungseinschränkungen für den Kindknochen. Ein Gelenk kann null bis zwei Freiheitsgrade haben. Haben alle Winkel einen Wert von 0° , hat das Kindelement die gleiche Ausrichtung wie das Eltern-element. Die meisten Knochen werden aber direkt in ihrer Endposition generiert, sodass zusätzliche Bewegungseinschränkungen gar nicht nötig sind. Nur bei der Positionierung der Extremitäten (siehe Abschnitt 5.5.2) spielen die Gelenke eine Rolle. In Abbildung 5.3 sind alle Gelenke der Extremitäten dargestellt. Der Einfachheit halber haben sie alle nur einen Freiheitsgrad.

In Abschnitt 5.4 wird die Grammatik vorgestellt, die verwendet wird um die Einzelteile des Skeletts zu generieren. Die Menge der Terminalsymbole enthält nur Knochen, keine Gelenke. Gelenke werden als Bestandteile des Elternknochens dargestellt und zusammen mit ihm generiert. Die Transformationsmatrix des Kindelements, lässt sich dann aus den Informationen zum Elternknochen und der Ausrichtung seines Gelenks berechnen.

5.3. Wirbel und Rippen

Bevor die Grammatik verwendet werden kann, müssen verschiedene Dinge festgelegt werden. Dazu gehören die Anzahl der Wirbel auf den einzelnen Teilen der Wirbelsäule und die Anzahl der Rippen. Auch die Anzahl der Vorder- und Hinterextremitäten muss festgelegt werden. Siehe dazu Abschnitt 5.5.

Wirbel

Die Anzahl der Wirbel orientiert sich an echten Wirbeltierskeletten (siehe Absatz 2.1). Auf dem Hals werden 7 Wirbel generiert, außer das Tier hat Flügel. Dann wird angenommen, dass ein Vogel generiert wird. Dementsprechend wird die Anzahl dann zufällig zwischen 10 und 30 gewählt. Auf dem Rücken liegen 25 Wirbel und auf dem Schwanz 5 bis 20. Da der Wurzelknochen in der Mitte der Rückenwirbelsäule liegt (oder zumindest ungefähr, da Bezierkurve bei 0,5 ausgewertet), wird die Rückenwirbelsäule in zwei Teile geteilt. Es werden 13 Wirbel auf der vorderen Hälfte und 12 auf der hinteren generiert.

Eine Bézierkurve ist ein Weg im \mathbb{R}^2 . Sie ist parametrisiert auf $[0, 1]$ und im Allgemeinen nicht nach Bogenlänge parametrisiert. Das heißt die Geschwindigkeit, mit der die Kurve durchlaufen wird, ist nicht konstant. Wertet man die Kurve also bei 0,5 aus, wurde nicht notwendigerweise die Hälfte der Strecke zurückgelegt.

Für die Generierung von n Wirbeln wurde die Kurve einfach an den Stellen $\frac{i}{n}$ ausgewertet, für $0 \leq i \leq n$. Die Länge des i -ten Wirbels ergibt sich dann einfach aus dem Abstand zwischen den Stellen $\frac{i-1}{n}$ und $\frac{i}{n}$ auf der Bézierkurve. Durch den oben genannten Effekt variiert dann die Länge der Wirbel über den Kurvenverlauf. Die Wirbel werden auf der Bézierkurve aufgereiht und so positioniert, dass sie an den ausgewerteten Stellen aneinander stoßen.

Wird die Bézierkurve B nach Bogenlänge umparametrisiert, so erhält man als neuen Parameter:

$$p(t) = \int_0^t \|B'(s)\| \, ds.$$

Die umparametrisierte Bézierkurve ist dann $t \mapsto B(p(t))$.

Das obenstehende Integral ist jedoch schwer zu berechnen, da im Allgemeinen keine Stammfunktion des Integranden zur Verfügung steht. Es gibt jedoch numerische Methoden, mit denen das Problem gelöst werden kann. [49]

Betrachtet man echte Wirbeltiere, so sind keine einfachen Regeln für die Länge ihrer Wirbel ersichtlich. Es gibt beispielsweise eine Studie, die die Beschaffenheit der Wirbel von Mäusen untersucht [42]. In dieser Studie auf Seite 19, Abbildung 5, ist sehr gut zu sehen, dass die Länge der Wirbel allein bei Mäusen im Verlauf der Wirbelsäule sehr stark schwankt.

Rippen

Die Anzahl der Rippen und auch die Ausdehnung des Brustkorbs variiert zwischen Wirbeltieren sehr stark. Es gibt Tiere, die an jedem Wirbel der Rückenwirbelsäule Rippen haben und es gibt Tiere die haben nur ein paar wenige auf dem vorderen Teil. Einige Beispiele mit unterschiedlich vielen Rippen sind auch unter den Beispielbildern für die PCA in Abbildung A.1 im Anhang zu finden.

Deshalb wird die Anzahl der Rippen r zufällig aus dem Intervall $[0, n]$, mit $n = \text{Anzahl der Rückenwirbel}$, bestimmt. Die Rippen beginnen an aufeinanderfolgenden Wirbeln der Rückenwirbelsäule, Lücken sind nicht erlaubt. Außerdem starten sie immer am ersten Wirbel der Rückenwirbelsäule, wie auch auf dem Grundbauplan in Abbildung 2.2 zu sehen ist. Aus n und r ergeben sich also die Eingabeparameter für die Grammatik, die angeben wieviele Wirbel insgesamt auf der vorderen bzw. hinteren Hälfte der Wirbelsäule zu finden sind (w_{rv}, w_{rh}) und wieviele davon jeweils Rippen haben (w_{rv}^r, w_{rh}^r) (siehe Abschnitt 5.4).

5.4. Aufbau als Grammatik

Das Regelwerk für die Generierung der Knochen des Skeletts ist durch eine kontextfreie Grammatik (siehe 3.4) gegeben. Da Wirbeltierskelette bilateralsymmetrisch aufgebaut sind (siehe Abschnitt 2.1), soll dies auch hier erzwungen werden. Mit der Grammatik werden nur die Knochen auf einer Seite der Wirbelsäule und die Knochen direkt auf der Wirbelsäule generiert. Die Knochen auf der anderen Seite der Wirbelsäule werden anschließend durch Spiegelung erzeugt.

Kontextfreie Grammatik

Die verwendete kontextfreie Grammatik $G = (\Sigma, N, S, P, p)$ unterstützt Klammerausdrücke, um die Baumstruktur des Skeletts darzustellen (so wie sie auch bei L-Systemen verwendet werden, siehe dazu Abschnitt 3.4). Außerdem bekommt sie als zusätzliche Eingabe eine Menge von Parametern p aus \mathbb{N}_0 , die in den Produktionen P verwendet werden.

$$\begin{aligned}\Sigma &= \{(,), \text{Wurzelknochen}, \text{Wirbel}, \text{Rippe}, \text{Schädel}, \\ &\quad \text{Schulterblatt}, \text{Oberarm}, \text{Unterarm}, \text{Hand}, \\ &\quad \text{Beckenknochen}, \text{Oberschenkel}, \text{Unterschenkel}, \text{Fuß}\} \\ N &= \{\text{Skelett}, \text{Vorderteil}, \text{Hinterteil}, \text{Schultergürtel}, \text{Beckengürtel} \\ &\quad \text{Hals}, \text{Vorderextremität}, \text{Hinterextremität}\} \\ S &= \text{Skelett} \\ p &= \{w_{rv}, w_{rv}^r, w_{rh}, w_{rh}^r, h, s, w_h, w_s, r, e_s, e_b\}\end{aligned}$$

Die Parameter werden im vorherigen Schritt des Algorithmus bestimmt (siehe Überblick 5.1). Sie haben folgende Bedeutungen:

- w_{rv} : Anzahl der Wirbel auf der vorderen Hälfte der Rückenwirbelsäule, mit $w_{rv} \geq 3$
- w_{rv}^r : Anzahl der Wirbel mit Rippe auf der vorderen Hälfte der Rückenwirbelsäule, mit $w_{rv}^r \leq w_{rv}$
- w_{rh} : Anzahl der Wirbel auf der hinteren Hälfte der Rückenwirbelsäule
- w_{rh}^r : Anzahl der Wirbel mit Rippe auf der hinteren Hälfte der Rückenwirbelsäule, mit $w_{rh}^r \leq w_{rh}$
- $h \in \{0, 1\}$ gibt an, ob es eine Halswirbelsäule gibt.
- $s \in \{0, 1\}$ gibt an, ob es eine Schwanzwirbelsäule gibt.
- w_h : Anzahl der Wirbel auf der Halswirbelsäule, mit $h > 0 \Rightarrow w_h \geq 7$ und $h = 0 \Rightarrow w_h = 0$
- w_s : Anzahl der Wirbel auf der Schwanzwirbelsäule, mit $s = 0 \Rightarrow w_s = 0$
- $e_s \in \{0, 1\}$ gibt an, ob eine Vorderextremität und damit auch ein Schultergürtel generiert werden soll.
- $e_b \in \{0, 1\}$ gibt an, ob eine Hinterextremität und damit auch ein Beckengürtel generiert werden soll.

Wie die Anzahl der Wirbel auf Hals (w_h), Schwanz (w_s) und Rücken ($w_{rv} + w_{rv}^r + w_{rh} + w_{rh}^r$) bestimmt werden, wird in Abschnitt 5.3 erklärt. Wieviele Rippen ($w_{rv}^r + w_{rh}^r$) generiert werden wird zufällig bestimmt (siehe ebenfalls Abschnitt 5.3). Ob eine Hals- bzw. Schwanzwirbelsäule (h, s) generiert wird, hängt davon ab, ob dieser Teil der Wirbelsäule in derjenigen Wirbelsäule, die durch Punkt q im PCA-Raum vorgegeben wird, vorhanden ist. Und schließlich werden *Schultergürtel* und *Beckengürtel* nur generiert, wenn das Skelett Vorder- bzw. Hinterextremitäten besitzt (e_s, e_b). Um das zu bestimmen wird u. a. betrachtet, welche Werte der Punkt q in den Merkmalen *Beine mit Bodenkontakt* und *Flügel* annimmt. Genaueres dazu ist in Abschnitt 5.5 zu finden.

Im folgenden sind alle Produktionen aus P aufgeführt.

$$\begin{aligned}
 \text{Skelett} &\rightarrow \text{Wurzelknochen } (\text{ Vorderteil }) \text{ Hinterteil} \\
 \text{Vorderteil} &\rightarrow \text{Wirbel}^{w_{rv} - (w_{rv}^r - 1)^+ - 1} \\
 &\quad [\text{Wirbel (Rippe)}]^{(w_{rv}^r - 1)^+} \\
 &\quad \text{Wirbel (Schulterpartie)} \\
 &\quad \text{Hals} \\
 \text{Schulterpartie} &\rightarrow \text{Rippe}^{\min(1, w_{rv}^r)} \text{ Schultergürtel}^{e_s} \\
 \text{Schultergürtel} &\rightarrow \text{Schulterblatt } \text{Vorderextremität} \\
 \text{Vorderextremität} &\rightarrow \text{Oberarm Unterarm Hand} \\
 \text{Hals} &\rightarrow \text{Wirbel}^{w_h} \\
 &\quad \text{Schädel} \\
 \text{Hinterteil} &\rightarrow [\text{Wirbel (Rippe)}]^{w_{rh}^r} \\
 &\quad \text{Wirbel}^{w_{rh} - w_{rh}^r} \text{ (Beckengürtel)} \\
 &\quad \text{Wirbel}^{w_s} \\
 \text{Beckengürtel} &\rightarrow [\text{Beckenknochen } \text{ Hinterextremität}]^{e_b} \\
 \text{Hinterextremität} &\rightarrow \text{Oberschenkel Unterschenkel Fuß}
 \end{aligned}$$

Der Ausdruck A^x mit $A \in (\Sigma \cup N)^+$ bedeutet, dass A genau x -mal hintereinander vorkommt. Der Ausdruck $A^{(x)^+}$ kann ersetzt werden durch $A^{\max(0, x)}$. Die angegebene

Anzahl x wird also nur verwendet, wenn sie positiv ist. Sonst wird x durch 0 ersetzt. Dadurch kann es passieren, dass leere Klammern, also Ausdrücke der Form $\alpha () \beta$ mit $\alpha, \beta \in (\Sigma \cup N)^*$, entstehen. Leere Klammern haben keine Bedeutung. Ein Ausdruck $\alpha () \beta$ ist hier also gleichbedeutend zu $\alpha \beta$.

Für jedes Nichtterminal gibt es in P genau eine Produktion und alle Parameter p sind vor der Anwendung der Grammatik G festgelegt. Das bedeutet G ist deterministisch und es wird genau ein Wort erzeugt. Die Darstellung als Grammatik ist also gar nicht unbedingt nötig. Die Grammatik sorgt jedoch dafür, dass Teile des Skeletts einfach verfeinert werden können. Es ist z.B. leicht G so anzupassen, dass der Fuß kein Terminalsymbol mehr ist, sondern in kleinere Einzelteile verfeinert werden kann. Außerdem können natürlich auch zusätzliche Produktionen für bestehende Nichtterminalsymbole eingefügt werden, z. B. für einen anders gearteten Schultergürtel.

Auch im Programmcode gibt es nichtterminale Elemente, die im Wesentlichen anhand der oben vorgestellten Regeln ersetzt werden. Im Programmaufbau ist vorgesehen, dass aus mehreren passenden Produktionen zufällig eine ausgewählt wird. Es ist also leicht den Code um weitere Regeln zu erweitern. Siehe dazu auch den Abschnitt 7.1 zum Aufbau der Software.

Positionierung der terminalen Elemente

Während die Bestandteile des Skeletts durch die Grammatik G generiert werden, wird auch gleichzeitig die Positionen und Ausdehnungen der erzeugten Terminalsymbole festgelegt.

Die Positionen der meisten terminalen Bestandteile sind durch den Verlauf der Wirbelsäule festgelegt. Dieser Verlauf wird im Wesentlichen durch den gewählten Punkt q im PCA-Raum bestimmt. Es kann jedoch sein, dass die Wirbelsäule, an den Punkten, an denen ihre Teile zusammenstoßen, nicht differenzierbar ist. In diesem Fall wird der Verlauf etwas angepasst (siehe Abschnitt 7.4.2).

Wirbel werden entlang der Wirbelsäule positioniert (näheres dazu in Abschnitt 5.3) und Rippen, Schulterblätter und Beckenknochen so, dass sie an den zugehörigen Wirbeln ansetzen. Der Schädel befindet sich am Ende der Halswirbelsäule. Kompliziertere Berechnungen werden nur zur Positionierung der Extremitäten gebraucht. Diese werden in Abschnitt 5.5 genauer ausgeführt.

Die Länge der Knochen der Extremitäten ist durch q festgelegt und die Länge der Wirbel ergibt sich durch ihre Positionierung auf der Wirbelsäule. Alle anderen Ausdehnungen von terminalen Elementen sind relativ beliebig im Code festgelegt und sind bei jedem erzeugten Skelett gleich.

Darstellung als Baum

Wie auch in Abschnitt 5.2 beschrieben, gibt es auf den Knochen bzw. den terminalen Elementen eine Baumstruktur. Diese wird in G durch die Klammern im generierten Wort dargestellt. Das bedeutet, dass auch nichtterminale Elemente zeitweise Teil dieser Baumstruktur sind. Nichtterminale Elemente haben aber weder Position noch Ausdehnung. Hätte ein Nichtterminal ein terminales Kindelement im Baum, so wäre nicht klar, wie dieses positioniert werden müsste, da seine Position von der seines Elternelements abhängt. Aus diesem Grund haben Nichtterminale niemals Kindelemente.

Um die Baumstruktur besser zu visualisieren, kann das erzeugte Wort w auch als Graph bzw. Baum B dargestellt werden.

Der Baum $B = (V, E)$ hat eine Knotenmenge $V \subset (\Sigma \cup N)$ und eine Kantenmenge $E \subset (\Sigma \cup (N \setminus S)) \times \Sigma$. Eine Kante (a, b) führt vom Knoten a zu seinem Elternknoten b . Nichtterminale Knoten können keine Kindknoten haben, sind also Blätter. Für jeden nichtterminalen Knoten v gibt es mindestens eine Ersetzungsregel aus P , die v durch einen

Baum B_v ersetzt. In diesem Prozess wird v aus B gelöscht und alle Knoten und Kanten von B_v in B eingefügt. Außerdem wird eine Kante zwischen der Wurzel von B_v und dem Elternknoten von v eingefügt, falls v nicht die Wurzel von B war. Wenn B vor der Anwendung der Ersetzungsregel ein Baum war, so ist G nach der Ersetzung also immer noch ein Baum.

Die Skelettgenerierung startet konkret mit einem Graphen B , der nur aus dem nichtterminalen Knoten *Skelett* besteht. Der initiale Graph B ist also ein Baum.

Die Ersetzungsregeln, die im weiteren Verlauf verwendet werden, sind in Abbildung 5.1 dargestellt. Elliptische Knoten stellen nichtterminale Knoten dar, eckige sind terminale Knoten. Die türkisfarbenen Kanten, die von einem nichtterminalen Knoten v ausgehen, führen zu den Knoten des größten Baumes B_v , durch den v bei Anwendung der Ersetzungsregel ersetzt werden kann. Die grauen Kanten, die von nichtterminalen Knoten ausgehen, führen zum jeweiligen Elternelement des Knotens. Terminale Knoten liegen auf durchgehenden, gerichteten Kurven. Die Richtung der Kurve gibt jeweils an, in welcher Richtung das dazugehörige Elternelement liegt.

Aufeinanderfolgende Wirbel sind in der Abbildung teilweise zu einem Knoten vereinfacht. Alle Benennungen von Wirbeln stehen für das Terminal „Wirbel“. Sie sind nur der Übersichtlichkeit halber unterschiedlich benannt. Auch die Rippen sind eigentlich separate Knoten, die jeweils Kindknoten genau eines Rückenwirbels sind.

Abbildung 5.1 zeigt nicht den Baum B , sondern visualisiert die Produktionen P und ihre Auswirkungen auf B . Alle schwarzen und grauen Kanten, deren Anfangs- und Endknoten in V enthalten sind, bilden die Kantenmenge E von G . Bei Anwendung einer Produktion verschwindet der dazugehörige, nichtterminale Knoten v aus B und dafür wird eine Teilmenge der Knoten v_i eingefügt, die durch türkisfarbene Kanten (v, v_i) mit v verbunden waren.

Gegebenenfalls werden also nichtterminale Knoten nur durch Teilmengen der hier angegebenen Knoten ersetzt. Dies ist durch die oben angegebenen Produktionen P und die Parameter p festgelegt. Schulterblatt und Beckenknochen werden beispielsweise nur dann generiert, wenn es auch Extremitäten gibt, die daran ansetzen sollen. Der Schultergürtel ist Kindknoten der Rippe am Schulterwirbel. Existiert diese Rippe jedoch nicht, ist er direkter Kindknoten des Schulterwirbels. Bei allen anderen Knoten gilt: fehlt das Elternelement, so wird dieser Knoten nicht eingefügt.

5.5. Extremitäten

Die Bestandteile einer Extremität sind durch den Grundbauplan (Abbildung 2.2) vorgegeben und die Länge der Bestandteile durch die entsprechenden Merkmale des PCA-Datenpunkts q . Die Schwierigkeit besteht nun darin, in den entsprechenden Ersetzungsregeln (siehe Abbildung 5.1) auch die Positionierung vorzunehmen.

Positionierung

Das Skelett soll in einer Art Ruheposition dargestellt werden (siehe auch Abschnitt 4.1). Im Allgemeinen ist aber nicht klar wie die Ruheposition einer Extremität aussieht. Das ist schon allein daran zu erkennen, dass auf Darstellungen von Wirbeltierskeletten Flügel manchmal ausgestreckt und manchmal eingefaltet sind. Auch Beine sind meist so angeordnet, dass es aussieht als würde das entsprechende Tier gerade laufen. Dies ist auf Abbildung 4.7a am Beispiel des Klippschließers sehr gut zu sehen.

