

# Gruppennummer 16

Andreas Cremer  
Hanna Huber  
Lena Trautmann

May 23, 2016

## 1. Kovarianzmatrix

- (a) `ourCov.m` erwartet eine  $d \times n$  Matrix und gibt die dazugehörige Kovarianzmatrix zurück.
- (b) Hier werden die Kovarianzmatrizen für `daten.mat` berechnet. In  $C_{11}$  steht die Varianz in der ersten Dimension. In  $C_{22}$  steht die Varianz in der zweiten Dimension. In  $C_{12}$  und  $C_{21}$  steht die Kovarianz.  
data1 hat eine hohe Varianz in der ersten und eine geringe Varianz in der zweiten Dimension. Die Kovarianz ist gering, die Datenpunkte bilden ein schmales Band parallel zur x-Achse.  
data2 hat eine geringe Varianz in der ersten und eine hohe Varianz in der zweiten Dimension und ebenfalls eine geringe Kovarianz. Die Datenpunkte bilden ein schmales Band parallel zur y-Achse.  
data3 hat eine sehr hohe und eine deutlich niedrigere Varianz sowie eine hohe Kovarianz. Dies führt zu einem leicht ansteigenden Band. data4 hat hohe nahe beieinander liegende Varianzen und eine Kovarianz nahe beim Nullpunkt. Dies führt zu einer Punktwolke ohne erkennbare Ordnung.

## 2. PCA

`pca.m` berechnet die PCA indem mithilfe von der Matlabfunktion `eig` die Eigenwerte und -vektoren abgefragt werden. Diese Funktion ordnet beides nach aufsteigenden Eigenwerten, daher wird danach noch die Reihenfolge umgekehrt.

- (a) Plot der Ergebnisse für die Daten aus `daten.mat` mit `plot2DPCA.m`.
- (b) Der erste Eigenvektor gibt die Richtung der höchsten Varianz an. Weitere Eigenvektoren stehen jeweils orthogonal auf alle schon vorhandenen Eigenvektoren und geben die Richtung der höchsten verbleibenden Varianz an. Im Plot sind die Eigenvektoren durch blaue Striche durch den Mittelwert gekennzeichnet.
- (c) Die Eigenwerte zu den Eigenvektoren geben die Varianz in Richtung des jeweiligen Eigenvektors an. Im Plot sind sie durch die Länge der Eigenvektormarkierungen dargestellt. Sie ergeben aufaddiert die Gesamtvarianz.
- (d) In die Berechnung von Varianz und Kovarianz fließt hier der Abstand der Datenpunkte vom Nullpunkt des verwendeten Koordinatensystems mit ein. Somit kann man keine sinnvollen Schlussfolgerungen mehr ziehen. Durch den Mittelwertsabzug wird die Kovarianzmatrix invariant gegen Translation.

## 3. Unterraum-Projektion

- (a) In `project.m` werden data3 und die Eigenvektoren übernommen und die Daten mithilfe von  $b = E^T * (x - m)$  in das Koordinatensystem des Eigenraums übertragen. Hierdurch können die Daten auf den Hauptvektor projiziert werden, indem weitere Dimensionen einfach weggeschnitten werden. Die Daten sind damit nur noch eindimensional.  
Anschließend werden die Daten im Eigenraum mit Nullen auf die ursprüngliche Dimension aufgefüllt. Diese neuen Daten werden in der obigen Gleichung als  $b$  eingesetzt und es wird nach  $x$  gelöst. So werden die auf den Hauptvektor projizierten Daten wieder im ursprünglichen Koordinatensystem dargestellt.  
Wie man im Plot sehen kann, liegen die Datenpunkte nach Projektion und Rekonstruktion alle auf einer Linie, da sie auf eine Dimension reduziert wurden. Der durchschnittliche Fehler liegt bei 0.7257.
- (b) Der im Plot sichtbare Unterschied zum vorigen Punkt ist, dass die Datenpunkte nach Projektion und Rekonstruktion viel näher zusammen liegen. Der durchschnittliche Fehler ist mit 8.9097 deutlich höher. Um den Fehler gering zu halten, verwendet man die Eigenvektoren mit den höchsten Eigenwerten. Hierbei gibt es einen Trade-off zwischen Genauigkeit und Anzahl der verwendeten Eigenvektoren.

