

# User Guide für das Erstellen von Druckstatistiken

In Bezug zu Abdrücken von Klauen auf Tekscan-Druck-Sensoren

Kai Hainke

29. Juni 2017

# KAPITEL 1

---

## VORAUSSETZUNGEN

### 1.1 Tools

In diesem Guide werden folgende Tools benutzt:

1. Python 2.7.13 mit folgenden Paketen (zu installieren mit zBsp. `pip install pycclipper`):
  - a) `numpy`
  - b) `pyclipper`
  - c) `sklearn`
2. Maya 2014
3. ProKlaue Plugin 0.3.4
4. XROMM Plugin
5. R für Plots

### 1.2 Verzeichnisstruktur

Das Basisverzeichnis für diesen Guide ist

»Documents\ProKlaue\testdaten\druck«.

Darunter sollte pro Klaue ein Ordner der Form

»Klaue <Nr.> (K<Nr.>T)« liegen.

In diesem Ordner wiederum sollten alle benötigten Dateien zur Klaue liegen und auch dorthin geschrieben werden.

### 1.3 Ausgangsdateien

Zusätzlich müssen folgende Dateien vorliegen (jeweils pro Klaue, im Ordner der Klaue):

1. Jeweils für links und rechts 1 `.obj`-Datei
2. 1 Ordner mit rigid-body-matrix-Dateien für das XROMM Plugin, dieser sollte jeweils für links und rechts mind. 1 `.csv`-Datei enthalten
3. 1 Tekscan `.csv`-Datei mit Daten von den Drucksensoren
4. 2 Definitionen von Messzonen für die Statistik, siehe 2
5. das `pressureStatistics.py` - Script zusammen mit dem `pressureStatisticsUtilFunctions.py` Script

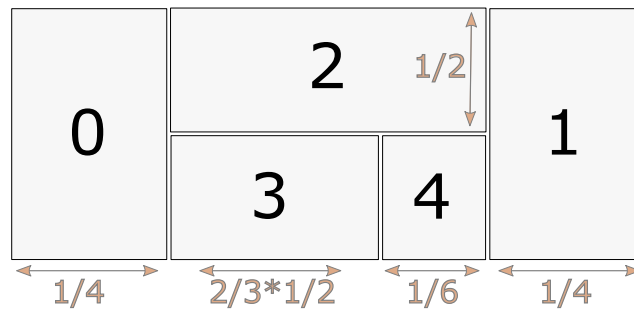


Abbildung 1.1: Standard Messzonen (rechte Klaue)

## 1.4 Variablen

Der Nutzer muss außerdem den Wert für folgende Variablen festlegen:

1. Einsinktiefe a.k.a. der Höhen-Wert für den der Abdruck des Oberflächen-netzes der Klaue bestimmt wird
2. Zonenraster, Definition der Zonen für die Statistik, diese sollten in relativen Koordinaten definiert werden ((0,0) ist der untere linke Punkt der Klaue in Ausrichtung vorne-hinten ( $\leftarrow x \rightarrow$ ), Links-Rechts ( $\leftarrow y \rightarrow$ ), (1,1) der obere rechte). In den Standard-Dateien (Documents\ProKlaue\testdaten\druck\segments\_for\_measurements\_<side>.csv) sind die Zonen definiert wie in Abbildung 1.1. Diese Dateien sollten auch im Basis-Verzeichnis liegen und dementsprechend benannt sein.

## KAPITEL 2

### WORKFLOW

Zunächst wird aus den Dateien für das XROMM-Plugin das Modell der Klaue in Maya eingeladen. Danach wird hierfür eine Liste von Dreieckssegmenten des Oberflächenmodells errechnet. Mittels eines Thresholds und Clipping wird durch Python ein Außenriss errechnet. Außerdem wird eine vorläufige Transformation zur Ausrichtung an den Druckdaten errechnet. Mittels R lässt sich dies darstellen und ggf. anpassen. Nach einer erneuten Berechnung durch Python lässt sich dann auch die Statistik zu den Zonen verwerten.