Wie in Kapitel 4 zur PCA schon erwähnt, ist es deshalb auch schwer möglich die Ausrichtung der Extremitäten bzw. die Winkel an den Gelenken zu erheben und als zusätzliche

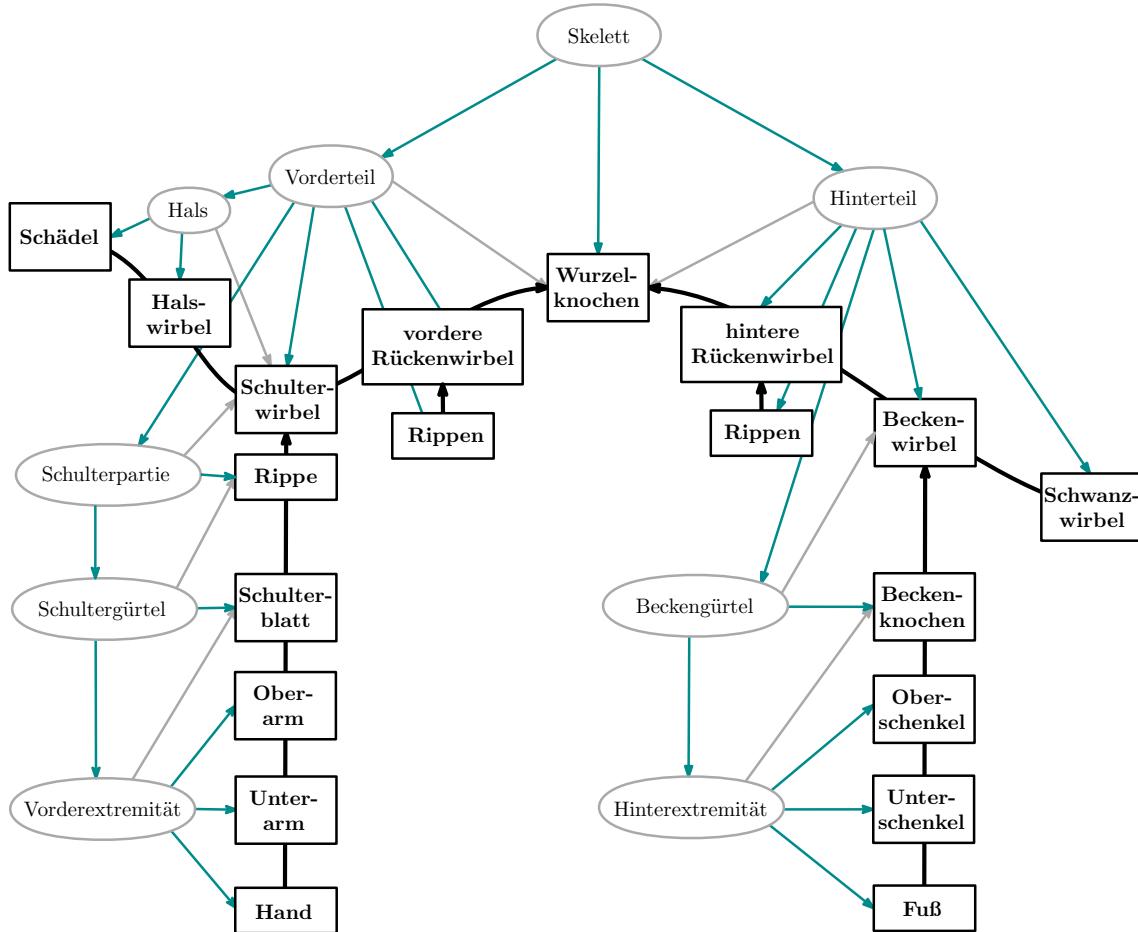


Abbildung 5.1.: Darstellung der Ersetzungsregeln bzw. Produktionen der Grammatik G als Graph. Elliptische Knoten sind Nichtterminalsymbole, rechteckige Knoten Terminalsymbole. Türkisfarbene Kanten zeigen auf diejenigen Symbole, durch die ein Nichtterminal bei Anwendung der entsprechenden Regel ersetzt wird. Graue Pfeile zeigen auf das Elternelement eines nichtterminalen Knotens und schwarze Kurven zeigen für Terminalen in welcher Richtung das jeweilige Elternelement liegt.

Dimension für die PCA mitaufzunehmen. Die Positionierung der Extremitäten bleibt also ein Problem mit unklaren Anforderungen und vielen Freiheitsgraden.

Ein erster Schritt an das Problem heranzugehen ist, es in kleinere Unterprobleme zu zer-teilen. Extremitäten können anhand ihrer Funktion in vier Kategorien eingeteilt werden: Flügel, Flossen, Extremitäten mit Bodenkontakt (im Folgenden als Beine bezeichnet), und Extremitäten ohne Bodenkontakt, die weder Flügel noch Flossen sind, (im Folgenden als Arme bezeichnet). Die Anzahl der Flossen und Arme wurde, im Gegensatz zu Beinen und Flügeln, nicht als Merkmal für die PCA erhoben. Im folgenden Absatz wird beschrieben wie stattdessen entschieden wird, ob eine Extremität eine Flosse oder ein Arm ist (wenn keine Benutzereingabe dazu vorhanden ist). Das ist aber relativ beliebig gewählt. Es wäre sicherlich auch eine gute Möglichkeit Arme und Flossen stattdessen als zusätzliche Merkmale mitaufzunehmen.

Für Flügel, Flossen und Arme gibt es keine besonderen Anforderungen außer, dass sie als solche zu erkennen sein sollten. Deshalb werden sie nach folgenden simplen Anweisungen orientiert (siehe auch Abbildung 5.2):

- Flossen: Ausrichtung gerade nach hinten (orientiert an Welt- x -Achse)
- Arme: Der Oberarm zeigt senkrecht nach unten (orientiert an Welt- y -Achse), im Ellenbogengelenk ist ein 90° Winkel und die Hand verlängert den Unterarm nach vorne.
- Flügel: Jedes beteiligte Gelenk hat ein Intervall mit festen Grenzen speziell für Flügel, aus dem zufällig ein Winkel gewählt wird.

Nun bleibt nur noch die Ausrichtung der Beine, für die die zusätzliche Anforderung gilt, dass sie den Boden berühren sollen.

Bestimmung von Art und Anzahl

Je nach Art wird die Anzahl der entsprechenden Extremitäten unterschiedlich bestimmt. Als Grundlagen dient der gewählte Punkt q im PCA-Raum. Hier werden die Werte v_b und v_f betrachtet, die q , zurücktransformiert ins ursprünglichen Koordinatensystem, für *Beine mit Bodenkontakt* und *Flügel* annimmt. Priorisiert werden die verschiedenen Arten von Extremitäten in der Reihenfolge, wie sie im Folgenden aufgezählt sind.

- Beine: Der Wert v_b liegt im Intervall $[0, 2]$ bzw. wenn $v_b < 0$ oder $v_b > 2$, so wird v_b auf 0 bzw. 2 festgelegt. Gilt $v_b \in [0, 1]$, so wird kein Vorderbein und mit Wahrscheinlichkeit v_b ein Hinterbein generiert. Gilt $v_b \in [1, 2]$, so wird mit Wahrscheinlichkeit 1 ein Hinterbein und mit Wahrscheinlichkeit $v_b - 1$ ein Vorderbein generiert.
- Flügel: Der Wert v_f liegt in $[0, 1]$ bzw. wird, wie bei den Beinen, darauf eingeschränkt. Wenn am Schultergürtel noch keine Extremität festgelegt ist, so wird dort mit Wahrscheinlichkeit v_f ein Flügel generiert.
- Arme: Die Berechnung basiert auf v_f und funktioniert gleich wie bei den Flügeln.
- Flossen: Wenn die Gesamtlänge der entsprechenden Extremität klein genug¹ ist und es Extremitätengürtel gibt, denen durch die obigen Regeln noch keine Extremität zugewiesen wurde, so wird an diesen Extremitätengürteln eine Flosse generiert.

Diese Berechnungen können dazu führen, dass Tiere, wie der Tyrannosaurus Rex, keine Arme bekommen, oder Fische ohne Flossen generiert werden. Das liegt daran, dass weder Arme noch Flossen Eingabemerkmale für die PCA sind. Deshalb verlässt sich der Algorithmus zusätzlich auf Benutzereingaben, die höher priorisiert werden, als die hier angegebenen Berechnungen.

¹Die maximale Länge wurde im Code auf 200px festgelegt. Das ist aber ein beliebig gewählter Wert.

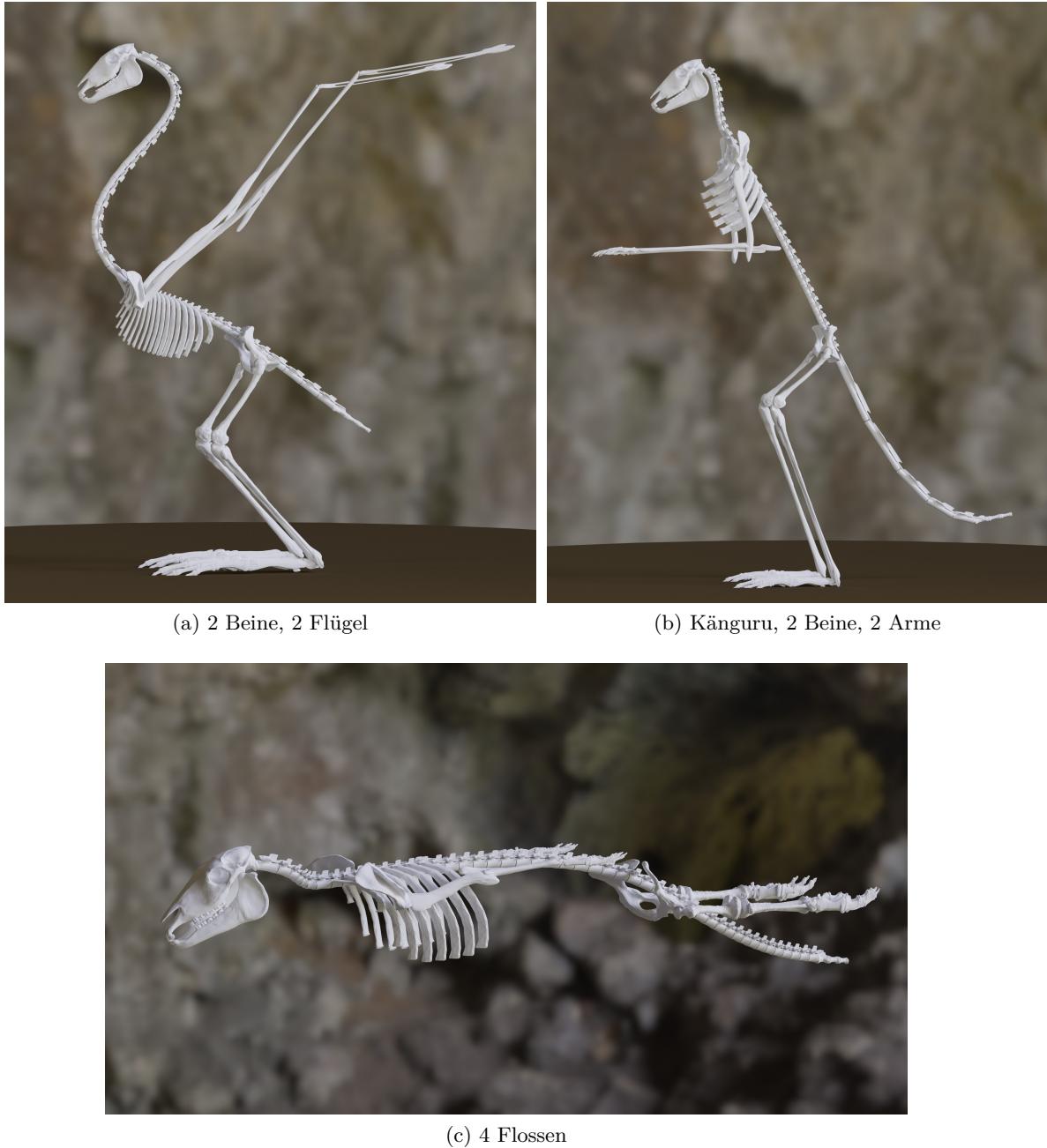


Abbildung 5.2.: Beispiele generierter Skelette mit verschiedenen Arten von Extremitäten.
 Die Bedingungen, die verwendet wurden, um die Skelette zu generieren, sind jeweils unter den Bildern zu finden. In (b) wurden die Daten verwendet, die für das Känguru erhoben wurden, mit den zusätzlichen Bedingungen, dass das Skelett 2 Beine und 2 Arme haben soll (siehe dazu auch Abschnitt 6.2). Als Hintergrund wurde [20] verwendet.

5.5.1. Berechnung der Bodenhöhe

Zunächst könnte man davon ausgehen, dass die Bodenhöhe einfach auf null festgelegt werden sollte. Das Problem hierbei ist aber, dass die von der PCA berechneten Längen für die Extremitäten meist so kurz sind, dass die Beine dann den Boden nicht mehr erreichen würden. Das liegt daran, dass auf den Bildern, die als Eingabebeispiele für die PCA verwendet wurden, der Boden meistens nicht ganz am unteren Rand ist. Hier würde rigoroses Abschneiden der Bilder auf Fußhöhe wahrscheinlich helfen, es würde in vielen Fällen aber auch ein Großteil der Füße verloren gehen.

Die Höhe des Bodens wird also nachträglich anhand der vorgegebenen Längen der Extremitäten festgelegt. Theoretisch würde es reichen einfach das kürzeste Bein im komplett ausgestreckten Zustand zu betrachten und den Boden auf die Höhe seines Endpunkts festzulegen. Das führt aber zu unnatürlich aussehenden Beinen, da das kürzeste Bein dann genau senkrecht nach unten führen muss um den Boden zu erreichen. Deshalb wird nur ein bestimmter Anteil² der Länge der Beine betrachtet. So wird erzwungen, dass die Beine, wenn sie auf dem neu definierten Boden stehen, auch etwas gekrümmmt sind.

Wie in Kapitel 2 beschrieben, gibt es verschiedene Arten von Füßen bzw. Händen. Die oben beschriebene Berechnung geht davon aus, dass der Boden mit der Fußspitze berührt wird. Aber natürlich gibt es auch Wirbeltiere, die mit dem flachen Fuß auf der Erde stehen. Deshalb wird zusätzlich zur Bodenhöhe noch eine Wahrscheinlichkeit berechnet, dass die Ferse den Boden berührt. Dies bestimmt im Algorithmus zur Ausrichtung der Beine (Abschnitt 5.5.2) ob die Fußspitze oder die Ferse auf den Boden bewegt werden soll.

Wird die Bodenhöhe nach oben verschoben, weil die Beine insgesamt zu kurz sind, ist diese Wahrscheinlichkeit Null. Sind die Beine so lang, dass schon ohne die Bodenhöhe anzupassen die Ferse bzw. das Handgelenk auf den Boden reicht, so ist sie eins. Ansonsten ist die Wahrscheinlichkeit

$$\frac{\text{Beinlänge} - \text{Höhe des Extremitätengürtels über 0}}{\text{Länge des Fußes}}$$

Wenn es Vorder- und Hinterbeine gibt, so wird diese Wahrscheinlichkeit für beide berechnet und dann der Mittelwert genommen. Es ist nämlich sinnvoll, dass alle Beine mit dem gleichen Punkt den Boden berühren.

5.5.2. Algorithmus zur Ausrichtung der Beine

Eine Herangehensweise das Problem der Ausrichtung der Beine zu lösen, wäre inverse Kinematik zu verwenden (siehe Abschnitt 3.3). Da das vorliegende Problem aber recht viele Randbedingungen hat, die ausgenutzt werden können, ist es gar nicht unbedingt nötig einen schwergewichtigen Algorithmus zu implementieren, der ein allgemeineres Problem löst.

Hier ist z. B. in den allermeisten Fällen klar in welche Richtung ein Gelenk gedreht werden muss, um den Fuß dem Boden zu nähern oder ihn vom Boden zu entfernen. Außerdem ist gar kein spezieller Punkt auf dem Boden vorgegeben, der erreicht werden soll. Deshalb wurde hier ein eigener, recht simpler Algorithmus entwickelt, der auf dieses spezielle Problem angepasst ist.

Die Gelenke in den Extremitäten werden hier so vereinfacht, dass sie genau einen Freiheitsgrad haben. Eltern- und Kindknochen sind also auf einer gemeinsamen Ebene festgelegt. Die verwendeten Gelenke mit ihren Einschränkungen sind in Abbildung 5.3 zu sehen. Der minimale und maximale Winkel für die Gelenke an Schulterblatt und Beckenknochen

²Dieser Anteil ist in der Implementierung mit 0,8 festgelegt, lässt sich aber natürlich auch variieren.

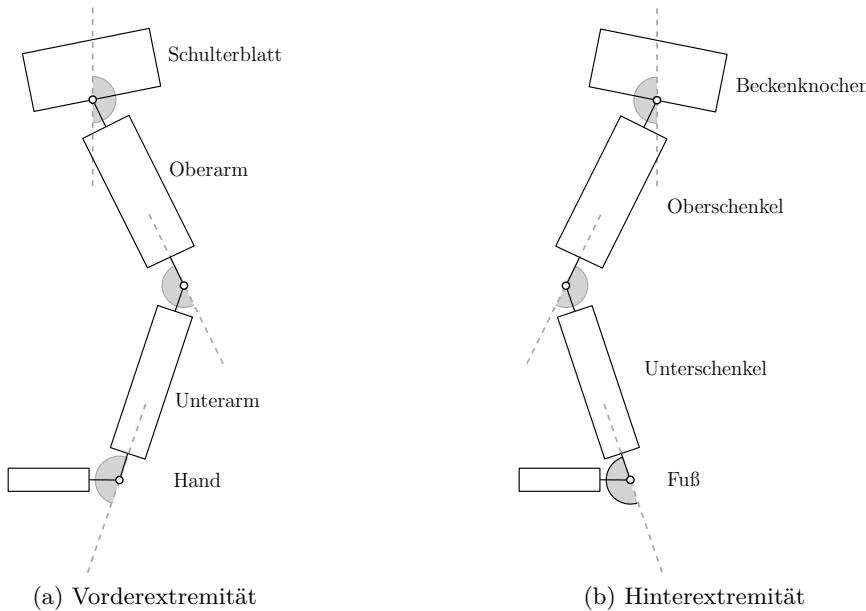


Abbildung 5.3.: Visualisierung der Bewegungsradien für die Gelenke in Extremitäten mit Bodenkontakt. Darstellung in Seitenansicht (wie in Abbildung 2.2). Die grauen Halbkreise zeigen die möglichen Winkel für eine Ruheposition des Skeletts. Unterarm, Unterschenkel und Fuß können jeweils maximal die Verlängerung des darüberliegenden Knochens bilden und minimal komplett an ihm anliegen. Oberarm und Oberschenkel können nicht über die Senkrechte hinaus gedreht werden.

orientieren sich an der Welt-*y*-Achse, da außerhalb dieses Bereichs keine sinnvollen Ruhepositionen entstehen würden.

Theoretisch haben das Hüft- und das Schultergelenk nicht nur einen, sondern zwei Freiheitsgrade. Sie lassen sich nicht nur nach vorne und hinten bewegen, sondern auch seitlich abspreizen. Das lässt sich auch leicht als zweite Art von Gelenk im Code abbilden. Allerdings liefert dies oft seltsam anmutende breitbeinige Tiere. Deshalb wurde der zweite Freiheitsgrad hier außen vor gelassen. Obwohl es natürlich in der Natur auch viele Tiere mit nach außen gestellten Beinen gibt, wie z. B. Echsen.

Ablauf

Der Algorithmus geht iterativ vor. In der Startposition ist die Extremität maximal angewinkelt. Die Gelenke beginnen also mit ihren kleinst- bzw. größtmöglichen Winkeln. In den folgenden Iterationen wird dann derjenige Endpunkt der Extremität dem Boden genähert, der zum Schluss Bodenkontakt haben soll. Je nach dem was während der Berechnung der Bodenhöhe (siehe Abschnitt 5.5.1) festgelegt wurde, soll entweder die Fußspitze oder die Ferse Bodenkontakt haben.

Eine andere Möglichkeit wäre mit einer komplett ausgestreckten Extremität zu beginnen und den Fuß von unten der Bodenhöhe anzunähern. Das würde wahrscheinlich genauso gut funktionieren.

In jedem Schritt werden nacheinander alle Gelenke einer Extremität betrachtet, beginnend mit dem obersten in der Knochenhierarchie. Dabei wird berechnet, ob der Winkel des entsprechenden Gelenks vergrößter oder verkleinert werden muss, um den dazugehörigen Kindknochen näher zum Boden zu bewegen. Dann wird das Gelenk um einen vorher bestimmten Winkel α in die berechnete Richtung gedreht.

Die Drehrichtung lässt sich relativ leicht herausfinden, indem die Ausrichtung des Knochens mit der Welt-*y*-Achse verglichen wird. Je senkrechter der Knochen ausgerichtet ist,

desto ausgestreckter ist das Bein. Es gibt also globale Randbedingungen und lokale Einschränkungen je nach Gelenk.

Ohne weitere Einschränkungen kann es nun passieren, dass unnatürliche Positionen auftreten, in denen sich z. B. der Fußspann näher am Boden befindet als die Fußsohle. Oder es kann passieren, dass ein Knochen über die positive Welt- y -Achse hinaus gedreht wird. Das Problem dabei ist, dass die Einschränkungen an den Gelenken nicht zulassen, dass der Knochen sich unbegrenzt in diese Richtung weiterdreht und der Knochen dann „feststeckt“. Deshalb wird nach jeder Drehung festgestellt ob solch eine Situation eingetreten ist und wenn ja, wird die Drehung rückgängig gemacht. Die Drehung wird ebenfalls rückgängig gemacht, falls sie bewirkt, dass ein Knochen unterhalb der Bodenhöhe liegt. So ist zu jeder Zeit durch Invarianten garantiert, dass die Knochen auf der „richtigen“ Seite der y -Achse und oberhalb der Bodenhöhe liegen.

Der Winkel α , um den die Gelenke gedreht werden, wird in jeder Iteration um einen bestimmten Anteil verkleinert. In einem Schritt werden aber alle Gelenke um den gleichen Winkel gedreht.

Zu Beginn soll mit großen Veränderungen eine grobe Ausrichtung der Gelenke vorgenommen werden, die dann immer weiter verfeinert wird. Der Startwinkel darf nicht zu klein sein, weil die Gelenke sonst ihre Zielpositionen nicht erreichen können. Ist der Startwinkel allerdings zu groß, bewirkt das in vielen Fällen, dass in den ersten Schritten des Algorithmus keine Drehung durchgeführt werden kann, weil die oben genannten Randbedingungen durch den großen Winkel verletzt werden.