## 4. Untersuchungen in 3D

- (a)
- (b) Nach Projektion auf den Unterraum, der durch die ersten beiden Eigenvektoren aufgespannt wird, haben die Daten Dimension zwei. Die verlorene Information ist die des dritten Eigenvektors. Die Daten liegen jetzt in einer Ebene.

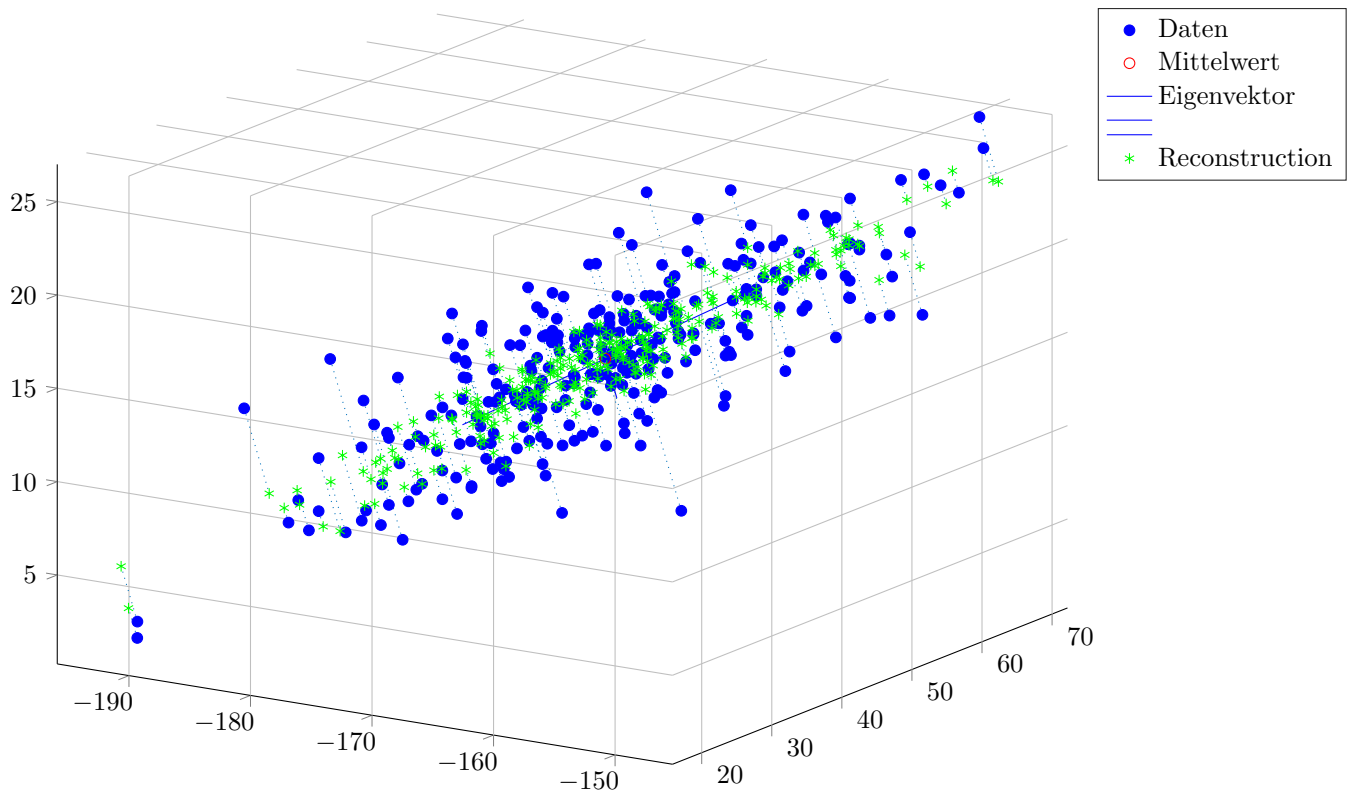


Figure 1: Projection of 3D data on first to principal components and its reconstruction