### 2.1 Erstellen des Außenriss

Zunächst ruft man das Import-XROMM-Plugin in Maya auf. Dann importiert man durch Klick auf den Button impObjs die entsprechenden `.obj`-Dateien der Klauen (in diesem Fall erst links, dann rechts, den Dialog zum Skalieren mit Yes bestätigen). Dann wählt man die entsprechenden `.csv`-Dateien mit den rigid-body-Koordinaten, hier wieder erst links dann rechts. Dann wählt man im Outliner die eben importierten Objekte aus (links, dann rechts). Dann die Option Rigid Body Matrix im Data type Reiter anklicken und den Haken bei Coloumn Headers in Row entfernen. Zuletzt auf import klicken. Siehe Abb. 2.2.

Jetzt mithilfe des app Plugins (ProKlaue) eine achsenparallele Ebene erstellen für eine der beiden Klauen. Standardeinstellungen belassen und auf create klicken.

Nun jeweils eine Klaue und die Ebene auswählen und den folgenden Python-Befehl benutzen:

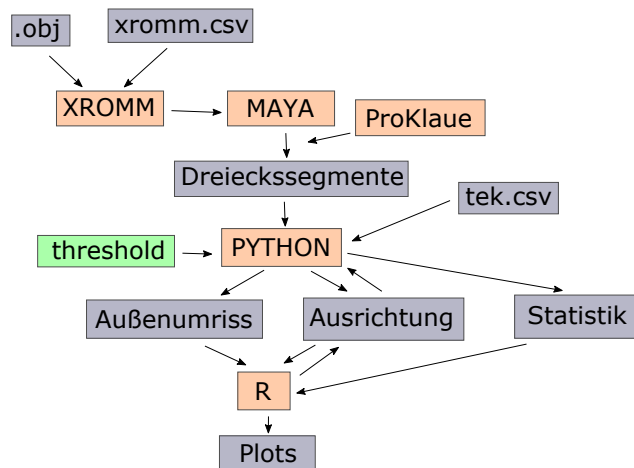


Abbildung 2.1: allgemeiner Workflow

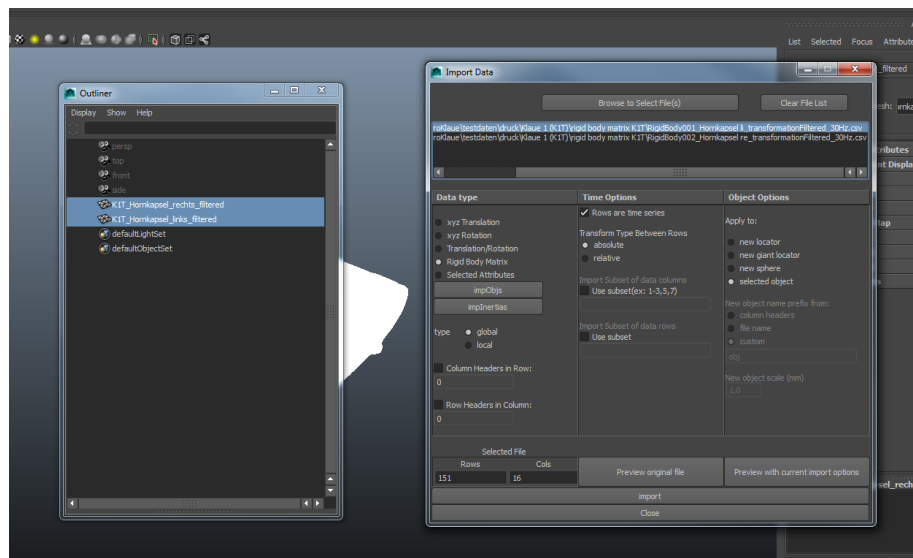


Abbildung 2.2: XROMM Optionen

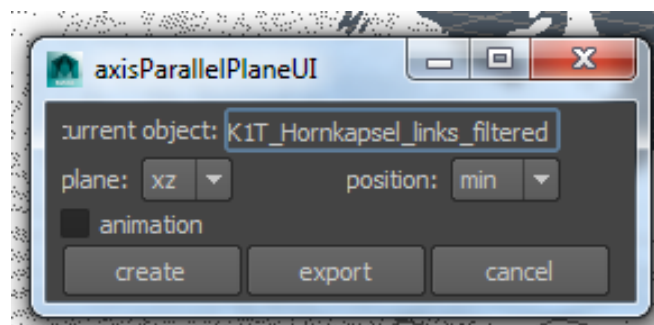


Abbildung 2.3: ProKlaue app

---

```
cmds.frontVertices(tsf="<>/Documents/ProKlaue/testdaten/druck/<Klaue-Verzeichnis>/tsf_<left oder right>.csv")
```

---

Stellen in Klammern <> sind zu ersetzen. Auch darauf achten, Pfade mit / und nicht mit \ zu schreiben.