Die Verkleinerung des Winkels darf nicht zu schnell geschehen, weil dann auch die Endposition nicht erreicht werden kann. Wenn sie aber zu langsam geschieht, passiert in vielen Schritten wiederum nichts wegen verletzter Randbedingungen.

Durch Ausprobieren wurden folgende Zahlen als sinnvoll erachtet: Startwinkel $\alpha = 40^\circ$, später dann jeweils $\frac{6}{7}$ davon.

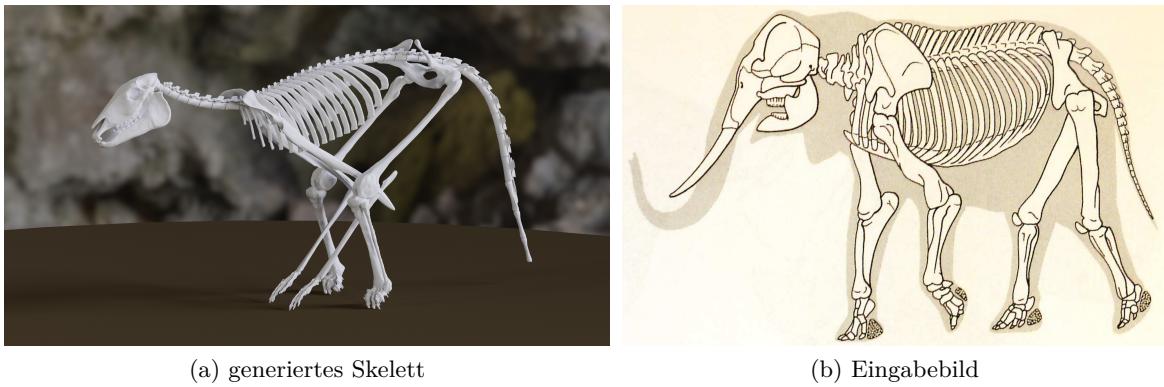
Falls sich der Abstand zum Boden kaum verändert, liegt die Vermutung nahe, dass die Gradzahl zu groß ist und deshalb alle möglichen Winkeländerungen invalide sind. Deshalb wird in diesem Fall die Gradzahl für die nächste Iteration stärker verkleinert (halbiert).

Hat das Tier sehr kurze Beine, hat der Algorithmus außerdem einige kleine Probleme. Diese werden genauer in Abschnitt 7.5 beschrieben. Da die Beine aber in diesen Fällen, wie gesagt, sehr kurz sind, ist es für den Gesamteindruck gar nicht besonders wichtig wie genau sie angeordnet sind.

Vergleich mit echten Beinstellungen

Wie in Kapitel 6 beschrieben, lassen sich auch die Eingabebeispiele der PCA laden. Bei ihnen sind dann alle Attribute, die die PCA liefert, schon festgelegt. Alle anderen müssen jedoch noch generiert werden. Dazu gehören auch die Beine. Vergleicht man nun die Beinstellung, die der oben beschriebene Algorithmus generiert, mit der Beinstellung auf dem Eingabebild, lassen sich teilweise sehr große Unterschiede feststellen. In Abbildung 5.2 b wurde ein Känguru generiert. Hier ist die Beinstellung relativ realistisch. Beim Elefanten in Abbildung 5.4 hingegen weicht die Beinstellung stark vom Eingabebild ab.

Um in allen Fällen eine realistisch wirkende Positionierung der Beine zu bekommen, müsste noch sehr viel mehr Arbeit in den Algorithmus gesteckt werden. Außerdem bräuchte der Algorithmus mehr Informationen zum Tier. Solche Zusatzinformationen könnten beispielsweise die Art des Fußes oder die Fortbewegungsart sein. Auch könnte es helfen, wenn es eine sinnvolle Möglichkeit gäbe, die Winkel an den Gelenken als Dimension für die PCA mitaufzunehmen. Dafür müsste man sich aber, wie zu Beginn des Kapitels schon erwähnt, auf eine kanonische Ruheposition einigen und dann auch noch Beispiele in genau dieser Position finden.



(a) generiertes Skelett

(b) Eingabebild

Abbildung 5.4.: (a) Skelett eines Elefanten, das anhand der erhobenen Daten generiert wurde. Als Hintergrund wurde [20] verwendet. (b) Die Abbildung des Skeletts eines Elefanten, die auch als Eingabe für die PCA verwendet wurde.

Ein fertig generiertes Skelett wird höchstwahrscheinlich auch noch weiterverarbeitet. Soll z. B. ein animiertes Tier daraus werden, so müssen Bewegungszyklen geschaffen werden. Dafür muss jedes Gelenk vielfach bewegt werden. Soll ein Tier mit Haut und Muskeln daraus werden, so müssen Muskeln an den Knochen ansetzen, die dann einen nicht unerheblichen Anteil an der Positionierung der Beine haben.

Die von dem hier beschriebenen Algorithmus generierte Position kann also gut als erster Eindruck dienen, muss aber in den meisten Fällen noch angepasst werden. Ausgehend von der gegebenen Datenlagen und von den zu erwartenden Anwendungen ist es aber nicht sinnvoll den Algorithmus weiter zu verfeinern.

5.5.3. Zusätzliche Ansatzpunkte für Extremitäten

Ansatzpunkte für Extremitäten sind zunächst der Hüftgürtel und der Schultergürtel. Um auch die Generierung fantastischer Tiere (siehe Abschnitt 2.3) zu ermöglichen, ist es aber möglich dies zu erweitern. Um die Anzahl der zusätzlichen Extremitäten zu bestimmen, ist es am einfachsten sich auf Benutzereingaben zu verlassen. Die Beispieldaten für die PCA sind nur echte Wirbeltierskelette. Deshalb lassen sich aus Punkten im PCA-Raum keine Informationen zu zusätzlichen Extremitäten ableiten.

Zwei Extremitätenpaare pro Extremitätengürtel

Eine einfache Möglichkeit ist zunächst die Anzahl der möglichen Extremitätenpaare von zwei auf vier zu erhöhen, indem einfach an der Hüfte und der Schulter jeweils zwei Paare ansetzen dürfen. Dafür wurden an der Hüfte bzw. der Schulter mehrere Gelenke direkt hintereinander angelegt. Das ermöglicht beispielsweise Tiere wie den Pegasus (siehe Abbildung 5.5).

Um das zu ermöglichen, wird die Grammatik $G = (\Sigma, N, S, P, p)$ aus Abschnitt 5.4 zu einer Grammatik $G' = (\Sigma', N', S, P', p')$ erweitert, mit $\Sigma' = \Sigma \cup \{\text{Beckenknochen2}\}$ und $N' = N \cup \{\text{Schulterpartie2}\}$. Das Terminal „Beckenknochen2“ steht hier für einen Beckenknochen mit zwei Gelenken für Extremitäten.

Die Menge der Parameter p wird zu $p' = p \cup \{e_{s2}, e_{b2}\}$ erweitert. Die neuen Parameter können jeweils Werte aus $\{0, 1\}$ annehmen und stehen dafür, ob eine zweite Vorder- bzw. Hinterextremität generiert werden soll. Es gilt $e_s = 0 \Rightarrow e_{s2} = 0$ und $e_b = 0 \Rightarrow e_{b2} = 0$. Gilt $e_{s2} = 0$ und $e_{b2} = 0$, so verhält sich G' wie G .

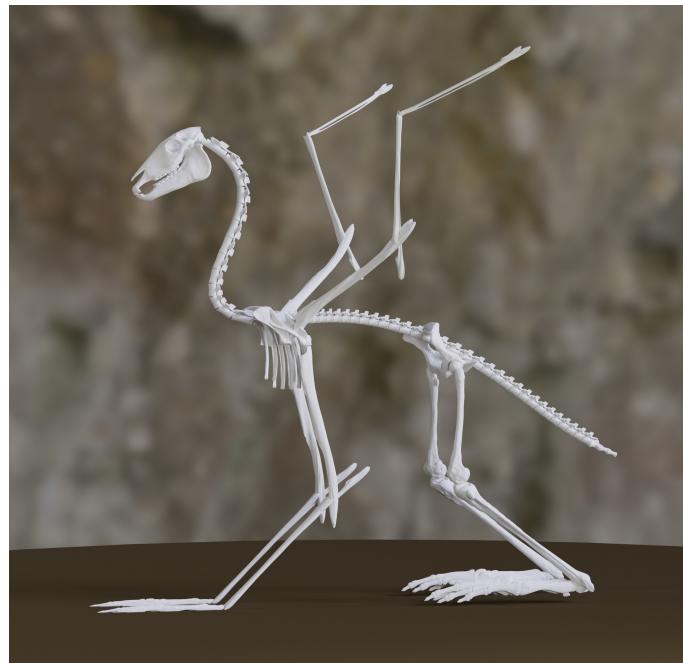


Abbildung 5.5.: Beispiel für ein Skelett, dass mit folgenden Bedingungen generiert wurde:
4 Beine, 2 Flügel. Mehrere Extremitätenpaare pro Extremitätengürtel waren erlaubt.
Als Hintergrund wurde [20] verwendet.

Die Produktionen in P für *Vorderteil* und *Beckengürtel* werden in P' durch folgende Produktionen ersetzt. Außerdem wird eine Produktion für *Schulterpartie2* eingefügt.

$$\begin{aligned}
 \textit{Vorderteil} &\rightarrow \text{Wirbel}^{w_{rv} - (w_{rv}^r - 3)^+ - 3} \\
 &\quad [\text{Wirbel (Rippe)}]^{(w_{rv}^r - 3)^+} \\
 &\quad \text{Wirbel (Schulterpartie2)} \\
 &\quad \text{Wirbel (Rippe}^{(\min(1, w_{rv}^r - 1))^+} \\
 &\quad \text{Wirbel (Schulterpartie)} \\
 &\quad \textit{Hals} \\
 \textit{Schulterpartie2} &\rightarrow \text{Rippe}^{\min(1, w_{rv}^r - 2)} \text{ Schultergürtel}^{e_{s2}} \\
 \textit{Beckengürtel} &\rightarrow [\text{Beckenknochen } \textit{Hinterextremität}]^{e_b - e_{b2}} \\
 &\quad [\text{Beckenknochen2 (} \textit{Hinterextremität} \text{) } \textit{Hinterextremität}]^{e_{b2}}
 \end{aligned}$$

Die zweite Schulterpartie beginnt am zweiten Wirbel hinter der ersten Schulterpartie und erzeugt ebenfalls einen Schultergürtel, falls zwei Extremitäten an der Schulter des Tieres ansetzen sollen. Am Beckengürtel wird anhand der Anzahl der Extremitäten entschieden ob der einfache Beckenknochen oder „Beckenknochen2“ mit zwei Gelenken erzeugt werden soll.

Flügel und Arme dürfen hierbei nur an der Schulter ansetzen, Beine und Flossen an beiden Stellen. Der Grund dafür ist, dass die meisten generierten Skelette seltsam wirken, wenn an der Hüfte Flügel oder Arme ansetzen und dafür an der Schulter Beine beginnen. Das liegt daran, dass existierende Tiere mit Flügeln oder Armen ihren Schwerpunkt im hinteren Bereich haben und sie auf den Hinterbeinen stehen. Deshalb wird die Wirbelsäule durch die PCA auch dementsprechend angelegt.

Mehrere Extremitätengürtel entlang der Rückenwirbelsäule

Eine Überlegung könnte auch sein, zwischen Schulter und Hüfte weitere Extremitätengürtel zu erlauben, um z. B. Tiere wie den asiatischen Drachen erzeugen zu können.

Das stellt sich aber als schwierig heraus. Die Wirbelsäule ist zwischen Hüfte und Schulter meist nach oben geschwungen und im Bauchraum befinden sich viele Organe. Ein zusätzlicher Extremitätengürtel würde den Bauchraum einschränken. Außerdem wirkt dann auch die nach oben geschwungene Wirbelsäule anatomisch seltsam. Verdoppelt man die Schwingung der Wirbelsäule und hängt einfach einen weiteren Rücken hinten oder vorne an, so wirkt es ebenso seltsam, da dann die „Höcker“ der Wirbelsäule für das Tier wahrscheinlich nicht von Vorteil sind und nur die Fortbewegung erschweren.

Asiatische Drachen sind Spezialfälle, bei denen die zusätzlichen Hüften nicht seltsam wirken, da sie einen langen, schlangenartigen Körper besitzen, bei dem die Wirbelsäule gerade verläuft.

Zweiter Schultergürtel

Eine weitere Möglichkeit ist Tiere, ähnlich zu Zentauren, zu ermöglichen. Zentauren haben zwei Schultergürtel. Hat das Tier, das generiert wird, einen Hals, der lang genug ist, kann darauf ein weiterer Schultergürtel kurz unterhalb vom Kopf angebracht werden. An diesem Schultergürtel dürfen dann alle Arten von Extremitäten außer Beinen ansetzen. Das wirkt tatsächlich meist auch anatomisch einigermaßen sinnvoll (siehe Abbildung 5.6).

Hierfür wird die Grammatik $G' = (\Sigma', N', S, P', p')$ aus dem obigen Abschnitt zu einer Grammatik $G'' = (\Sigma', N', S, P'', p'')$ mit $p'' = p' \cup \{e_h, e_{h2}\}$ erweitert. Die Parameter $e_h \in \{0, 1\}$ und $e_{h2} \in \{0, 1\}$ stehen jeweils dafür, ob eine erste bzw. eine zweite Vorderextremitäten am zweiten Schultergürtel am Hals generiert werden soll. Es gilt $e_h = 0 \Rightarrow e_{h2} = 0$. Hier gilt wiederum: ist $e_h = 0$ und $e_{h2} = 0$, so verhält sich G'' gleich wie G' .

Die Produktion für den *Hals* in P' bzw. P wird in P'' durch folgende Produktion ersetzt.

$$\begin{aligned}
 \text{Hals} &\rightarrow \text{Wirbel}^{w_h - 5} \\
 &\quad \text{Wirbel (} \text{Schultergürtel}^{e_{h2}} \text{)} \\
 &\quad \text{Wirbel} \\
 &\quad \text{Wirbel (} \text{Schultergürtel}^{e_h} \text{)} \\
 &\quad \text{Wirbel}^2 \\
 &\quad \text{Schädel}
 \end{aligned}$$

Wie an der Schulter, können auf dem Hals, mit dem Abstand von einem Wirbel, zwei Schultergürtel erzeugt werden. Die Gesamtzahl der Wirbel bleibt dabei gleich. Dies funktioniert auch problemlos, da der Hals, falls vorhanden, immer ≥ 7 Wirbel besitzt.

Ein solcher Schultergürtel wirkt aber nur sinnvoll, wenn der Hals lang genug ist. Deshalb wird die Benutzereingabe, die einen zweiten Schultergürtel erzwingt, in eine Bedingung für die Halslänge umgewandelt und an die PCA weitergegeben (siehe Abschnitt 6.1). Gibt es keine Benutzereingabe, so wird ein zweiter Schultergürtel nur generiert, wenn der Hals lang genug ist.

5.6. Knochenmodelle

Um ein 3D-Modell des gesamten Skeletts zu erstellen, wird zunächst jeder terminale Knochen als Quader dargestellt. Diese Quader ergeben sich aus Position, Orientierung und Abmessungen des zugehörigen Knochens. Alle Quader zusammen ergeben ein rudimentäres Modell. Jedoch lassen sie sich auch relativ leicht durch feinere 3D-Modelle der entsprechenden Knochen ersetzen.

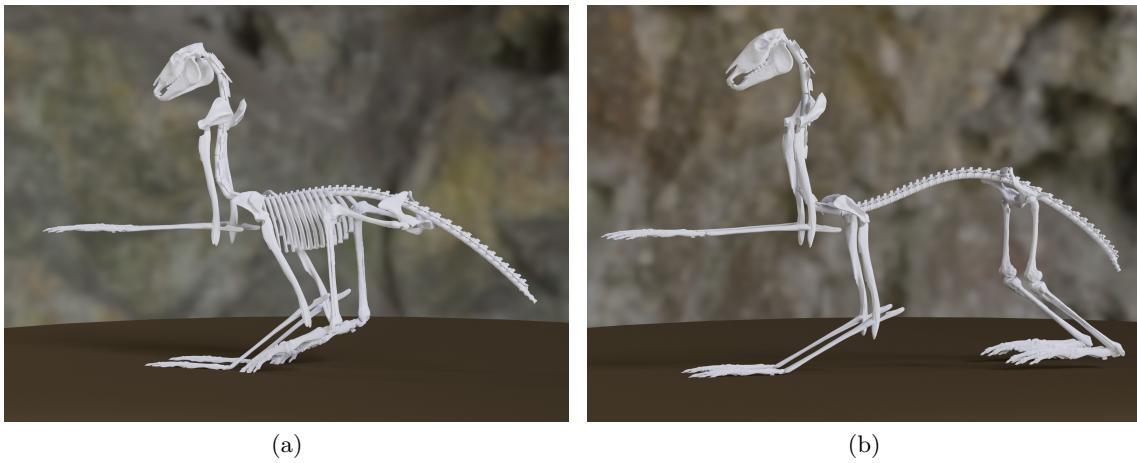


Abbildung 5.6.: Zwei Beispiele für Skelette, die mit folgenden Bedingungen generiert wurden: 4 Beine, 2 Arme und ein zusätzlicher Schultergürtel. Mehrere Extremitäten pro Extremitätengürtel sind nicht erlaubt. Als Hintergrund wurde [20] verwendet.

Vorverarbeitung der Knochenmodelle

Die Knochenmodelle müssen im .obj-Format vorliegen, da auch das resultierende Modell dieses Format hat. Außerdem muss jedes Modell an den Koordinatenachsen ausgerichtet und so verzerrt sein, dass es einen Würfel mit Kantenlänge 1 in jeder Richtung möglichst gut ausfüllt. So ist es sehr einfach die Längen der Knochen auf das entsprechende 3D-Modell zu übertragen.

Bei dieser Vorgehensweise treten jedoch einige Schwierigkeiten auf. Es ist z.B. relativ schwierig herauszufinden, wie man die einzelnen Knochen skalieren muss, sodass sie an den Gelenken gut zusammenpassen und nicht verzerrt aussehen. Im Folgenden sind einige Anpassungen aufgezählt, die die Positionierung vereinfachen.

- Kleine Fortsätze, die nicht wirklich zur (optischen) Größe des Knochens beitragen, z. B. die Fortsätze der Wirbel, ragen aus dem Würfel heraus. Der Knochen ist somit an manchen Stellen etwas länger als vorgegeben.
- Teile eines Knochens, bei denen es wichtig ist, dass sie eine bestimmte Länge haben, sind genau auf die Kantenlänge des Würfels abgestimmt. So können sie leicht auf die richtige Länge skaliert werden. Ein Beispiel dafür ist die Länge der Wirbel entlang der Wirbelsäule. Die Wirbel müssen genau aneinander ansetzen. Mit dieser Skalierung lässt sich einfach die Länge des Wirbels auf das Modell übertragen.
Es kann aber auch um Längen gehen, die nur einen Teil des Knochens betreffen. Rippen müssen z.B. mit ihrer Breite in x -Richtung zur Breite des Wirbels passen, an welchem sie ansetzen. Deshalb ist das 3D-Modell der Rippe so skaliert, dass die Kantenlänge des Würfels in x -Richtung genau auf die Breite des Wirbels skaliert werden kann und die resultierende Breite der Rippe dann genau dazu passt.
Da die Rippe in x - und z -Richtung stark gebogen ist, führt das dazu, dass die Gesamtlänge der Rippe in x -Richtung viel länger ist als die Kantenlänge des Würfels. Das Problem daran ist, dass die gegebenen Abmessungen des Knochens dann nicht mehr viel mit der Ausdehnung des eingesetzten 3D-Modells haben. Je nach dem um welchen Knochen es sich handelt, möchte man diesen Effekt evtl. vermeiden.
- Kantenlängen, die nicht speziell vorgegeben werden, z. B. die Dicke der Extremitätenknochen, sind einfacher passend zu bestimmen, wenn sie nicht komplett unabhängig von den anderen Raumrichtungen sind. Ist z.B. die x - und y -Skalierung eines

Knochens vorgegeben, und die Skalierung in z -Richtung soll nur möglichst gut dazu passen, so ist es sinnvoll das 3D-Modell schon so zu speichern, dass die z -Richtung relativ zu einer anderen Richtungen angegeben werden kann. Tut man dies nicht, so führt das leicht dazu, dass die Knochen verzerrt aussehen.

Offsets zwischen Knochen

Es ist eventuell nicht ganz leicht zu erkennen, wie die Knochen ineinander greifen bzw. wie die Gelenke die Ausrichtung der Knochen beeinflussen. Das erfordert etwas Wissen zur Anatomie und viel „Finetuning“.

Für jeden Knochen sind für diese Ausrichtung zwei Offsets gespeichert: das Offset zu dem Gelenk, das ihn mit seinem Elternknochen verbindet und das Offset zu dem Gelenk, das ihn mit seinem Kindknochen verbindet (oder mehrere, falls vorhanden). Dies sorgt dafür, dass die Positionierung der Knochen stimmt, egal wie groß sie sind. Ist ein Knochen sehr groß und ein anschließender sehr klein (oder anders herum), so kommt es natürlich trotzdem vor, dass die Gelenke nicht wirklich ineinander passen. Für solche Situationen bräuchte man verschiedene 3D-Modelle, die je nach Gegebenheit eingesetzt werden.