## 5. Shape Modell

- (a) generateShape.m berechnet eine neue Shape anhand der mit  $b$  gewichteten Eigenvektoren und dem Mittelwert aller Shapes.
- (b) Die ersten 13 Eigenwerte sind größer als 1, alle weiteren Eigenwerte gehen gegen 0 (Werte der Größenordnung  $10^{-13}$  und kleiner). Daher tragen eigentlich nur die ersten 13 Modes zur Gesamtvarianz bei und werden im Folgenden genauer betrachtet (siehe Abb. 2).  
Der erste Modus beinhaltet mehr als 50% der Gesamtvarianz und beeinflusst hauptsächlich die Größe der Struktur. Der zweite Modus beinhaltet ein weiteres viertel der Gesamtvarianz und beeinflusst am meisten die Länge (negatives  $b$  für diesen Modus) bzw. Breite (positives  $b$  für diesen Modus) der Knochenstruktur. Bereits der dritte Modus deckt nur noch etwas mehr als 10% der Gesamtvarianz ab und ist für leichte Änderungen der exakten Knochenstruktur verantwortlich. Der vierte und fünfte Modus (mit zusammen knapp 6% der Gesamtvarianz) bestimmen wie weit die Epiphysen des Knochens im Gegensatz zur Diaphyse herausstehen. Alle Modi ab dem sechsten decken bis zu gut 1% der Gesamtvarianz ab und führen zu spezifischen Ausbuchtungen.
- (c) Mit Hilfe eines zufällig generierten  $b$ -Vektors wurde eine neue Knochenstruktur berechnet. Die Anzahl der Einträge in  $b$ , und damit die Anzahl der verwendeten Eigenvektoren, wurde variiert zwischen eins, drei, sechs, zehn, 20, 100 und allen. In Abbildung 3 sieht man die Unterschiede in der erzeugten Struktur. Wird nur der erste Eigenvektor verwendet, ist die Abweichung vom gesamten Modell am größten. Werden die ersten zehn Eigenvektoren verwendet, sind nur noch geringe Abweichungen erkennbar, wenn man entsprechende Teile der Graphik vergrößert betrachtet. Da die Eigenwerte ab dem 14. gegen Null gehen, ist es nicht überraschend, dass keine Unterschiede zwischen dem Modell mit 20 bzw. 100 Eigenvektoren und dem kompletten Modell mit allen Eigenvektoren erkennbar sind.  
In Abbildung 4 ist die Anzahl der verwendeten Eigenvektoren anhand der mindestens abgedeckten Gesamtvarianz bestimmt. Da die ersten beiden Eigenvektoren eine Gesamtvarianz von ungefähr 75% abdecken, werden sowohl für 80% als auch für 90% drei Eigenvektoren verwendet. Für 95% sind zwei weitere Eigenvektoren notwendig. Für ein vollständiges Modell werden die ersten 13 Eigenvektoren benötigt, alle weiteren tragen derart geringfügig zur Gesamtvarianz bei, dass sie vernachlässigt werden können. Der Fehler beläuft sich hierbei auf  $1.4158 \times 10^{-15}$ .