Es wurden vor allem Modelle von Pferde- und Menschenknochen verwendet, da sie leicht verfügbar waren. Manche Knochen sind jedoch auch von anderen Tieren oder keinem speziellen Tier zugeordnet. Das führt z.B. bei dem verwendeten Unterarmknochen vom Pferd dazu, dass er etwas überdimensionierte Fortsätze am Ellenbogen bekommen, wenn man ihn stark verlängert. Das liegt daran, dass dieser Knochen beim Pferd eigentlich relativ kurz ist. Ansonsten passen die Knochen verschiedener Wirbeltiere aber erstaunlich gut ineinander.

Leider unterscheiden sich die Knochen verschiedener Tiere dennoch so stark, dass die Offsets für jedes Knochenmodell individuell bestimmt werden müssen. Dies, und die oben genannten Abweichungen von einem Würfel mit Kantenlänge 1, führt dazu, dass die 3D-Modelle nicht einfach austauschbar sind. Um andere 3D-Modelle für Knochen zu verwenden, müsste nicht nur das Modell ausgetauscht werden, sondern eben auch die Offsets angepasst werden. Das ist momentan im Programm nicht möglich ohne den Code zu ändern. Dies ließe sich aber leicht hinzufügen. Die Offsets könnten z.B. in einer Textdatei gespeichert werden und zusammen mit den Modellen eingelesen werden.

Spezielle Knochen

Für die meisten Knochen wird genau ein 3D-Modell verwendet. Bei manchen Knochen und in manchen Fällen wirkt das aber recht seltsam. Man möchte eigentlich etwas mehr Variation haben.

Besonders stark ist dies am *Schädelknochen* zu beobachten. Er variiert, im Gegensatz zu anderen Knochen, bei Wirbeltieren sehr stark. Hier ist es also ein großer Mehrwert verschiedene Modelle anzubieten. Die Frage ist dann nur wie entschieden wird welcher Schädelknochen wann verwendet wird. Eine Möglichkeit wäre in die Grammatik spezifischere Terminalsymbole für verschiedene Schädel und dazugehörige Regeln einzufügen. Tut man dies nicht, gibt es wenig Anhaltspunkte. Es ist deshalb sinnvoll die Auswahl des passenden Schädelknochens dem Nutzer zu überlassen.

Bezüglich der Offsets ist der Schädelknochen ein recht einfacher Fall. Die verschiedenen Modelle lassen sich leicht alle gleich skalieren, sodass keine verschiedenen Offsets gespeichert werden müssen.

Bei *Händen und Füßen* ist das Problem ebenso, dass es sehr viele verschiedene Ausprägungen davon gibt (siehe z.B. Abbildung 2.1). Diese lassen sich jedoch gut nach Extremitätentyp unterscheiden, wobei diese Unterscheidung natürlich beliebig fein sein kann. Hier

wurde die Unterscheidung verwendet, die auch bei der Positionierung der Extremitäten zum Einsatz kommt (siehe Abschnitt 5.5).

Es gibt also Flügel, Arme, Flossen und Beine mit Bodenkontakt. Wobei bei Beinen mit Bodenkontakt zusätzlich noch nach dem Winkel unterschieden wird mit dem der Fuß auf den Boden aufkommt. Bei weniger als 45° wird eine menschliche Hand eingesetzt, sonst ein Pferdehuf. Arme bekommen ebenfalls eine Hand. Für Flossen wurde kein spezielles Modell eingefügt, bei ihnen wird ebenfalls eine Hand eingesetzt. Flügel haben ein eigenes Modell. In Ermangelung gemeinfreier Modelle fehlen hier aber leider einzelne Bestandteile.

Es wurde ein Modell für alle *Wirbel* verwendet. Bei Wirbeltieren gibt es aber sehr viele unterschiedliche Wirbel, die auch in ihrer Größe stark variieren. Ein Beispiel ist das Ende der Schwanzwirbelsäule. Hier werden die Wirbel bei echten Tieren kleiner. Um das Gesamtmodell realistischer zu machen, lassen sich die Modelle der Wirbel verfeinern. Als Beispiel wurde ein zusätzliches Modell eingeführt, dass die letzten drei Schwanzwirbel ersetzt. Es besteht aus drei aufeinanderfolgenden Wirbeln, die zum Ende hin kleiner werden. Das sieht in den meisten Fällen auch gut aus. Ein Nachteil davon, ein einzelnes Modell mit drei Wirbeln zu haben, ist aber, dass die Positionierung dann nicht mehr so fein erfolgen kann. Anfangs- und Endpunkt des Modells werden auf der Bézierkurve positioniert. Aber die Anfangs- und Endpunkte der einzelnen Wirbel können teilweise stark vom Verlauf der Kurve abweichen. Dies lässt sich natürlich leicht durch eine weitere Verfeinerung der Modelle beheben, indem jeder der drei Wirbel ein eigenes Modell bekommt.

Setzten *mehrere Extremitäten* an einer Hüfte an, so ist ein realistisches Modell des Beckenknochens nicht mehr ausreichend, da nicht genug Gelenke vorhanden sind. Dieses Problem wurde so behoben, dass für das Terminal „Beckenknochen2“ ein kombiniertes 3D-Modell aus zwei einzelnen Beckenknochen erstellt wurde, an dem nun zwei Gelenke vorhanden sind.

6. Zusätzliche Funktionen

In Kapitel 5 wird beschrieben wie der Algorithmus abläuft. In diesem Kapitel geht es nun darum die Verwendungsmöglichkeiten des Algorithmus auszuschöpfen. Über eine Benutzeroberfläche wird die Bedienung vereinfacht und gezeigt welche Möglichkeiten es gibt die Skelettgenerierung zu steuern. Außerdem ist es möglich die Metadaten eines Skeletts zu speichern und Variationen zu einem bestehenden Skelett zu generieren.

6.1. Benutzeroberfläche

Mit dem in Kapitel 5 vorgestellten Algorithmus können nicht nur zufällige Skelette generiert werden. Über eine Benutzeroberfläche ist es auch möglich, dem Algorithmus zusätzliche Eingaben zu geben. Die Benutzeroberfläche ist in Abbildung 6.1 zu sehen.

Diese Eingaben werden unter anderem in Bedingungen umgewandelt, die schon von der PCA berücksichtigt werden. Wie eine bedingte Verteilung als Eingabe für die PCA berechnet wird, wird in Abschnitt 4.5 erklärt. Wie dort auch beschrieben, werden bei allen Bedingungen zufällige kleine Werte aufaddiert oder abgezogen um mehr Variation auf den erzeugten Skeletten zu bekommen. Dies ist sinnvoll, da der Benutzer so nicht gezwungen wird so etwas wie 1,4 Beinpaare zu verlangen, um z. B. die Form der Wirbelsäule zu verändern. Bei solch einer Eingabe müsste dann außerdem durch den Algorithmus ausgelost werden, ob das Skelett am Ende 1 oder 2 Beinpaare hat. Das wäre unter Umständen verwirrend.

Ist in Zahlenfeldern nichts eingegeben, haben diese keinen Einfluss auf den Ablauf (im Gegensatz zur Eingabe der Zahl 0).

Die folgenden Eingaben werden durch die PCA berücksichtigt:

- Die Anzahl der Beine mit Bodenkontakt (*floored legs*) muss eine gerade Zahl n zwischen 0 und 8 sein. Als Bedingung für das Merkmal *Paare von Beinen mit Bodenkontakt* wird dann $\frac{n}{2}$ verwendet. Bei echten Wirbeltieren liegt die Anzahl der Beine zwischen 0 und 4. Werden Werte größer 4 als Bedingung an die PCA gestellt, so werden sehr stark geschwungene Wirbelsäulen generiert. Diese sehen aber, bis zu einem Wert von ca. 8 gar nicht schlecht aus (siehe Abbildung 6.2 a). Deshalb werden diese Werte auch als Bedingungen erlaubt. Sie werden nur mit einer gewissen Wahrscheinlichkeit etwas verkleinert, weil es auch möglich ist fantastische Tiere mit mehr als 4 Beinen mit Wirbelsäulen für 4-beinige Tiere zu generieren.

load from file	<input type="file"/> choose
construct from example	<input type="text"/> Archaeopteryx
create variations	<input type="checkbox"/>
floored legs	<input type="checkbox"/>
wings	<input type="checkbox"/>
arms	<input type="checkbox"/>
fins	<input type="checkbox"/>
neck y length	<input type="checkbox"/>
tail x length	<input type="checkbox"/>
2 extremities per girdle allowed?	<input type="checkbox"/>
second shoulder on neck?	<input type="text"/> allowed
head kind	<input type="text"/> horse_skull
number of skeletons to generate	<input type="text"/> 1
skeleton name	<input type="text"/> skeleton
resolution	<input type="text"/> low
save metadata to file	<input type="checkbox"/> <input type="text"/>
show skeleton	<input type="button" value="choose skeleton"/>
<input type="button" value="start generator"/>	

Abbildung 6.1.: Die Benutzeroberfläche des Programms. Der Knopf „start generator“ startet den Algorithmus mit den zusätzlichen Eingaben aus den Feldern darüber.

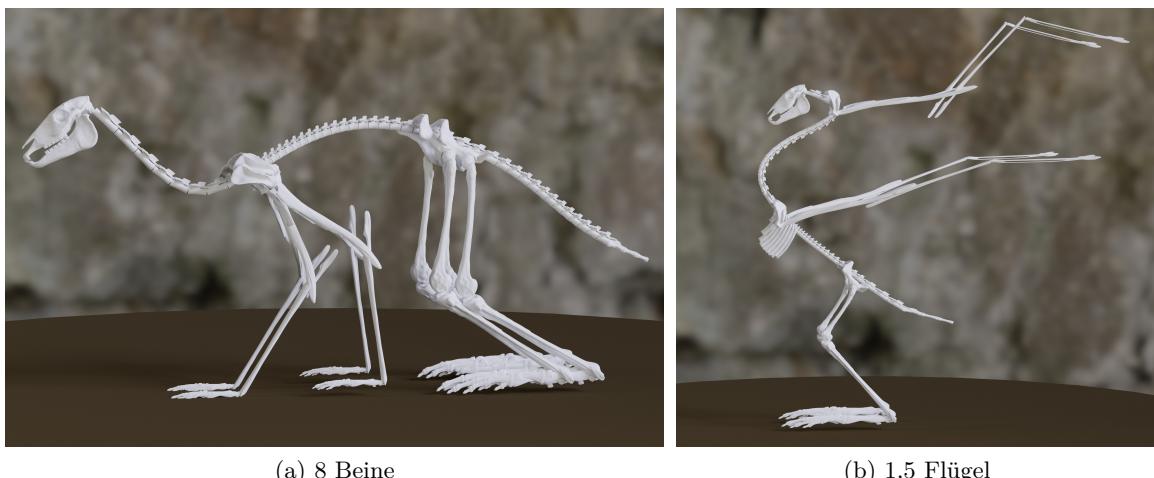


Abbildung 6.2.: Zwei Skelette, die mit extremen Bedingungen für die PCA generiert wurden. (a) Benutzereingabe: 8 Beine und 4 Beinpaare als Bedingung für *Beine mit Bodenkontakt*. Es ist kein zweiter Schultergürtel am Hals erlaubt, aber zwei Extremitätenpaare pro Extremitätengürtel. (b) Benutzereingabe: 4 Flügel und 1,5 Flügelpaare als Bedingung für die PCA. Ein zweiter Schultergürtel am Hals wird erzwungen, aber es ist nur ein Extremitätenpaar pro Extremitätengürtel erlaubt.

- Die Anzahl der Flügel (wings) muss ebenfalls gerade sein. Die Hälfte des Eingabewerts, aber maximal 1, wird zu einer Bedingung für das Merkmal *Flügel*. Die Wirbelsäulen, die unter Bedingungen für *Flügel* größer 1 generiert werden, sehen mit großer Wahrscheinlichkeit unrealistisch aus. Deshalb werden größere Werte nicht erlaubt. In Abbildung 6.2 b ist ein Skelett zu sehen, dass einen Wert von 1,5 für *Flügel* hat. Dieser Wert kann tatsächlich noch erreicht werden, da ein zufälliger Wert aus $[-0,5, 0,5]$ auf die Bedingung aufaddiert wird. Bei noch größeren Werten ist der Hals noch extremer geschwungen.
- Die Zahlen für Arme (arms) und Flossen (fins) gehen nicht als Bedingung in die PCA ein (siehe Abschnitt 5.5).
- Die y -Komponente des Abstands zwischen Kopf und Schultergürtel (neck y length) und die x -Komponente des Abstands zwischen Hüfte und Schwanzspitze (tail x length) können wiederum direkt als Bedingung an die PCA weitergegeben werden.

Die Anzahlen der verschiedenen Extremitätentypen werden außerdem verwendet, um die Parameter e_v , e_h und e_{v2} für die Anzahl der Vorder- und Hinterextremitäten der Grammatik (siehe Abschnitte 5.4 und 5.5.3) zu bestimmen und um den Typ und damit die Positionierung der verschiedenen generierten Extremitäten festzulegen (siehe Anfang Abschnitt 5.5). Zusätzlich muss mit eingerechnet werden, ob zwei Extremitätenpaare pro Extremitätengürtel erlaubt sind (2 extremities per girdle allowed?) und ob ein zweiter Schultergürtel auf dem Hals erlaubt, erzwungen oder verboten ist (second shoulder on neck?) (siehe auch Abschnitt 5.5.3).

Außerdem gibt es theoretisch noch die Möglichkeit anzugeben, welches 3D-Modell für den Kopf verwendet werden soll (head kind). Leider gibt es hier, Stand Abschluss der Arbeit, nur eines zu Auswahl. Dies ließe sich aber leicht anpassen (siehe Abschnitt 5.6). Zusätzlich kann eingestellt werden, welche Auflösung das 3D-Modell haben soll (resolution). Hier gibt es die Möglichkeit nur Quader zu generieren oder Modelle von echten Knochen in niedriger oder hoher Auflösung einzusetzen.

Im Feld „skeleton name“ kann der Dateiname des zu generierenden Modells eingestellt werden und in „number of skeletons to generate“ kann festgelegt werden, wieviele Skelette auf einmal generiert werden sollen. Und schließlich kann man sich auch ein generiertes Modell anzeigen lassen (show skeleton). Dazu wird das Programm *JavaView* [27] verwendet. Es ist darauf spezialisiert 3D-Geometrie interaktiv zu visualisieren und kann in Javacode eingebunden werden.

6.2. Speichern und Laden von Skeletten

Um ein vorgegebenes Skelett reproduzieren zu können, reicht es einige Metadaten zu speichern. Dazu gehören die Dimensionen der PCA (Position der Wirbelsäule, Länge der Knochen in den Extremitäten und Gewicht) und Daten, die zusätzlich generiert werden (Anzahl und Art der Extremitäten an den jeweiligen Extremitätengürteln, die Winkel an den Gelenken der Extremitäten, Anzahl der Wirbel und Rippen und welches 3D-Modell für den Kopf verwendet werden soll). Das alles ist in wenigen Java-Klassen gebündelt. Deshalb können die Daten über Java-Serialisierung [26] leicht in eine Textdatei geschrieben werden. Diese Datei kann dann wieder eingelesen werden, um die Klassen wieder herzustellen.

Im Benutzerinterface (Abbildung 6.1) kann diese Funktion verwendet werden, indem „save metadata to file“ ausgewählt und ein Dateiname eingegeben wird. Dann wird für jedes generierte Skelett eine Textdatei mit dem angegebenen Namen generiert, die die serialisierten Daten enthält. Um ein Skelett aus solch einer Textdatei zu laden, muss die entsprechende Datei bei „load from file“ ausgewählt werden. Dann werden die dazugehörigen Java-Klassen wieder hergestellt und das gleiche 3D-Modell noch einmal generiert.

Das funktioniert natürlich nur über das implementierte Programm, das genau die serialisierten Java-Klassen enthält und liefert keine Zusatzinformationen zu dem generierten Skelett an den Benutzer. Solche Zusatzinformationen könnten aber hilfreich für die Weiterverarbeitung des erzeugten Modells sein. In weiterführenden Arbeiten könnten sie zusätzlich generiert werden. Hilfreiche Informationen könnten dabei z. B. die Knochenhierarchie oder die Winkeleinschränkungen an den Gelenken sein.

Zusätzlich gibt es die Funktion konkrete PCA-Eingabebeispiele auszuwählen (construct from example). Damit wird ein Skelett mit den Daten generiert, die für dieses Beispiel erhoben wurden. Mit einem PCA-Punkt sind aber noch nicht alle Daten festgelegt, die ein Skelett bestimmen. Es ist nur festgelegt welcher PCA-Punkt in Schritt 2 des Algorithmus ausgewählt wird, nicht welche konkreten Parameter die Grammatik bekommt oder welche Typen von Extremitäten generiert werden (siehe Abschnitt 5.1). Deshalb werden zusätzlich die Benutzereingaben, z. B. für die Anzahl der Arme, berücksichtigt.

6.3. Erzeugung von Variationen

Sucht man Inspiration zu einem bestimmten Typ von Skelett oder Tier, so kann es hilfreich sein Variationen zu einem vorgegebenen Skelett generieren zu können. Vielleicht wurde ein interessantes Skelett generiert und man möchte dazu schnell viele ähnliche Skelette finden. Dies ist mit dem hier vorgestellten Algorithmus relativ leicht möglich. In der Benutzeroberfläche muss „create variations“ ausgewählt werden.

Ein möglicher Ansatz wäre zur Erzeugung von Variationen genetische Algorithmen (siehe Abschnitt 3.7) zu verwenden. Die PCA gibt aber schon einen Raum vor, in dem relativ leicht zufällige Punkte erzeugt werden können. Das lässt sich zur Erzeugung von Variationen gut ausnutzen. Deshalb wurden hier keine genetischen Algorithmen verwendet.

Als Grundlage für Variationen kann ein gespeichertes Skelett oder ein Eingabebeispiel der PCA dienen (siehe Abschnitt 6.2). Es ist also mindestens ein Punkt mit den erhobenen Merkmalen für die PCA gegeben. Bei einem gespeicherten Skelett ist zusätzlich noch gegeben wieviele Wirbel, Rippen und Extremitäten generiert werden und welche Art von Extremität an welchem Ansatzpunkt beginnt. Mit Ansatzpunkten sind hier die Elternknochen von Extremitäten, also Schulterblatt und Beckenknochen(2), gemeint. Ist nur ein Eingabebeispiel der PCA gegeben, so fehlen diese Angaben. Es werden aber zusätzlich Benutzereingaben, z. B. zur Anzahl verschiedener Extremitätentypen, berücksichtigt.

Für die Variationen wird zwischen Eingabemerkmale für die PCA und unabhängig generierten Daten unterschieden. Um Variationen von einem Punkt $p = (x_1, x_2, \dots, x_n)$ im PCA-Raum zu generieren, wird für jede Dimension eine neue Zufallszahl erzeugt. Dies ergibt dann einen Punkt $p' = (x'_1, x'_2, \dots, x'_n)$. Ein x'_i wird mit einer Normalverteilung zum Erwartungswert x_i erzeugt. Der konkrete Wert für die Varianz wurde experimentell bestimmt. Er sollte nicht zu groß sein, damit die generierten Skelette sich ähneln, aber auch nicht zu klein, damit sie sich etwas unterscheiden. In der Implementierung wird die Hälfte der Varianz verwendet, die die PCA für Dimension i berechnet hat, also die Hälfte des Eigenwerts, der zur i -ten Dimension gehört.

Bei den Merkmalen, die unabhängig von der PCA sind, wird unterschiedlich vorgegangen. Wieviele Wirbel und Rippen generiert werden sollen wird einfach neu zufällig bestimmt. Dies ist also unabhängig vom gegebenen Skelett.

Für Typ und Anzahl der Extremitäten wird etwas mehr Aufwand betrieben, da dies mehr Einfluss auf das Aussehen des Skeletts hat. Sind noch keine Ansatzpunkte und Typen von Extremitäten vorgegeben, werden diese zunächst berechnet. Dies passiert, wenn kein gespeichertes Skelett geladen, sondern ein PCA-Beispiel verwendet wird. Danach wird ein Ansatzpunkt ausgelost. Wurde diesem Ansatzpunkt noch keine Extremität zugeteilt oder

ist noch Platz für eine weitere, wird mit Wahrscheinlichkeit 0,5 eine Extremität mit einem Typ, der an diesen Ansatzpunkt passt, hinzugefügt. Gibt es an dem Punkt hingegen schon eine Extremität, so wird ihr Typ, ebenfalls mit Wahrscheinlichkeit 0,5, verändert. Wurde eine Veränderung ausgelost, wird die Extremität an dem gewählten Ansatzpunkt gelöscht und mit einer gewissen Wahrscheinlichkeit¹ wird eine Extremität mit gleichem Typ an einem anderen Ansatzpunkt wieder eingefügt, falls möglich.

Diese Prozedur verschiebt keine Knochen. Sie wird vor Ausführung der Grammatik durchgeführt. Das bedeutet, es wird bestimmt an welchen Stellen wieviele Vorder- und Hinterextremitäten generiert werden (Parameter $e_h, e_{h2}, e_s, e_{s2}, e_b$ und e_{b2}) und welchen Typ sie haben, also ob sie bei der Positionierung als Beine, Flügel, Flossen oder Arme behandelt werden. Die Längen der Extremitäten werden durch p' vorgegeben und hängen nur davon ab, ob es sich um Vorder- oder Hinterextremitäten handelt.

Sind alle Parameter für die Grammatik und die Typen der Extremitäten festgelegt, werden die Skelettelemente durch die Grammatik neu generiert. Dabei werden auch die Extremitäten neu positioniert.

¹Im Code wurde hier eine Wahrscheinlichkeit von 0,2 verwendet.

7. Implementierungsdetails

7.1. Aufbau der Software

Die Implementierung wurde mit der Programmiersprache Java durchgeführt. In Abbildung 7.1 ist der Programmaufbau stark vereinfacht dargestellt. Das Diagramm ist an UML-Klassendiagramme angelehnt. Pfeile stehen jedoch nicht für Teilbeziehungen oder Vererbung, sondern für den Datenfluss.

Die zentrale Klasse ist der „SkeletonGeneratorHandler“. Er steuert das Zusammenspiel der einzelnen Komponenten. Externe Daten bekommt er über den „PcaDataReader“ und die Benutzeroberfläche (GUI). Der „PcaDataReader“ liest die Beispiele für die PCA (PcaDataPoints) ein. Aus den Benutzereingaben ergeben sich die Bedingungen für die Verteilung, die durch die PCA untersucht werden soll. Diese und die eingelesenen Beispiele werden dann dem „PcaHandler“ übergeben, der alles verwaltet, was mit der PCA zu tun hat.

Der „SkeletonGenerator“ ist dafür verantwortlich die Einzelteile des Skeletts zu generieren. Zu jedem Zeitpunkt hält er alle aktuell generierten terminalen und nichtterminalen Elemente. Die Elemente speichern jeweils selbst welches ihr Elternelement und welches ihre Kindelemente sind. Es reicht also die Elemente unsortiert in Listen zu speichern. Die Ersetzungsregeln der Grammatik sind im „RuleDictionary“ und die Parameter für die Grammatik in „SkeletonMetaData“ gespeichert. Die Klasse „SkeletonMetaData“ implementiert das „Serializable“ Interface [26] und wird, wenn ein Skelett gespeichert werden soll, als Textdatei herausgeschrieben (siehe Abschnitt 6.2).

Sind alle Einzelteile generiert, werden alle zu spiegelnden Elemente mit der Methode „calculateMirroredElements()“ gespiegelt. Das fertige Skelett geht dann an den „Obj-Generator“, der ein 3D-Modell mit Hilfe der 3D-Modelle aus der „BoneLibrary“ erstellt. In der „BoneLibrary“ sind alle zur Verfügung stehenden 3D-Modelle einzelner Knochen gespeichert.

7.2. Dateiformate für 3D-Modelle

Das einfachste Format zur Darstellung von 3D-Objekten ist das offene Dateiformat Wavefront OBJ [50]. Es kann nur geometrische Formen speichern, keine zusätzlichen Informationen wie z. B. Gelenke und deren Einschränkungen.

Andere weit verbreitete Formate sind FBX von Autodesk [5] und Alembic [3]. Sie bieten mehr Möglichkeiten als OBJ, sind aber auch ungleich aufwändiger programmatisch zu erzeugen.

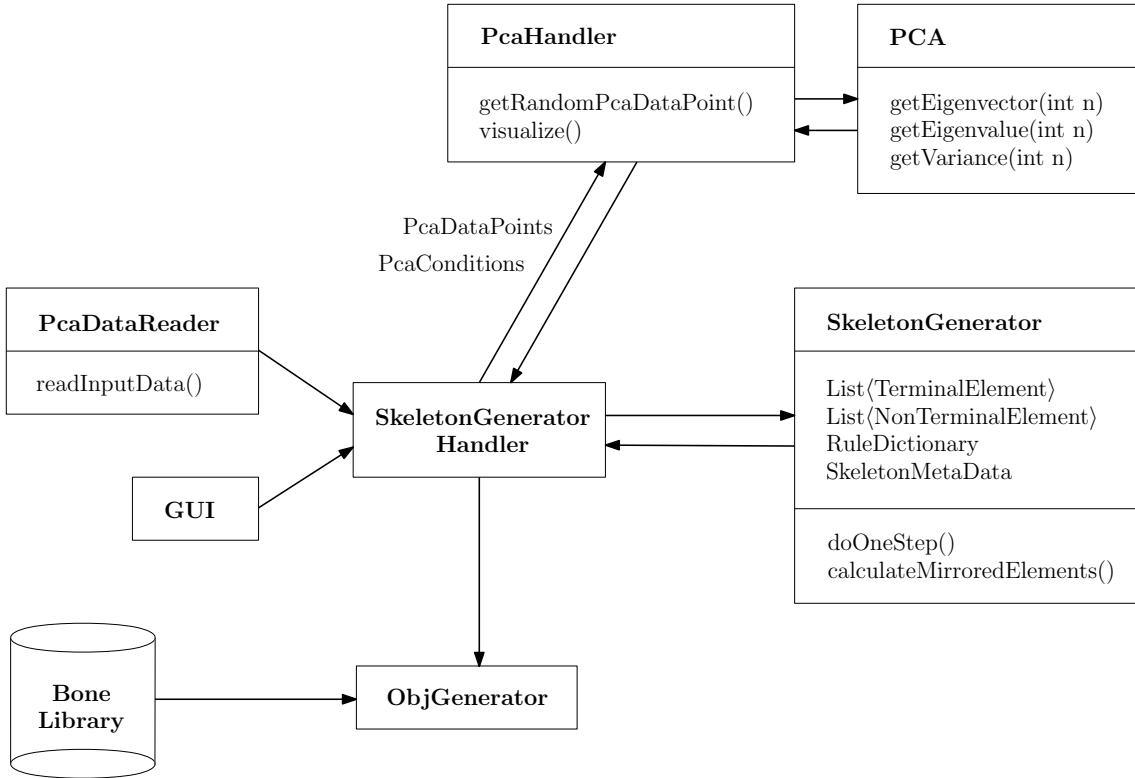


Abbildung 7.1.: Stark vereinfachte Darstellung der Softwarearchitektur. Jeder Kasten steht für eine Klasse, wie in UML-Klassendiagrammen. Pfeile sollen den Datenfluss darstellen.

Um das 3D-Modell des erzeugten Skeletts zu speichern reicht das OBJ-Format aus. Zur Weiterverarbeitung, z. B. für Animationen, können die Daten dann leicht in das dann benötigte Dateiformat umgewandelt werden. Für das Laden und Schreiben von OBJ-Dateien wird die Java-Bibliothek Obj [34] verwendet. Sollen in Zukunft zusätzliche Daten exportiert werden, dann müsste neu evaluiert werden, welches Dateiformat am besten geeignet ist.

7.3. Transformationsmatrizen

Jedes Kindelement k im Skelett speichert eine homogene Transformationsmatrix T_k , die seine lokalen Koordinaten in das Koordinatensystem des Elternelements e umrechnet (siehe Abschnitt 5.2 und Abbildung 7.2a). In einer Matrix sind also eine Translation und eine Rotation gespeichert. Es treten verschiedene Situationen auf, in denen unterschiedliche Transformationsmatrizen berechnet werden müssen.

In Schritt 4 des Algorithmus werden z. B. alle Elemente, die nicht auf der Wirbelsäule liegen, gespiegelt (siehe Abschnitt 5.1). Wird hier einfach eine Spiegelung an der xy -Ebene angewendet, so ist das resultierende Koordinatensystem nicht mehr rechtshändig, sondern linkshändig. Das führt beispielsweise dazu, dass im neuen Koordinatensystem nicht mehr gegen den Uhrzeigersinn, sondern mit dem Uhrzeigersinn gedreht wird. Das kann zu Problemen führen, wenn auf den gespiegelten Elementen Transformationen ausgeführt werden müssen, v. a. wenn linkshändige und rechtshändige Koordinatensysteme verwendet werden. Außerdem kann es dazu führen, dass die Flächen der 3D-Modelle so interpretiert werden, als würde ihre Vorderseite nach innen zeigen, anstatt nach außen. Wird beim Rendering Backface Culling [8] eingesetzt, führt das dazu, dass diese Flächen nicht sichtbar sind.

Um den Überblick über die Berechnung der Transformationsmatrizen in den verschiedenen Situationen zu behalten, wurden vier Übersichtsgrafiken erstellt (Abbildung 7.2 a und b, Abbildung 7.3 und Abbildung 7.4). Im Folgenden wird jeweils erklärt, welche Situationen sie beschreiben, wann diese auftreten und wie die entsprechenden Transformationsmatrizen berechnet werden.

- In Abbildung 7.2 b ist die Situation dargestellt, die u. a. dann auftritt, wenn die Position der Wirbel auf der Wirbelsäule berechnet werden soll. Die Position der Wirbel im globalen Koordinatensystem ist bekannt, ihre Position soll aber im lokalen Koordinatensystem des Elternelements angegeben werden.
Es ist also ein Elternelement e mit zugehörigem Kindelement k gegeben. Für beide Elemente ist die Transformation ins globale Koordinatensystem T_{eg} bzw. T_{kg} bekannt. Die lokale Transformationsmatrix von k ist dann $T_k = T_{eg}^{-1} \cdot T_{kg}$.
- Die Situation, die entsteht, wenn das Kindelement k eines Elements e , das auf der Wirbelsäule liegt, gespiegelt werden soll, ist in Abbildung 7.3 dargestellt. Gesucht ist die Transformation T_{kr} , die die lokalen Koordinaten des gespiegelten Kindelements k_r in die lokalen Koordinaten von e umwandelt. Das Element k_l entsteht, wenn k einfach an der xy -Ebene gespiegelt wird. Dieses Element hat aber kein rechtshändiges Koordinatensystem mehr, sondern ein linkshändiges. Die Grafik macht deutlich, dass $T_{kr} = T_{eg}^{-1} \cdot T_r \cdot T_{kl} \cdot T_{kx}$. Die Matrix T_r spiegelt die z -Koordinate und T_{kx} wandelt das linkshändige Koordinatensystem von k_l in das rechtshändige von k_r um. Diese Umwandlung T_{kx} besteht aus einer Verschiebung in z -Richtung um die Länge von k in z -Richtung und einer Spiegelung an der xy -Ebene.
- Abbildung 7.4 zeigt die Situation, die entsteht, wenn das Elternelement e selbst schon gespiegelt wurde. Es ist also ein gespiegeltes Elternelement e_r gegeben und das dazugehörige gespiegelte Kindelement k_r . Gesucht ist die Transformationsmatrix T_{kr} . Wie in Abbildung 7.3 sind die linkshändigen Koordinatensysteme der gespiegelten Elemente e_l und k_l gepunktet dargestellt. Die Transformationen T_{ex} und T_{kx} wandeln linkshändige Koordinatensysteme in rechtshändige um. Dann ergibt sich $T_{kr} = T_{ex} \cdot T_k \cdot T_{kx}$, da $T_k = T_{kl}$.

7.4. PCA

7.4.1. Annotation der Bilder

Die Bilder der Skelette wurden folgendermaßen für die Datenerhebung vorbereitet:

1. Zuschneiden des Bildes, so dass möglichst nur das Skelett mit wenig Rand außen herum zu sehen ist.
2. Einfügen in eine 1000×1000 Pixel große Bildumgebung.
3. Vergrößern des Bildes, sodass es möglichst viel Platz innerhalb der Bildumgebung einnimmt, ohne die Seitenverhältnisse zu verändern.
4. Verschieben innerhalb der Bildumgebung an den unteren Rand und horizontal in die Mitte.

Ist das geschehen kann die Lage der Wirbelsäule und die Länge der Knochen der Extremitäten annotiert werden.

Die Annotation der Bilder wurde per Hand mit dem Programm Inkscape [25] durchgeführt. Es kann verwendet werden um Vektorgrafiken zu erstellen und zu bearbeiten.

Für jedes zu markierende Element wurde eine Strecke oder eine Bézierkurve eingefügt

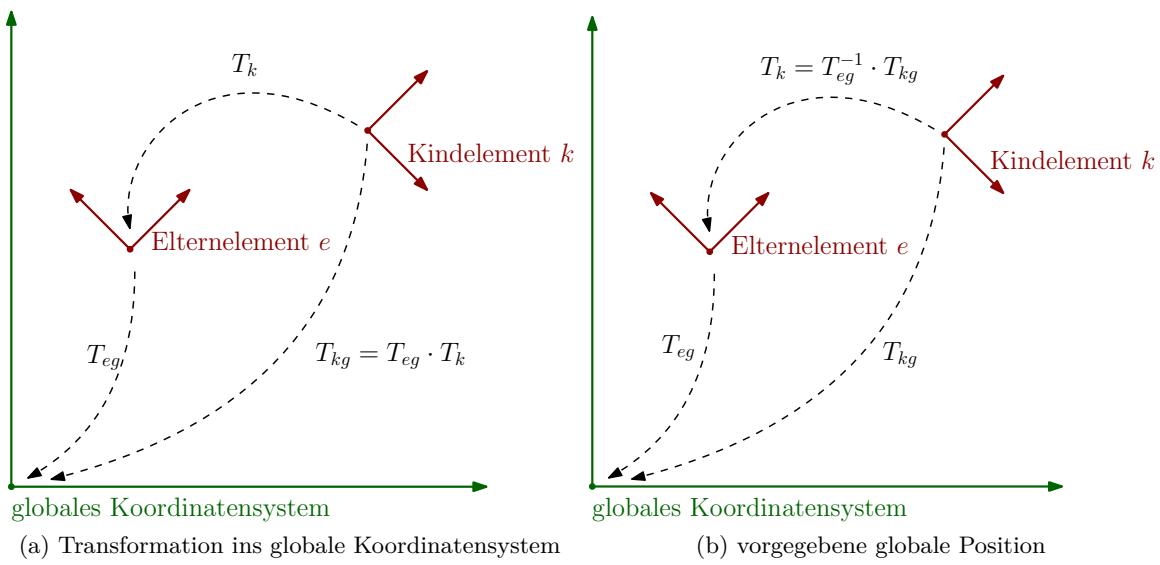


Abbildung 7.2.: Jedes Kinderelement k speichert eine Transformation T_k ins lokale Koordinatensystem seines Elternelements e . (a) Ist T_k gegeben, so lässt sich die Transformation T_{kg} des lokalen Koordinatensystems von k ins globale Koordinatensystem durch $T_{kg} = T_{eg} \cdot T_k$ berechnen. (b) Ist T_{kg} gegeben, so lässt sich $T_k = T_{eg}^{-1} \cdot T_{kg}$ berechnen.

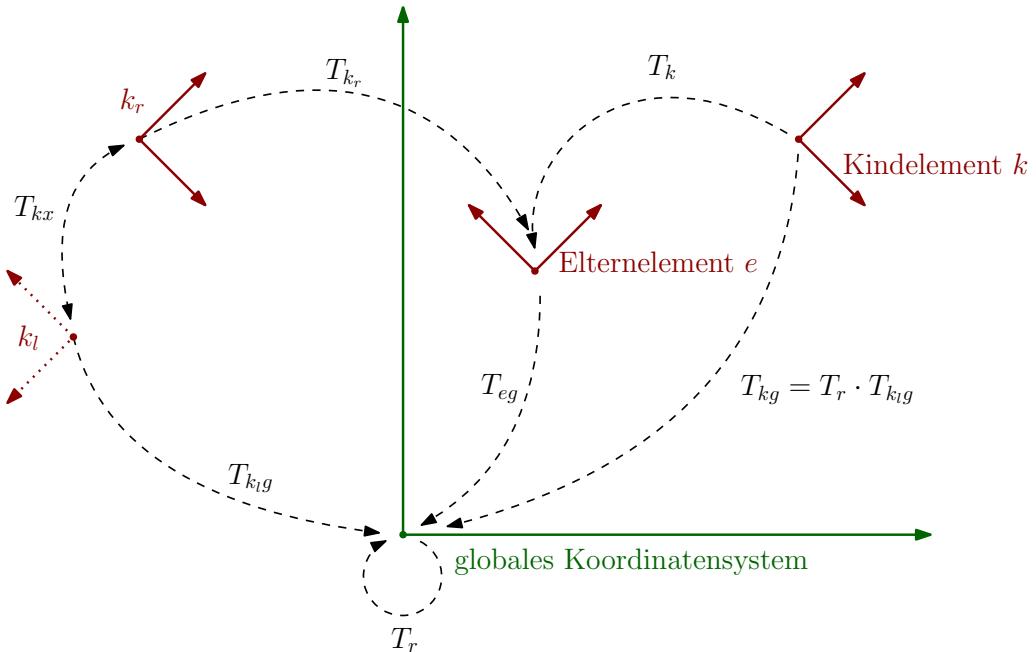


Abbildung 7.3.: Die Transformationsmatrix T_{kr} des gespiegelten Kinderelements k_r lässt sich berechnen durch $T_{kr} = T_{eg}^{-1} \cdot T_r \cdot T_{kg} \cdot T_{kx}$. Die Matrix T_r spiegelt die z -Koordinate und T_{kx} wandelt das linkshändige Koordinatensystem von k_l in das rechtshändige von k_r um.

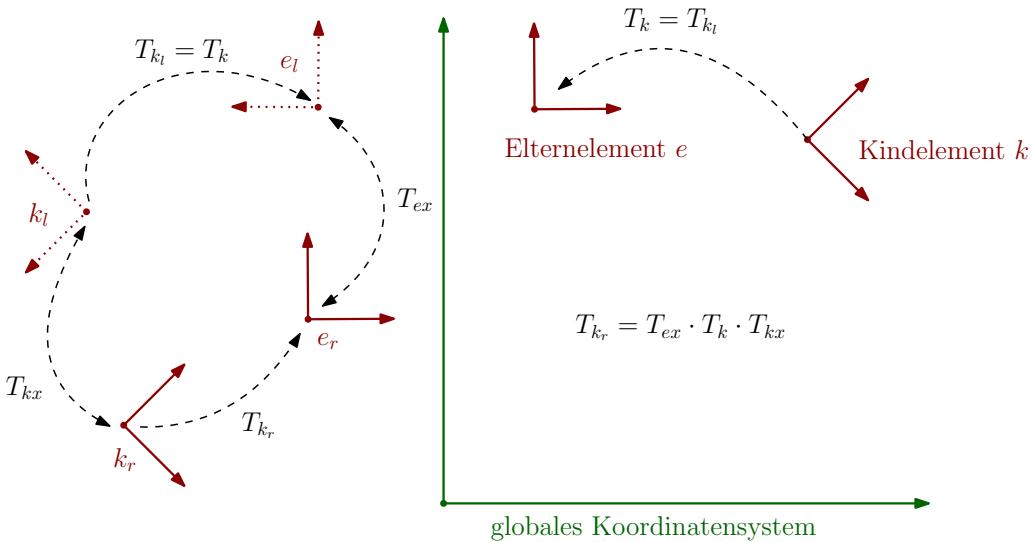


Abbildung 7.4.: Die Transformationsmatrix T_{k_r} des gespiegelten Kindelements k_r eines ebenfalls gespiegelten Elternelements e_r lässt sich berechnen durch $T_{k_r} = T_{ex} \cdot T_k \cdot T_{kx}$. Die Matrizen T_{ex} und T_{kx} wandeln das linkshändige Koordinatensystem von e_l bzw. k_l in das rechtshändige von e_r bzw. k_r um.

und mit einem vorher festgelegten Namen benannt. Diese Elemente wurden dann durch Inkscape als Pfade in der erzeugten svg-Datei gespeichert. Aus dieser Datei wurden dann automatisiert die eingetragenen Pfade mit ihren Koordinaten ausgelesen.

Es ist wichtig folgende Details zu beachten, damit dieser Vorgang reibungslos abläuft.

- Man kann in Inkscape einstellen, dass Koordinaten immer absolut angegeben werden. Das ist sinnvoll um die Koordinaten leichter auslesen zu können.
- Der Ursprung des Koordinatensystems in Inkscape ist unten links, im svg-Format ist er aber oben links.
- Ebenen in Inkscape sollten nicht verschoben sein, sonst verschieben sich mit ihnen auch die Koordinaten.

7.4.2. Anpassung des Verlaufs der Wirbelsäule

Die Wirbelsäule, die durch einen Punkt im PCA-Raum vorgegeben wird, besteht aus drei einzelnen Bézierkurven. Diese Bézierkurven haben an den Punkten, an denen sie zusammenstoßen die gleichen Koordinaten. Die Gesamtkurve der Wirbelsäule muss an diesen Punkten aber nicht unbedingt differenzierbar sein. Deshalb müssen vor der Weiterverwendung die jeweils nächsten Kontrollpunkte vor und nach dem Kontaktspunkt verschoben werden.

Das wurde hier so gemacht, dass die zu verschiebenden Kontrollpunkte jeweils um den gleichen Winkel aber in gegensätzliche Richtungen rotiert wurden (siehe Abbildung 7.5).

Grundsätzlich könnten die Kontrollpunkte beliebig auf einer gewählten Tangente des Kontaktspunkts platziert werden (rot im Bild) um eine übereinstimmende Steigung zu bekommen. Um den Verlauf der Wirbelsäulenteile möglichst wenig zu verändern, ist es von Vorteil auch die Kontrollpunkte möglichst wenig zu verschieben. Man könnte die Punkte z. B. auch senkrecht auf die Tangente projizieren. Welches Verfahren angewendet wird, ist jedoch nicht von großer Bedeutung.