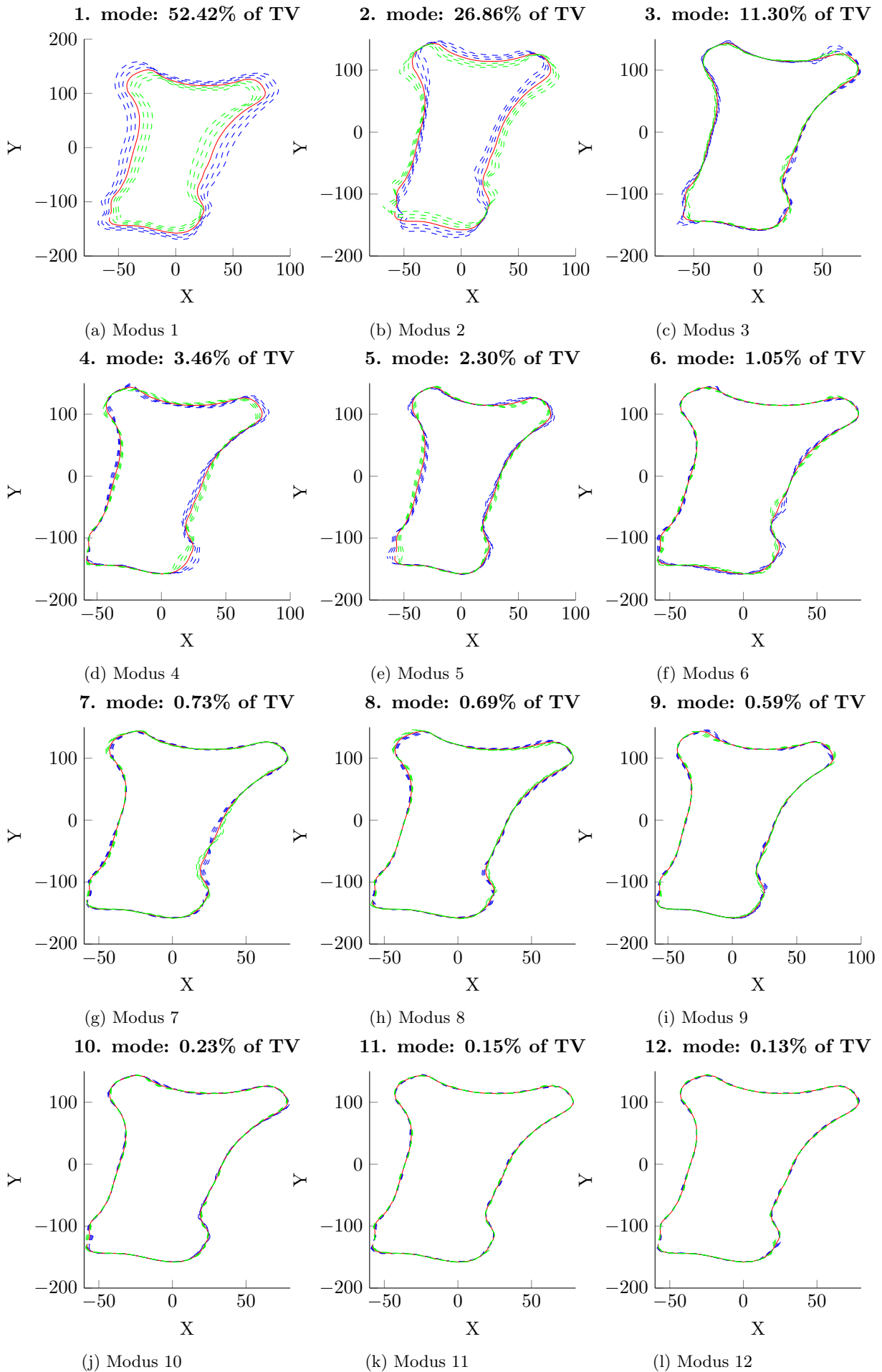


Figure 2: Die ersten dreizehn Modi

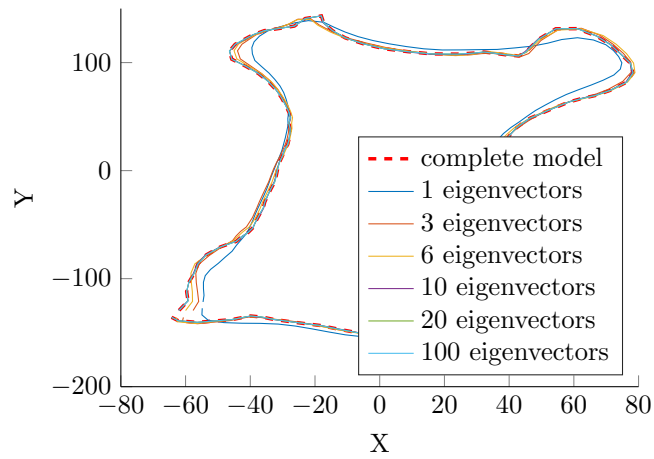


Figure 3: Neue Shape bei variierter Anzahl an verwendeten Eigenvektoren

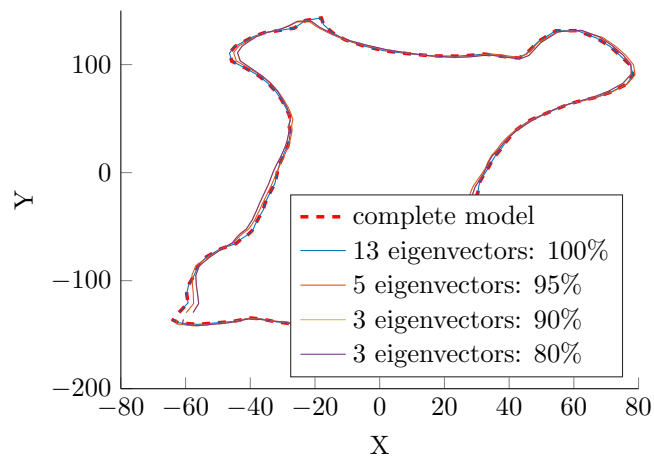


Figure 4: Neue Shape bei variiertem Prozentsatz an abgedeckter Gesamtvarianz