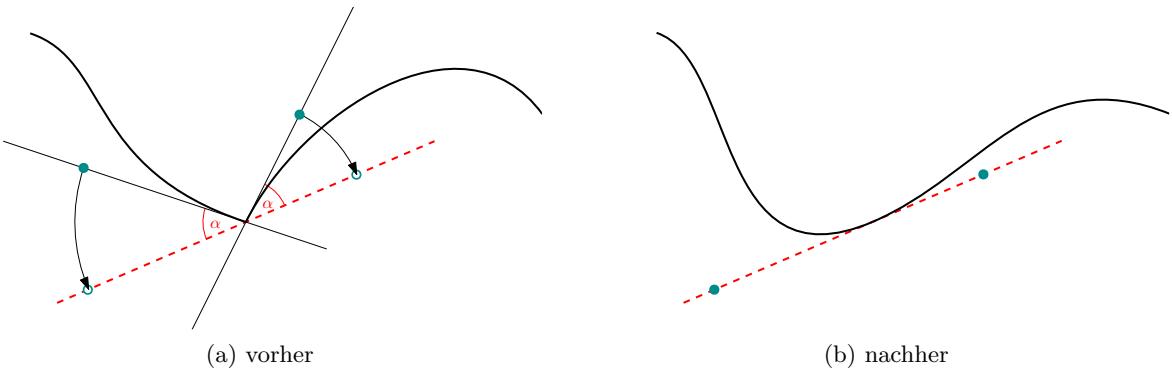


Abbildung 7.5.: Anpassung der Kontrollpunkte der Wirbelsäulenteile, wenn die Steigung an den Kontaktstellen ungleich ist. Die beiden Teile der Wirbelsäule und die Steigung am Kontaktstellen sind hier in schwarz dargestellt. Die zu drehenden Kontrollpunkte in cyan. In rot ist die resultierende Steigung und der Winkel, der für die Drehung verwendet wird, zu sehen.

7.5. Probleme der Positionierung von kurzen Beinen

Wie in Absatz 5.5.2 bereits beschrieben, treten bei der Ausrichtung sehr kurzer Beine ein paar Probleme auf. Diese werden im Folgenden kurz umrissen. Da die Beine in diesen Fällen aber sehr kurz sind, hat die konkrete Positionierung keine große Auswirkung auf die Gesamtwirkung des Skeletts.

- Bei der Berechnung der Bodenhöhe wird die Beinlänge von den entsprechenden Kontrollpunkten der Bézierkurve aus gemessen, da die Position des Schulterblatts bzw. des Beckenknochens in diesem Stadium noch nicht klar ist. Deshalb kann es bei sehr kurzen Beinen sein, dass der Abstand zwischen Boden und Gelenk länger als das Bein ist und der Boden nicht erreicht werden kann.

In diesem Fall wird das Bein einfach komplett ausgestreckt, senkrecht nach unten positioniert. Da der Abstand zwischen dem Hüft- bzw. Schultergelenk und dem Kontrollpunkt der Bézierkurve nicht sehr groß ist, fällt auch der Abstand zur Bodenhöhe nicht allzu sehr auf.

- Bei sehr kurzen Knochen ändert sich der Abstand zum Boden durch Drehung der Gelenke nicht so stark wie bei langen Knochen. Wenn der Winkel sich zu wenig ändert, wird aber im Algorithmus davon ausgegangen, dass die Winkeländerung zu klein ist und alle Änderungen sofort wieder zurückgesetzt wurden. Deshalb wird die Winkeländerung dann für die nächste Iteration stärker verkleinert. Das ist aber in diesem Fall kontraproduktiv. Der Algorithmus schafft es dann nämlich nicht mehr die Knochen in die richtige Lage zu bringen, da der Bewegungsspielraum dadurch zu stark eingeschränkt wird.

Dem könnte man entgegenwirken, indem man abfragt, wie oft die Drehungen in der jeweiligen Iteration zurückgesetzt wurden, anstatt die Änderung des Abstandes zum Boden zu messen.

- Bei wirklich sehr kurzen Beinen (hier eine Gesamtlänge unter 5) macht es gar keinen Sinn sie anzugeben, da man sie kaum sieht. Außerdem tritt hier der Effekt auf, der in Punkt 2 schon beschrieben wurde.
 - Die Knochen der Beine können schon vor der ersten Iteration unterhalb der Bodenhöhe liegen. Das tritt auf, wenn mindestens ein Paar Beine sehr kurz ist und die

Wirbelsäule an den Ansatzpunkten auf sehr unterschiedlicher Höhe liegt (siehe auch Absatz 5.5.1 zur Berechnung der Bodenhöhe).

In diesem Fall ist der Algorithmus natürlich nicht anwendbar. Das Bein wird dann einfach senkrecht nach unten ausgerichtet und mit einem Fuß versehen, der mit der Sohle auf dem Boden steht.

5. Bei kurzen Beinknochen kann es dazu kommen, dass der Oberschenkel nicht näher zum Boden bewegt werden kann, weil er schon aufliegt, aber Unterschenkel und Fuß durch das Gelenkoffset (siehe Abschnitt 5.6 zur Positionierung der Knochenmodelle) über dem Boden schweben und nicht näher zum Boden bewegt werden können. Dieser Fall wird nicht abgefangen. Der Fuß berührt dann also nicht den Boden.

8. Fazit und Ausblick

Zusammenfassung

Das Ergebnis dieser Arbeit ist ein Algorithmus, der 3D-Modelle von Wirbeltierskeletten generieren kann, und eine Implementierung desselben. Die Informationen, auf deren Grundlage der Algorithmus arbeitet, sind annotierte 2D-Bilder von Wirbeltierskeletten, z. B. aus Lehrbüchern der Zoologie. Mit Hilfe einer *Principal Component Analysis* auf diesen Datenpunkten (siehe Kapitel 4) erhält man eine Möglichkeit zufällig normalverteilte Punkte zu erzeugen, die der gleichen Verteilung folgen wie die Datenpunkte. Ein so erzeugter Punkt liefert alle Informationen, die nötig sind, um ein Skelett zu generieren. Diese Technik liefert ohne weitere Einschränkungen viele unterschiedliche Skelette, die in den allermeisten Fällen auch gut aussehen.

Außerdem können gezielt Skelette mit bestimmten Eigenschaften generiert werden, indem die *Principal Component Analysis* auf bedingten Verteilungen ausgeführt wird oder Variationen zu schon generierten Skeletten erzeugt werden. Dies ist sehr hilfreich, wenn man schon eine Vorstellung davon hat, wie das zu generierende Skelett ungefähr aussehen soll.

Zusätzlich zu „realistischen“ Wirbeltierskeletten lassen sich mit dem Algorithmus auch fantastische Skelette generieren. Es können zusätzliche Extremitäten und ein zusätzlicher Schultergürtel erzeugt werden. Die Datengrundlage bleibt jedoch die selbe. Es werden die gleichen Bilder von echten Wirbeltierskeletten verwendet. Deshalb wurde darauf geachtet, dass sich die generierten Skelette durch die genannten Veränderungen nicht zu stark von echten Wirbeltierskeletten entfernen.

Eine kontextfreie Grammatik legt fest, welche Knochen generiert werden und wie die Baumstruktur des Skeletts aufgebaut ist. Diese Grammatik sorgt dafür, dass die Ersetzungsregeln relativ übersichtlich dargestellt und leicht erweiterbar sind. Da sie aber im Voraus, durch den zufällig erzeugten Punkt, schon so stark festgelegt wird, dass immer genau ein Wort generiert wird, ist sie nicht unbedingt notwendig.

Die Positionierung der Beine liefert in vielen Fällen keine schönen Ergebnisse, da hierzu keine Informationen aus den 2D-Bildern erhoben werden. Trotzdem kann das generierte Skelett als erster Eindruck bzw. Inspiration für ein Modell dienen. Wenn das erzeugte Skelett die Grundlage für ein animiertes Wesen sein soll, muss die Positionierung der Knochen sowieso noch öfter angepasst werden.

Zukünftige Arbeiten

- schnell viele verschiedene Skelette generierbar, Variationen zu einem vorgegebenen Skelett, leicht bedienbar
- mehr Daten für PCA sammeln
- Erweiterung auf Mensch? Mensch auch als PCA Datenpunkt?
- austauschbare 3D-Modelle, mehr verschiedene 3D-Modelle
- Gewicht in Algo verwenden
- Interaktivität: Teile, die einem nicht gefallen, sollten geändert werden können (Eine Anwendung, bei der nach Eingabe von Parametern sofort das komplette Tier generiert wird, ist weniger hilfreich als eine, bei der schrittweise Teile davon generiert werden können (und auch rückgängig gemacht werden können))
- Muskeln, Haut etc. anbauen (an 3D-Modell oder auch Algo erweitern)

Literatur

- [1] *3ds Max*. URL: <https://www.autodesk.de/products/3ds-max/overview> (besucht am 21.05.2020).
- [2] *3ds Max. Understanding Biped*. URL: <https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2017/ENU/3DSMax/files/GUID-C828940D-A40E-4A4C-96C5-D936DC3951E1-htm.html> (besucht am 21.05.2020).
- [3] *Alembic*. URL: <http://www.alembic.io/> (besucht am 27.05.2020).
- [4] A. Aristidou u. a. “Inverse Kinematics Techniques in Computer Graphics: A Survey”. In: *Computer Graphics Forum* 37.6 (2018), S. 35–58. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13310>.
- [5] *Autodesk FBX*. URL: <http://help.autodesk.com/view/FBX/2019/ENU/> (besucht am 27.05.2020).
- [6] *Avatar Schreckenspferd*. URL: <https://james-camerons-avatar.fandom.com/de/wiki/Schreckenspferd> (besucht am 21.05.2020).
- [7] Esben Bach und Andreas Madsen. “Building and Generating Facial Textures using Eigen Faces”. Masterarbeit. Harbin Institute of Technology und Linköping University, 2016.
- [8] *Backface Culling*. URL: <https://wiki.delphigl.com/index.php/Backfaceculling> (besucht am 28.05.2020).
- [9] Nuno Barreto und Licinio Roque. “A Survey of Procedural Content Generation tools in Video Game Creature Design”. In: *Second Conference on Computation Communication Aesthetics and X*. Juni 2014. URL: https://www.researchgate.net/publication/298307183_A_Survey_of_Procedural_Content_Generation_tools_in_Video_Game_Creature_Design.
- [10] Peter Bentley. “An Introduction to Evolutionary Design by Computers”. In: *Evolutionary Design by Computers* (Jan. 1999). URL: https://www.researchgate.net/publication/244418116_An_Introduction_to_Evolutionary_Design_by_Computers.
- [11] Volker Blanz und Thomas Vetter. “A Morphable Model for the Synthesis of 3D Faces”. In: *SIGGRAPH'99 Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (Sep. 2002). URL: <https://dl.acm.org/doi/10.1145/311535.311556>.

- [12] *Demoszene*. URL: <https://www.demoscene.info/> (besucht am 06.05.2020).
- [13] *Development of No Man's Sky*. URL: https://en.wikipedia.org/wiki/Development_of_No_Man%27s_Sky (besucht am 21.05.2020).
- [14] Morris L. Eaton. *Multivariate Statistics. A Vector Space Approach*. Lecture Notes-Monograph Series. Beachwood, Ohio: Institute of Mathematical Statistics, 1983. ISBN: 9780940600690. URL: <https://epdf.pub/multivariate-statistics-a-vector-space-approacheed2a525dca43e48f58a88b289ff519f66368.html>.
- [15] Wilhelm Ellenberger. *Handbuch der Anatomie der Tiere für Künstler*. Bd. 1. Leipzig: Dietrichsche Verlagsbuchhandlung, 1900. URL: https://archive.org/details/bub_gb_4JFmAAAMAAJ/page/n19/mode/2up.
- [16] Jonas Freiknecht und Wolfgang Effelsberg. "A Survey on the Procedural Generation of Virtual Worlds". In: *Multimodal Technologies and Interaction*. Bd. 1. 4. Artikel 27. 2017. URL: <https://doi.org/10.3390/mti1040027>.
- [17] Michael Girard und Anthony Maciejewski. "Computational Modeling for the Computer Animation of Legged Figures". In: *ACM Siggraph Computer Graphics*. Bd. 19. Juli 1985, S. 263–270. DOI: 10.1145/325334.325244. URL: <https://dl.acm.org/doi/10.1145/325334.325244>.
- [18] Ernst Hadorn und Rüdiger Wehner. *Allgemeine Zoologie*. 21. Aufl. Stuttgart: Georg Thieme Verlag, 1986. ISBN: 3-423-04061-0.
- [19] Michael A. Harrison. *Introduction to Formal Language Theory*. Addison-Wesley series in computer science. Addison-Wesley Publishing Company, 1978. ISBN: 0-201-02955-3.
- [20] *HDRI Haven. 360°-Bild einer kleinen Höhle*. URL: https://hdrihaven.com/hdri/?c=nature&h=small_cave (besucht am 20.05.2020).
- [21] Chris Hecker u. a. "Real-time motion retargeting to highly varied user-created morphologies". In: *ACM Trans. Graph. Article 27* (Sep. 2008). URL: <https://dl.acm.org/doi/10.1145/1360612.1360626>.
- [22] Milton Hildebrand und George E. Goslow. *Vergleichende und funktionelle Anatomie der Wirbeltiere*. Berlin, Heidelberg: Springer-Verlag, 2004.
- [23] John H. Holland. *Adaptation in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992. ISBN: 0-262-08213-6.
- [24] Jon Hudson. "Creature Generation using Genetic Algorithms and Auto-Rigging". Masterarbeit. Bournemouth University, National Centre for Computer Animation, 2013.
- [25] *Inkscape*. URL: <https://inkscape.org/de> (besucht am 28.05.2020).
- [26] *Java Serializable Interface*. URL: <https://docs.oracle.com/javase/7/docs/api/java/io/Serializable.html> (besucht am 25.05.2020).
- [27] *Java View*. URL: <http://www.javaview.de/> (besucht am 25.05.2020).
- [28] I.T. Jolliffe. *Principal Component Analysis*. 2. Aufl. New York: Springer-Verlag, 2002. ISBN: 0-387-95442-2.
- [29] Robert Krogh. "Procedural Character Generation. Implementing Reference Fitting and Principal Components Analysis". Masterarbeit. Aalborg University, 2007. URL: [https://projekter.aau.dk/projekter/en/studentthesis/procedural-character-generation-implementing-reference-fitting-and-principal-components-analysis\(8a268f9e-a382-496b-8a94-a94fe6606d2e\).html](https://projekter.aau.dk/projekter/en/studentthesis/procedural-character-generation-implementing-reference-fitting-and-principal-components-analysis(8a268f9e-a382-496b-8a94-a94fe6606d2e).html).

- [30] Holger Langkabel. "QQ-Plots als Instrument zum Vergleich von Verteilungen". In: *Konferenz der SAS-Anwender in Forschung und Entwicklung*. Bd. 20. März 2016, S. 105–117. URL: http://de.saswiki.org/wiki/KSFE_2016#Datavisualisierung.
- [31] *Maya*. URL: <https://www.autodesk.com/products/maya/overview> (besucht am 21.05.2020).
- [32] *Maya User Guide. Skeleton Hierarchy*. URL: <https://knowledge.autodesk.com/support/maya/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/Maya/files/GUID-DC88B9A7-593B-427E-9BED-4D7822B0E0B6-htm.html> (besucht am 21.05.2020).
- [33] *No Man's Sky*. URL: <https://www.nomanssky.com/> (besucht am 08.10.2019).
- [34] *Obj - a simple Wavefront OBJ file loader and writer*. URL: <https://github.com/javagl/Obj> (besucht am 27.05.2020).
- [35] *Our Centaurs Are Different aka: Centauroid Form*. URL: <https://tvtrtropes.org/pmwiki/pmwiki.php/Main/CentauroidForm> (besucht am 07.05.2020).
- [36] *Pegasus*. URL: <https://tvtrtropes.org/pmwiki/pmwiki.php/Main/Pegasus> (besucht am 07.05.2020).
- [37] Martin Petratsch. "Prozedurale Städtegenerierung mit Hilfe von L-Systemen". Großer Beleg. Technische Universität Dresden, 2008. URL: http://www.inf.tu-dresden.de/index.php?node_id=2234&ln=de.
- [38] Hartmut Prautzsch, Wolfgang Boehm und Marco Paluszny. *Bezier and B-Spline Techniques*. Berlin, Heidelberg: Springer-Verlag, 2002. ISBN: 3540437614.
- [39] Przemyslaw Prusinkiewicz und Aristid Lindenmayer. *The Algorithmic Beauty of Plants*. Berlin, Heidelberg: Springer-Verlag, 1990. ISBN: 0-387-97297-8. URL: <http://algorithmicbotany.org/papers/#abop>.
- [40] Eric N. Rittmeyer u. a. "Ecological Guild Evolution and the Discovery of the World's Smallest Vertebrate". In: *PLoS ONE* 7 (Jan. 2012). URL: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0029797>.
- [41] Hyewon Seo und Nadia Thalmann. "An Automatic modeling of human bodies from sizing parameters". In: *Proceedings of the Symposium on Interactive 3D Graphics*. Jan. 2003, S. 19–26. URL: <https://dl.acm.org/doi/10.1145/641480.641487>.
- [42] Harumichi Shinohara. "The Mouse Vertebrae: Changes in the Morphology of Mouse Vertebrae Exhibit Specific Patterns Over Limited Numbers of Vertebral Levels". In: *Okajimas folia anatomica Japonica* 76 (Juni 1999), S. 17–31. URL: https://www.jstage.jst.go.jp/article/ofaj1936/76/1/76_17/_pdf%EF%BB%BF.
- [43] *SpeedTree*. URL: <https://store.speedtree.com/> (besucht am 05.05.2020).
- [44] *Spore*. URL: www.spore.com (besucht am 21.05.2020).
- [45] Hideyuki Takagi. "Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation". In: *Proceedings of the IEEE* 89 (Okt. 2001), S. 1275–1296. URL: https://www.researchgate.net/publication/2985886_Interactive_evolutionary_computation_Fusion_of_the_capabilities_of_EC_optimization_and_human_evaluation.
- [46] *Vertebrate with Extra Limbs*. URL: <https://tvtrtropes.org/pmwiki/pmwiki.php>Main/VertebrateWithExtraLimbs> (besucht am 07.05.2020).
- [47] Günter Vogel und Hartmut Angermann. *dtv-Atlas zur Biologie*. 6. Aufl. Band II. München: Deutscher Taschenbuch Verlag GmbH & Co. KG, 1968. ISBN: 3-423-03012-7.

- [48] *Vogelskelett*. URL: <https://de.wikipedia.org/wiki/Vogelskelett> (besucht am 06.05.2020).
- [49] Hongling Wang, Joseph Kearney und Kendall Atkinson. “Arc-length parameterized spline curves for real-time simulation”. In: (Jan. 2002). URL: https://www.researchgate.net/publication/228718576_Arc-length_parameterized_spline_curves_for_real-time_simulation.
- [50] *Wavefront OBJ File Format Summary*. URL: <https://www.fileformat.info/format/wavefrontobj/egff.htm> (besucht am 27.05.2020).
- [51] Rüdiger Wehner und Walter Gehring. *Zoologie*. 24. Aufl. Stuttgart: Georg Thieme Verlag, 2007. ISBN: 978-3-13-367424-9.
- [52] Wilfried Westheide und Reinhard Rieger. *Spezielle Zoologie. Teil 2: Wirbel- oder Schädeltiere*. München: Spektrum Akademischer Verlag, 2004. ISBN: 3-8274-0307-3.
- [53] *Ziva*. URL: <https://zivadynamics.com/> (besucht am 21.05.2020).
- [54] *Ziva. General Character Setup*. URL: <https://zivadynamics.com/character-docs/general-setup> (besucht am 21.05.2020).

A. PCA Daten

A.1. Eingabedaten für PCA

A.1.1. Bilder

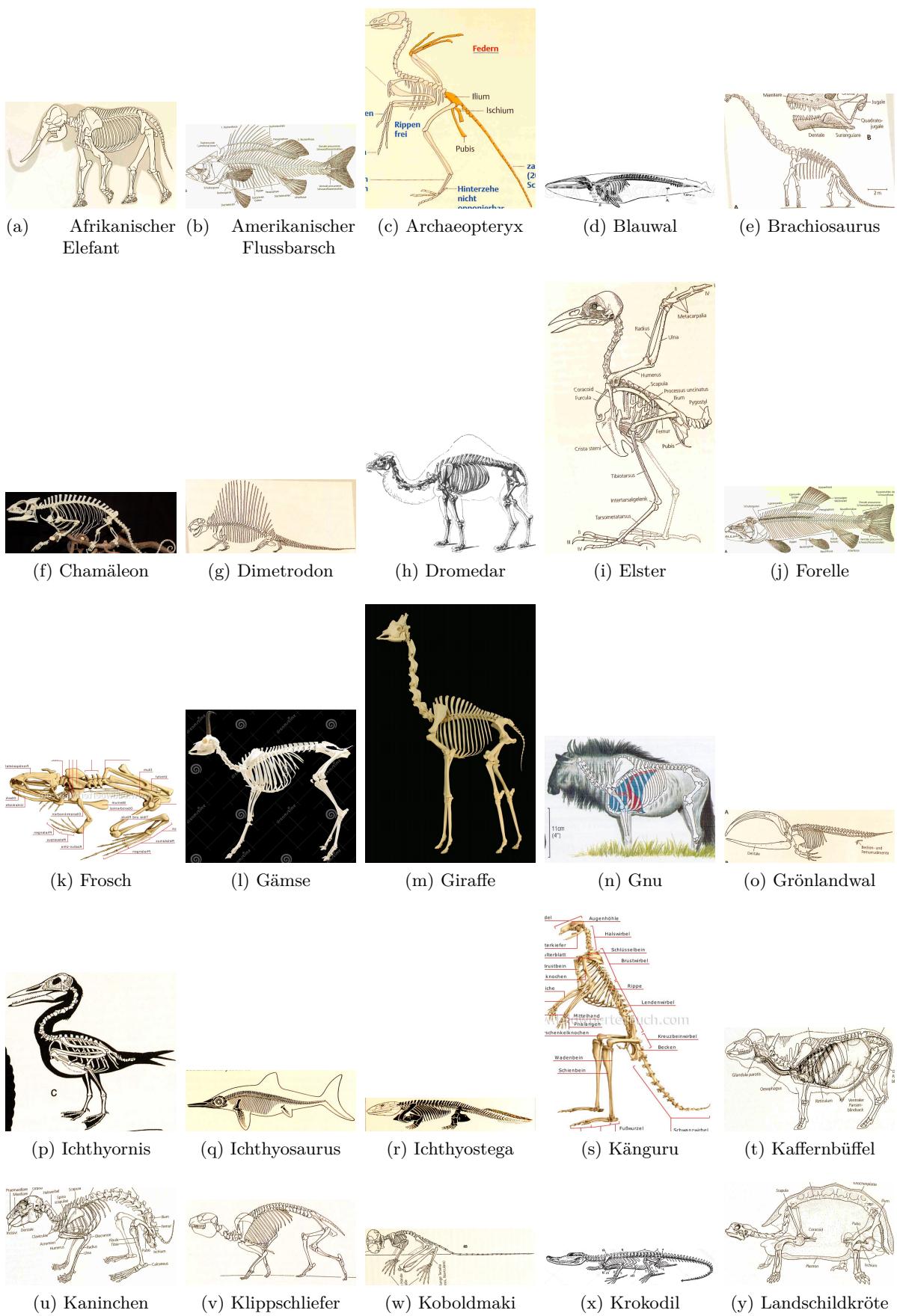
Alle Bilder, die als Eingabe für die PCA verwendet wurden, sind in Abbildung A.1 zu finden. Im Folgenden sind die Bildquellen aufgezählt. Alle Tiere, die nicht aufgezählt sind, sind aus [52] entnommen.

- Archaeopteryx, Ichthyosaurus, Ichthyostega, Urpferdchen aus [51]
- Blauwal <https://www.quagga-illustrations.de/wp-content/uploads/2014/05/h0001705.jpg>
- Chamäleon <https://www.madcham.de/de/anatomie/>
- Dormedar https://upload.wikimedia.org/wikipedia/commons/a/ac/Camel_Skeleton_-_Richard_Owen_-_On_the_Anatomy_of_Vertebrates_%281866%29.jpg
- Giraffe
https://de.wikipedia.org/wiki/Giraffen#/media/Datei:Giraffe_skeleton.jpg
- Gnu <https://lutzmoeller.net/Afrika/2007/Lutz-Juli/8-Gnu.php>
- Känguru <http://www.bildwoerterbuch.com/tierreich/beuteltiere/kaenguru/skelett-eines-kaengurus.php>
- Krokodil <https://de.depositphotos.com/210906852/stock-illustration-skeleton-crocodile-vintage-line-drawing.html>
- Pferd <https://www.kosmos.de/content/buecher/ratgeber/pferde-reiten/vorwaerts-abwaerts-eine-frage-der-haltung/>
- Schlange: zu Schlangen gibt es keine Bilder, die deren Skelett in ausgestrecktem Zustand von der Seite darstellen. Deshalb wurde ein leeres Bild genommen und der Verlauf des Rückens durch eine Gerade angenähert (Extremitäten besitzt eine Schlange nicht. Außerdem ist nicht erkennbar ob bzw. wo Hals in Rücken und Rücken in Schwanz übergeht. Deshalb wurde die komplette Wirbelsäule als Rücken markiert.)
- Schwan <https://www.alamy.de/skelett-eines-schwans-osteographia-oder-die-anatomie-der-knochen-london-1733-quelle-47-ich-12-kapitel-v-saitenhalter-autor-cheselden-william-image226921369.html>

- Sinornis und Taube aus [22]
- Strauß <https://www.alamy.de/stockfoto-skelett-von-strauss-24658845.html>
- Tyrannosaurus Rex https://upload.wikimedia.org/wikipedia/commons/9/9f/Tyrannosaurus_skeleton.jpg

A.1.2. Gewichte

- Afrikanischer Elefant 4000kg, <https://de.upali.ch/gewicht-und-grosse/>
- Afrikanischer Strauß bis 135kg, https://de.wikipedia.org/wiki/Afrikanischer_Stra%C3%9F
- Amerikanischer Flussbarsch 2kg, http://tierdoku.com/index.php?title=Amerikanischer_Flussbarsch
- Archaeopteryx 1kg, <https://de.wikipedia.org/wiki/Archaeopteryx>
- Blauwal 120 Tonnen, <http://tierdoku.com/index.php?title=Blauwal, „das schwerste bekannte Tier der Erdgeschichte“> <https://de.wikipedia.org/wiki/Blauwal>
- Brachiosaurus 23-44 Tonnen, <https://de.wikipedia.org/wiki/Brachiosaurus>
- Chamäleon 0,1-2kg, <https://www.tierchenwelt.de/echsen/128-chamaeleon.html>
- Dimetrodon 250kg, <https://de.wikipedia.org/wiki/Dimetrodon>
- Dromedar 300-700kg, <https://de.wikipedia.org/wiki/Dromedar>
- Durschnittsgewicht (Warmblut-)Pferd 600 kg, <https://www.reitarena.com/de/blog/blog-post/2015/03/03/das-pferd-grundlegende-fakten.html>
- Elster 0,2kg, <https://de.wikipedia.org/wiki/Elster>
- Forelle 10-50kg (je nach Art), <https://de.wikipedia.org/wiki/Forelle>
- Frosch 10g, <http://www.biologie-schule.de/frosch-steckbrief.php>
- Gämse 25-50kg, <https://de.wikipedia.org/wiki/G%C3%A4mse>
- Giraffe bis 2 Tonnen, <https://www.tierchenwelt.de/huftiere/73-giraffe.html>
- Gnu 140-250kg, <https://de.wikipedia.org/wiki/Gnus>
- Grönlandwal 50-100 Tonnen, <https://de.wikipedia.org/wiki/Gr%C3%B6nlandwal>
- Ichthyornis 0.3kg, http://dinodata.de/animals/birds/pages_i/ichthyornis.php
- Ichthyosaurus 90kg, <https://www.tiere-online.de/sonstige-tiere/dinosaurier/ichthyosaurus/>
- Ichthyostega 80kg, <https://dinosaurierwelt.com/ichthyostega/>
- Kaffernbüffel 350-900kg, <https://de.wikipedia.org/wiki/Kaffernb%C3%BCffel>
- Känguru 2-90kg ,<https://de.wikipedia.org/wiki/K%C3%A4ngurus>
- Kaninchen je nach Art, ganz grob 1kg
- Klippschliefer 2-5kg, <https://de.wikipedia.org/wiki/Klippschliefer>
- Koboldmaki 0,1kg, <https://de.wikipedia.org/wiki/Koboldmakis>
- Krokodil 100-1000kg, <https://de.wikipedia.org/wiki/Krokodile>
- Landschildkröte je nach Art, grob 50kg



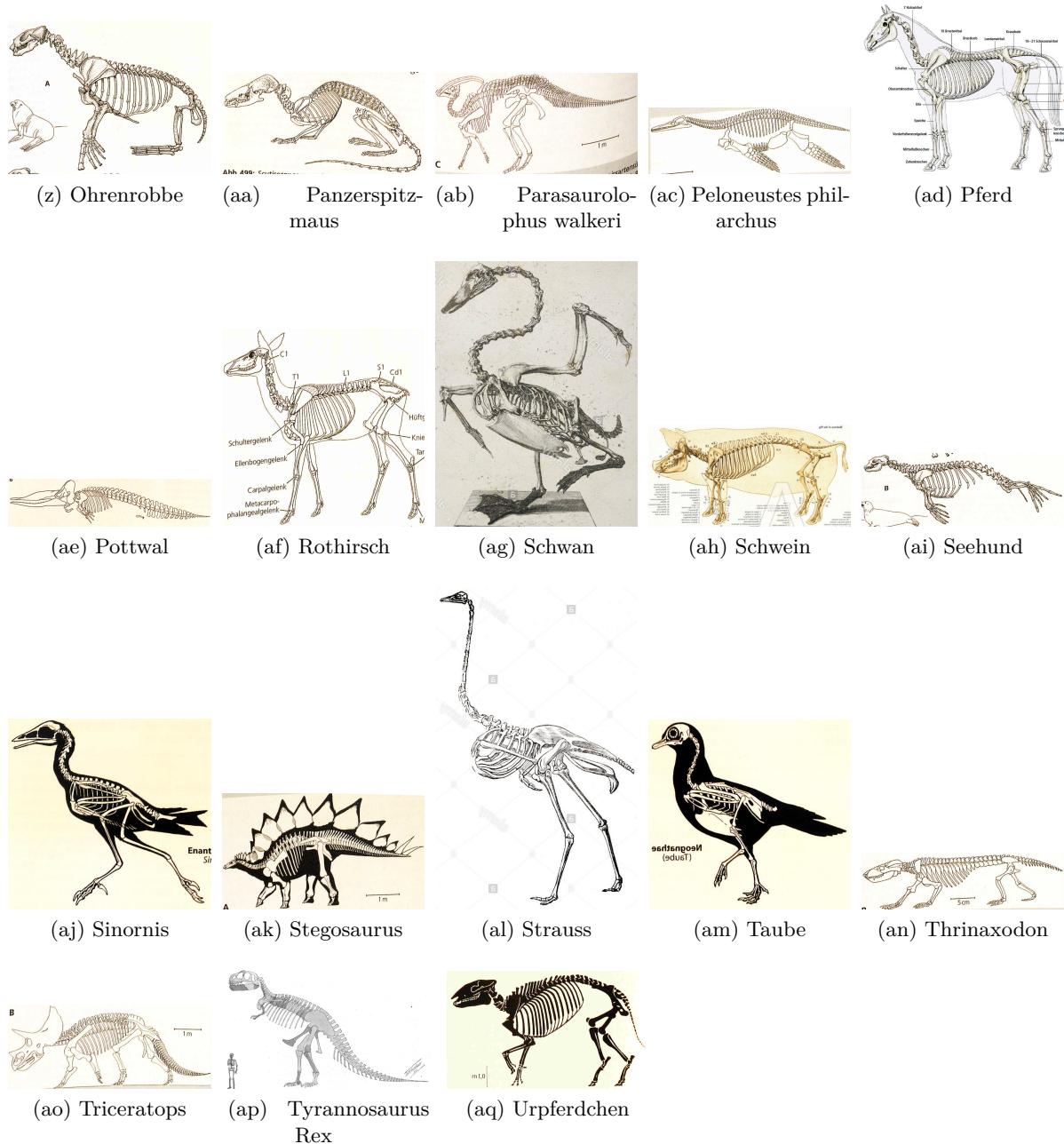
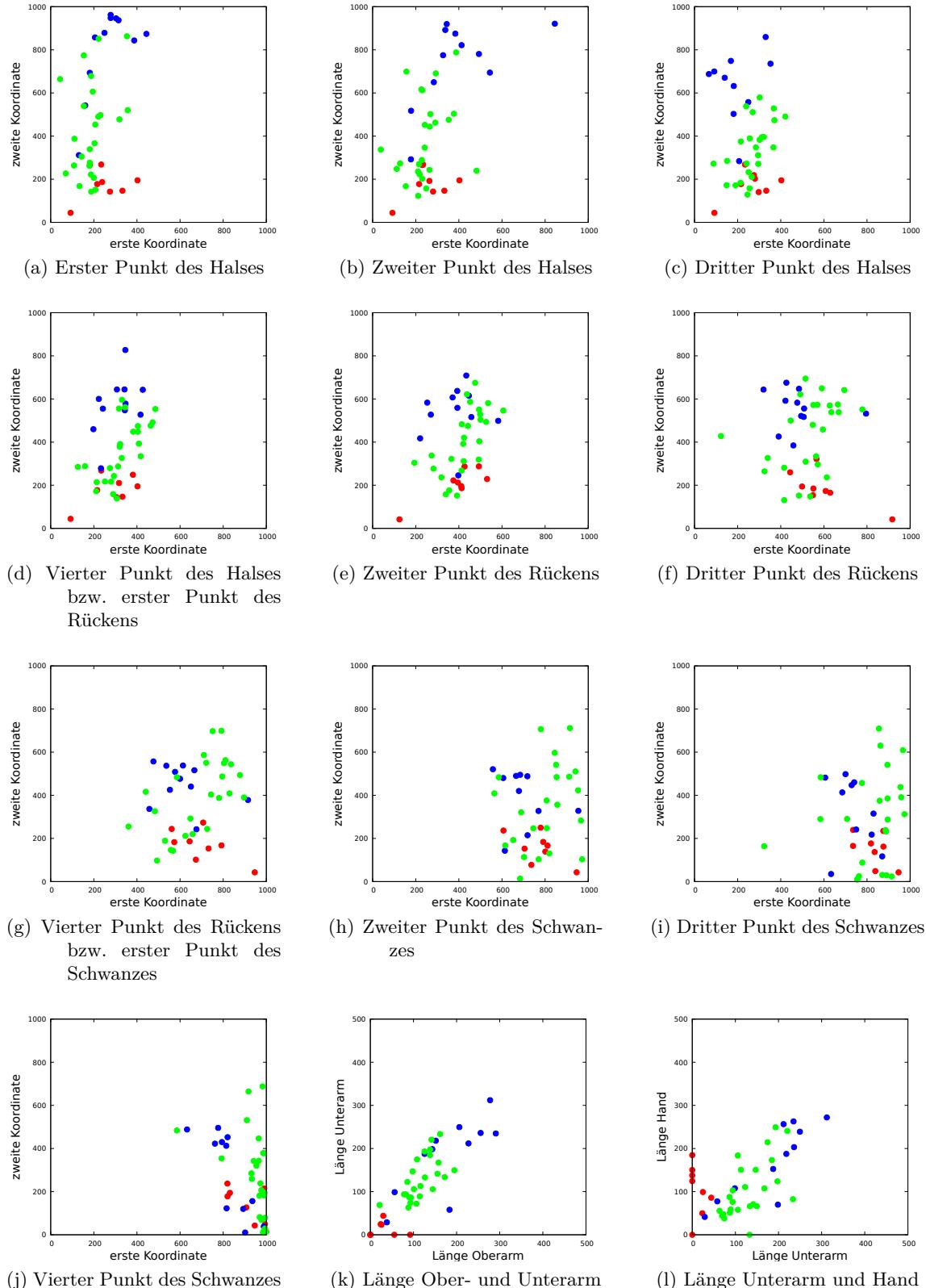


Abbildung A.1.: Alle Bilder, die als Eingabe für die PCA verwendet wurden.

- Ohrenrobbe 25-500kg, <https://de.wikipedia.org/wiki/Ohrenrobben>
- Panzerspitzmaus 100g ,<https://de.wikipedia.org/wiki/Panzerspitzmaus>
- Parasaurolophus walkeri 4-5 Tonnen, http://tierdoku.com/index.php?title=Parasaurolophus_walker
- Peloneustes philarchus 100kg, <https://de.wikipedia.org/wiki/Peloneustes>
- Pottwal bis 50 Tonnen, <https://de.wikipedia.org/wiki/Pottwal>
- Rothirsch 80-350kg, <https://de.wikipedia.org/wiki/Rothirsch>
- Schlange bis 100kg bei Riesenschlangen, <https://de.wikipedia.org/wiki/Schlangen>
- Schwan 14kg, <https://de.wikipedia.org/wiki/Schw%C3%A4ne>
- Schwein 100kg, <https://de.wikipedia.org/wiki/Hausschwein>
- Seehund 100-150kg, <https://de.wikipedia.org/wiki/Seehund>
- Sinornis 20g, http://dinodata.de/animals/birds/pages_s/sinornis.php
- Stegosaurus 4,5 Tonnen, <https://de.wikipedia.org/wiki/Stegosaurus>
- Taube je nach Art, grob 1-2kg
- Thrinaxodon Reptil „ein paar Pfund“, <https://www.thoughtco.com/thrinaxodon-1091887>
- Triceratops 6-12 Tonnen, <https://de.wikipedia.org/wiki/Triceratops>
- Tyrannosaurus 9 Tonnen, <https://de.wikipedia.org/wiki/Tyrannosaurus>
- Urpferdchen (Propalaeotherium) 30kg, <https://de.wikipedia.org/wiki/Propalaeotherium>



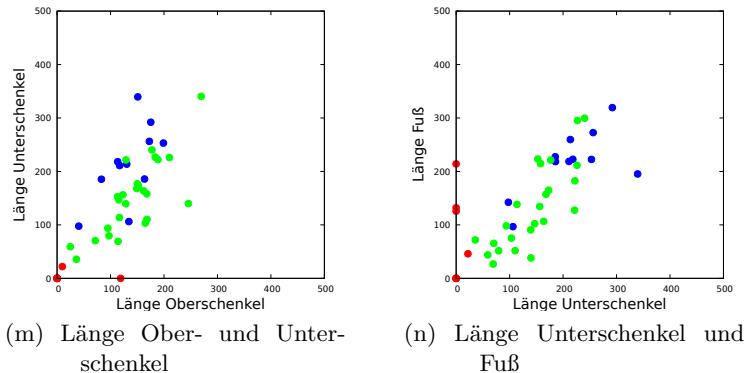


Abbildung A.2.: Erhobene Daten: Punkte der Bézierkurven der Wirbelsäule und Längen der Extremitäten. Bei den Extremitäten ist jeweils gegeneinander abgetragen Ober- und Unterarm, Ober- und Unterschenkel, Unterarm und Hand, Unterschenkel und Fuß. Markiert ist jeweils ob die Datenpunkte 0 (rot), 2 (blau) oder 4 (grün) Beine haben.

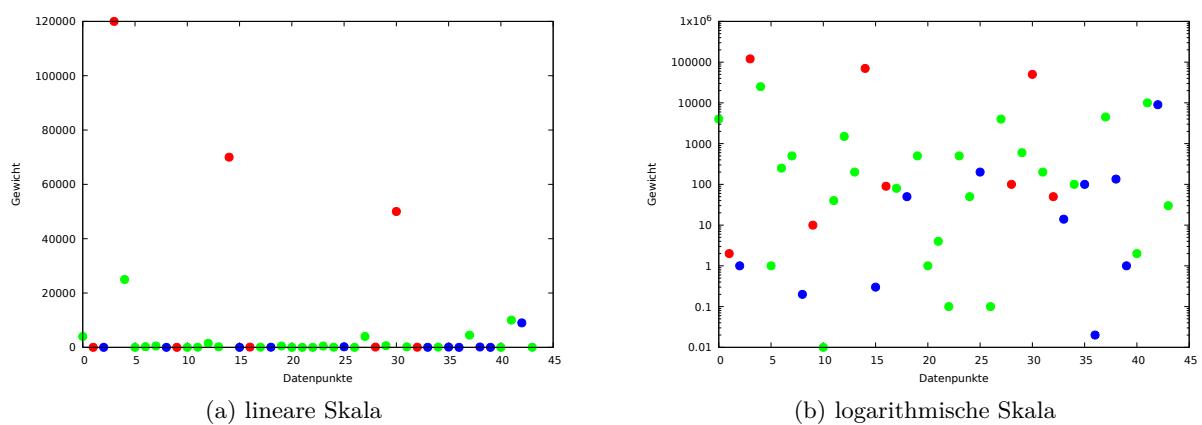
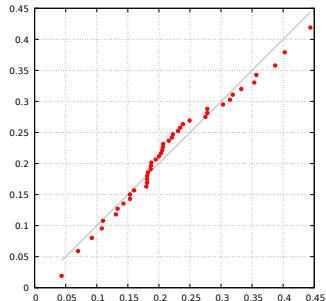
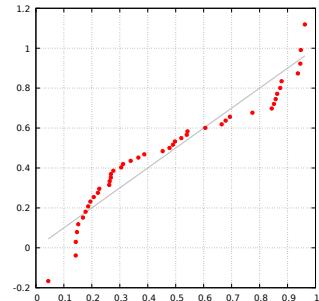


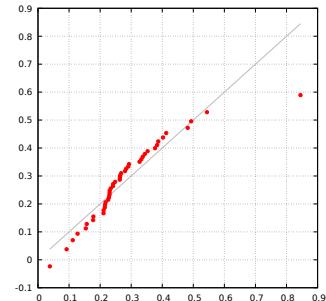
Abbildung A.3.: Erhobene Daten: Gewicht. Markiert ist jeweils ob die Datenpunkte 0 (rot), 2 (blau) oder 4 (grün) Beine haben.



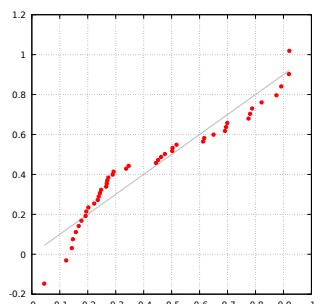
(a) Erster Punkt des Halses, x-Koordinate



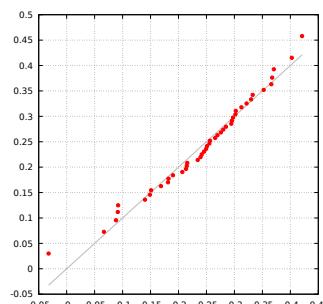
(b) Erster Punkt des Halses, y-Koordinate



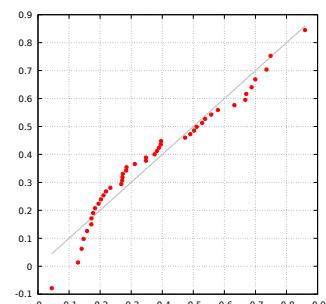
(c) Zweiter Punkt des Halses, x-Koordinate



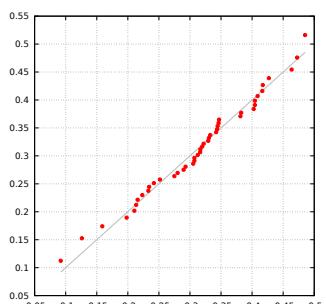
(d) Zweiter Punkt des Halses, y-Koordinate



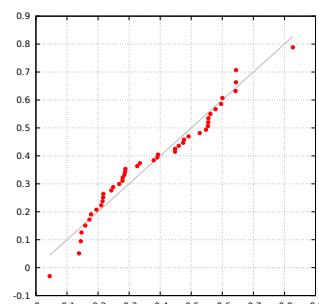
(e) Dritter Punkt des Halses, x-Koordinate



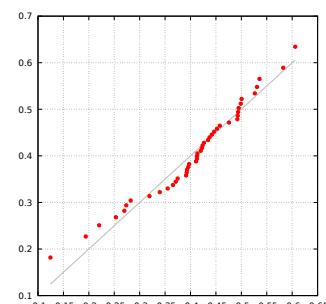
(f) Dritter Punkt des Halses, y-Koordinate



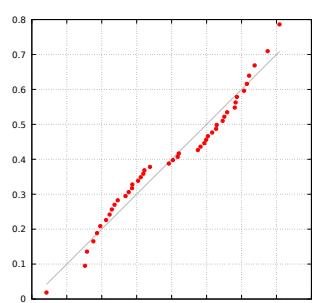
(g) Erster Punkt des Rückens, x-Koordinate



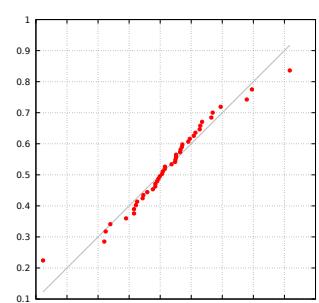
(h) Erster Punkt des Rückens, y-Koordinate



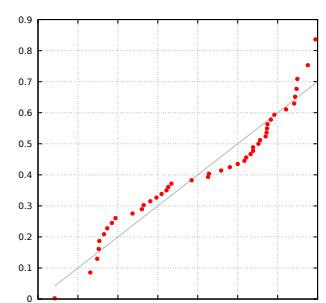
(i) Zweiter Punkt des Rückens, x-Koordinate



(j) Zweiter Punkt des Rückens, y-Koordinate



(k) Dritter Punkt des Rückens, x-Koordinate



(l) Dritter Punkt des Rückens, y-Koordinate

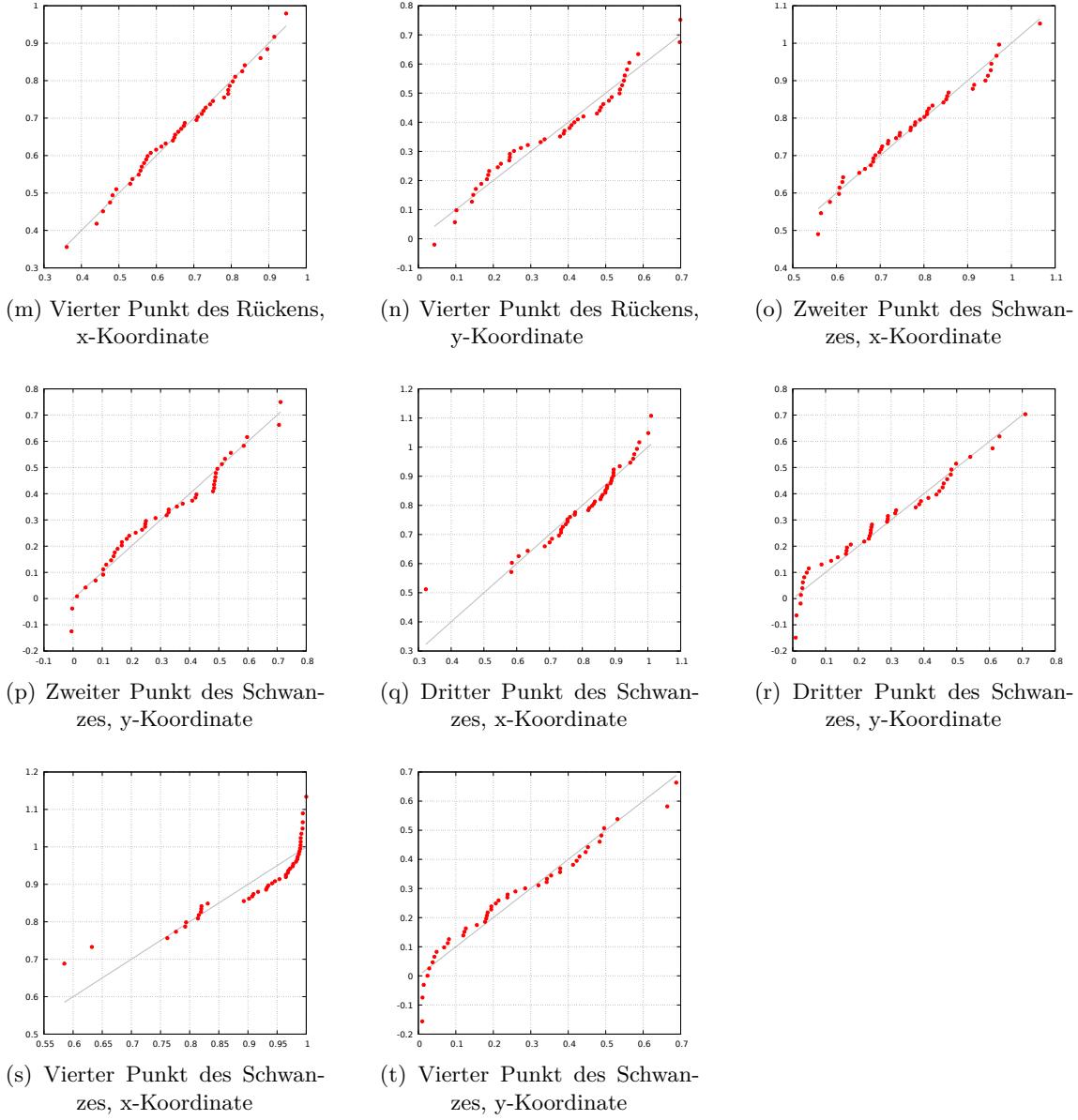


Abbildung A.4.: Quantil-Quantil-Diagramme für alle Dimensionen der Wirbelsäule

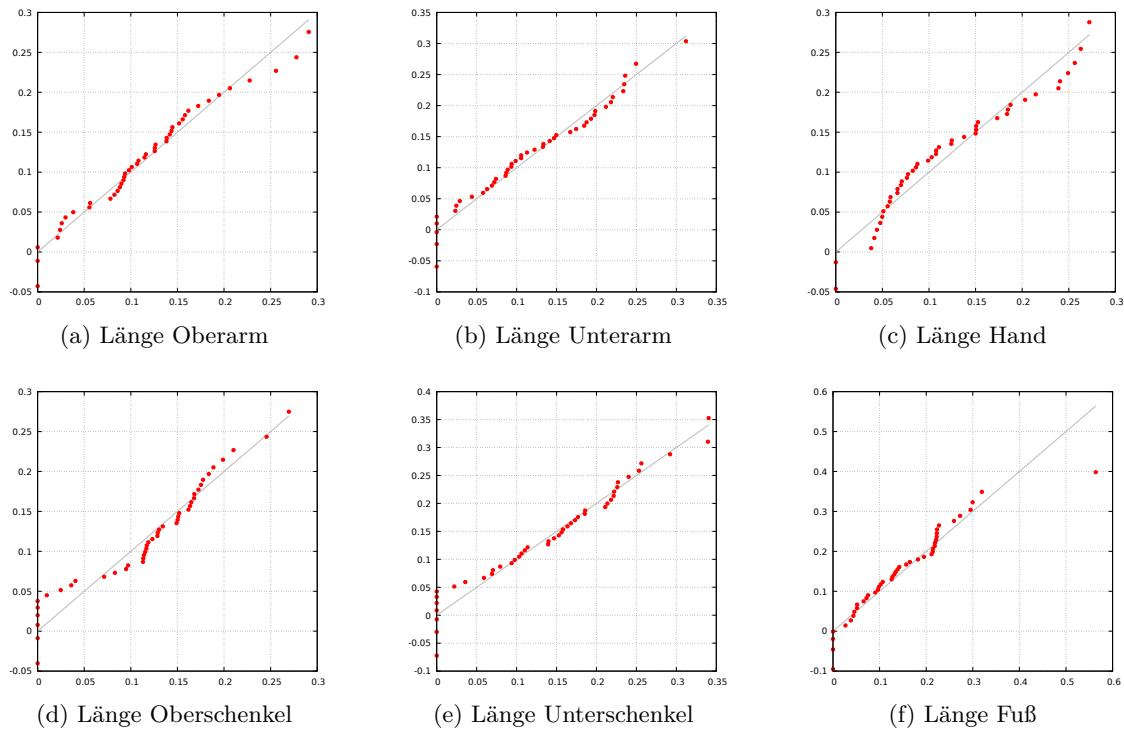
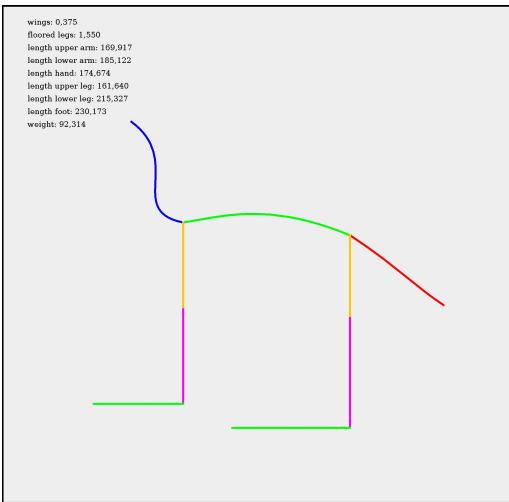
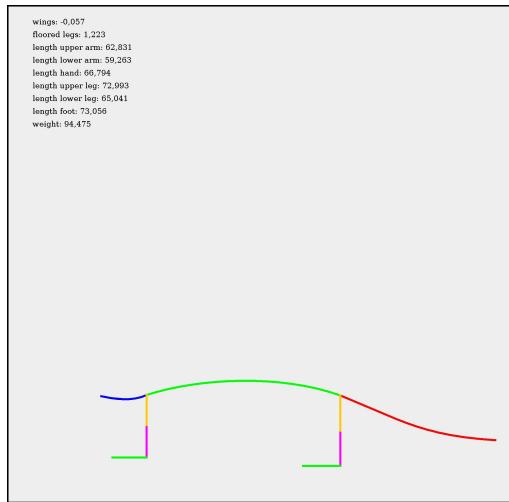


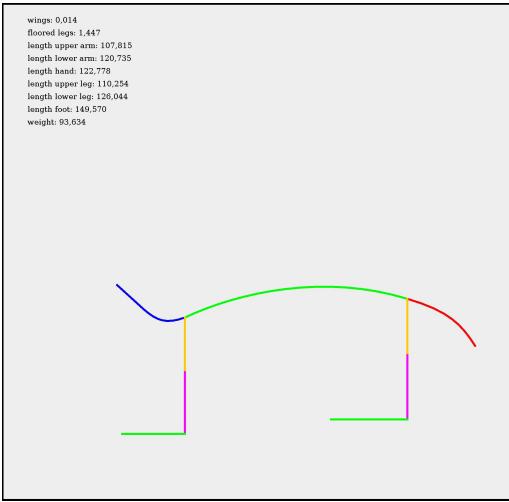
Abbildung A.5.: Quantil-Quantil-Diagramme für die Dimensionen der Eingabedaten, die zusätzlich zur Wirbelsäule erhoben wurden. Nicht dargestellt ist das binäre Attribut *Flügel* und die Anzahl der Beine mit Bodenkontakt.



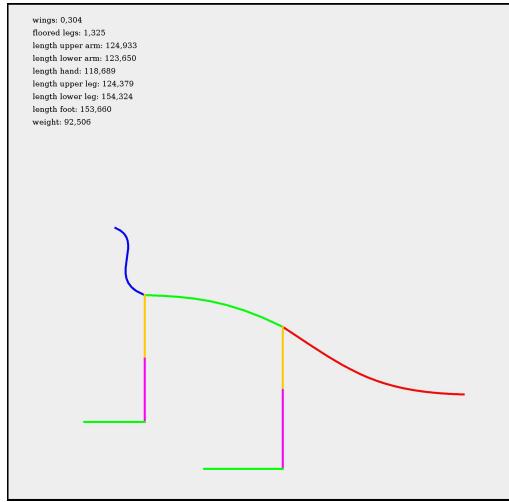
(a) 1-, Flügel 0,38, Beine 1,6, Gewicht 92kg



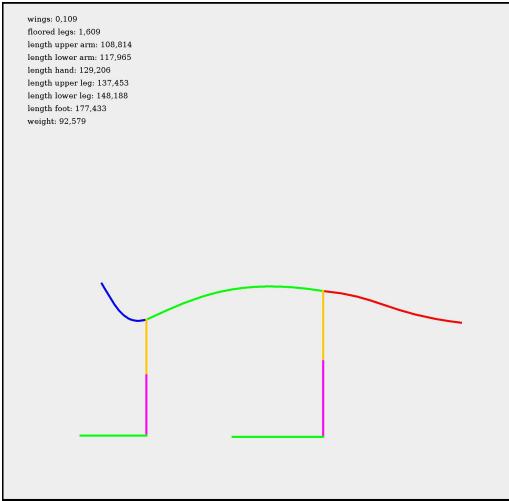
(b) 1+, Flügel -0,057, Beine 1,2, Gewicht 94kg



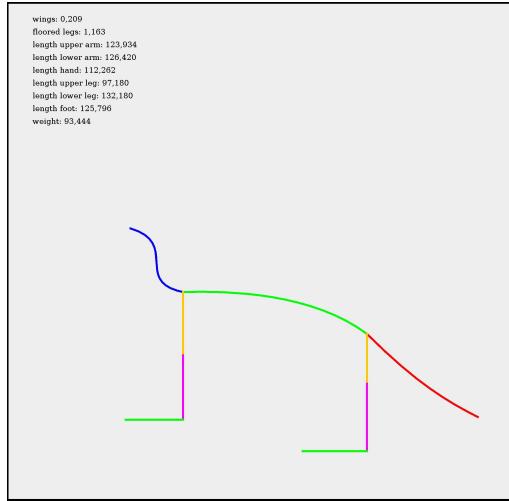
(c) 2-, Flügel 0,014, Beine 1,4, Gewicht 94kg



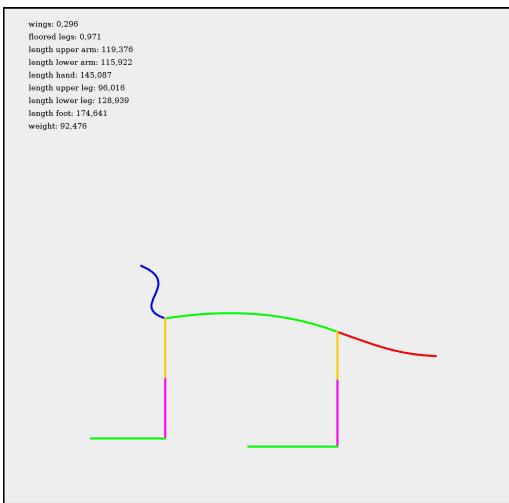
(d) 2+, Flügel 0,3, Beine 1,3, Gewicht 93kg



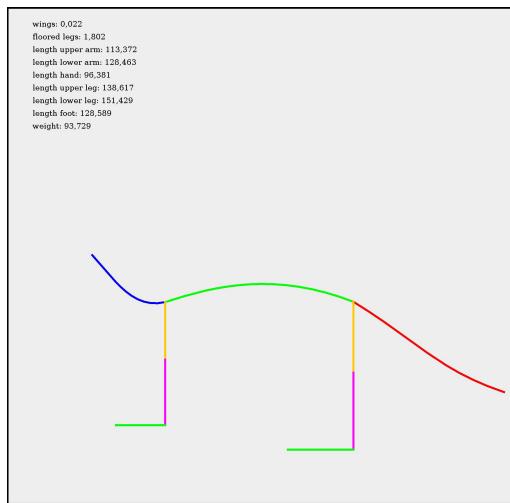
(e) 3-, Flügel 0,11, Beine 1,6, Gewicht 93kg



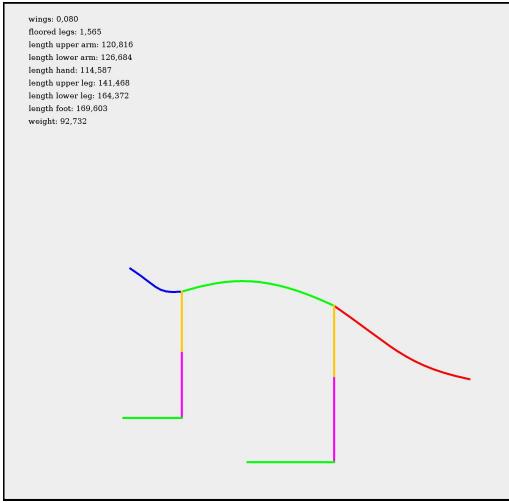
(f) 3+, Flügel 0,21, Beine 1,2, Gewicht 93kg



(g) 4-, Flügel 0,3, Beine 1, Gewicht 92kg



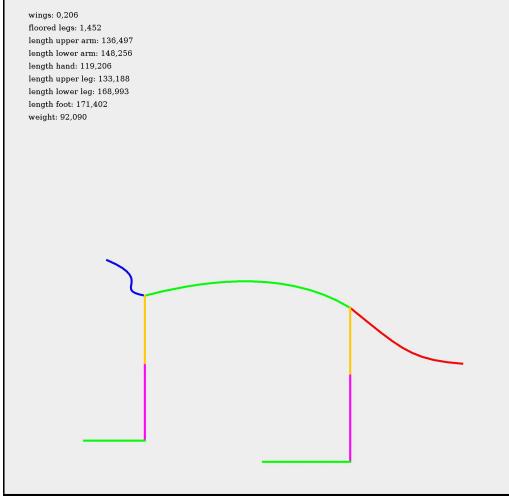
(h) 4+, Flügel 0,022, Beine 1,8, Gewicht 94kg



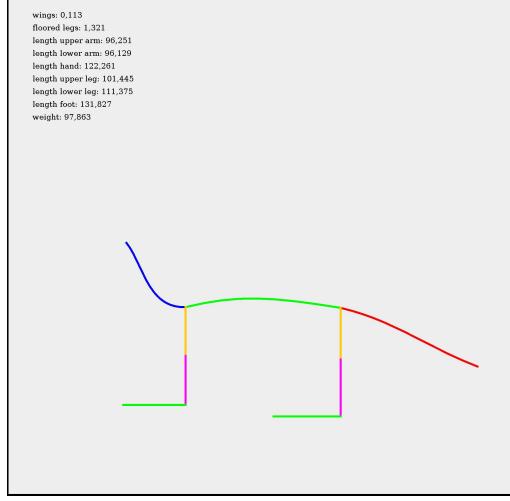
(i) 5-, Flügel 0,08, Beine 1,6, Gewicht 93kg



(j) 5+, Flügel 0,24, Beine 1,2, Gewicht 93kg



(k) 6-, Flügel 0,21, Beine 1,5, Gewicht 92kg



(l) 6+, Flügel 0,11, Beine 1,3, Gewicht 98kg

Abbildung A.6.: Datenpunkte im PCA-Koordinatensystem. Eine Koordinate nimmt den Wert der positiven (+) bzw. negativen (-) Standardabweichung in die entsprechende Richtung an, alle anderen sind null. Von oben nach unten sind die Ergebnisse für den größten (1) bis zum sechstgrößten (6) Eigenwert dargestellt. *Beine* steht für *Beine mit Bodenkontakt*.

Erklärung

Ich versichere, dass ich die Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht und die Satzung des KIT zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet habe. Die Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt und von dieser als Teil einer Prüfungsleistung angenommen.

Karlsruhe, den 4. Juni 2020

(Nina Zimbel)