Nun findet sich im Ordner der Klaue 2 neue Dateien `tsf_left.csv` und `tsf_right.csv`. Anschließend navigiert man zum `pressureStatistics.py` Script und führt es mit folgenden Parametern aus:

---

```
ProKlaue\scripts>python pressureStatistics.py -s <step> -t <threshold> -b "<klaunennamen>" -g "<bodentyp>" -d "<base_directory>"
```

---

th ist der Wert der Einsinktiefen (hier in cm), siehe 1. Der Wert für -b (bone) muss dem Namen des Klauen-Verzeichnisses entsprechen. Und der Name des

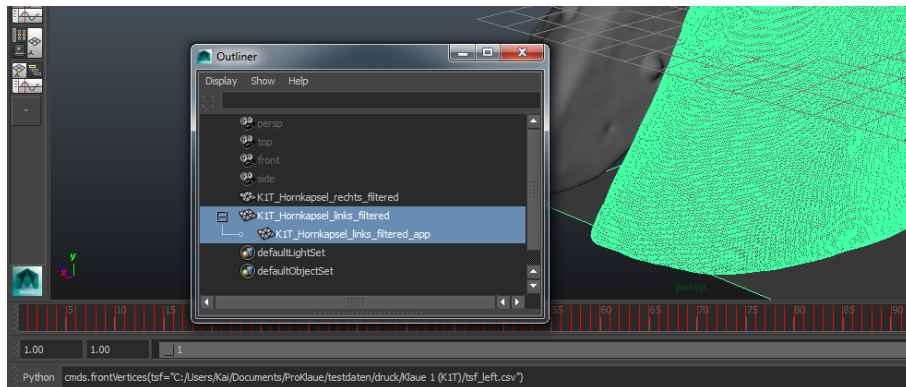


Abbildung 2.4: ProKlaue frontVertices

Boden (-g) sollte kleingeschrieben sein, die Beschriftung für die TEK-Dateien sollte jedoch den Namen großgeschrieben beinhalten. Wieder darauf achten, Pfade mit / und nicht mit \ zu schreiben.

Zunächst beginnt man mit step=0.

(OPTIONAL: Dabei sollte man den PYTHONPATH so anpassen, dass er auch das script modul findet. Dies macht man am besten unter Windows, Systemsteuerung, System und Sicherheit, System, Erweiterte Systemeinstellungen, Reiter Erweitert, Umgebungsvariablen, Reiter Systemvariablen, Neu, dann:

Name der Variable: PYTHONPATH

Wert der Variable: <>\Documents\ProKlaue;<>\Documents\ProKlaue\scripts

Die Ordner so anpassen, dass das ProKlaue-Verzeichnis und das scripts Verzeichnis dort auftauchen, durch Semikolon getrennt. Bestätigen und Eingabeaufforderung öffnen.)

Dann das Skript ausführen, siehe auch Abb. 2.5:

---

```
C:\Users\Kai\Documents\ProKlaue\scripts>python pressureStatistics.py -s
0 -t 0.5 -b "Klaue 1 (KIT)" -g "gummi" -d "C:/Users/Kai/Documents/
ProKlaue/testdaten/druck"
```

---

Danach findet man 2 Segment-Dateien für den Umriss im Klauen-Verzeichnis, diesen Schritt muss man nur einmal pro Klaue machen (Heißt man könnte auch hier weiter machen, falls man den 2ten Boden bearbeitet). Die nächsten Schritte müssen pro Boden ausgeführt werden.

Dann definiert man den Bodenparameter im Skript und setzt den Schritt-parameter auf 1:

---

```
C:\Users\Kai\Documents\ProKlaue\scripts>python pressureStatistics.py -s
1 -t 0.5 -b "Klaue 1 (KIT)" -g "gummi" -d "C:/Users/Kai/Documents/
ProKlaue/testdaten/druck"
```

---

Ausführen, man findet Dateien zur Transformationen und Druckdaten im Klauenverzeichnis.

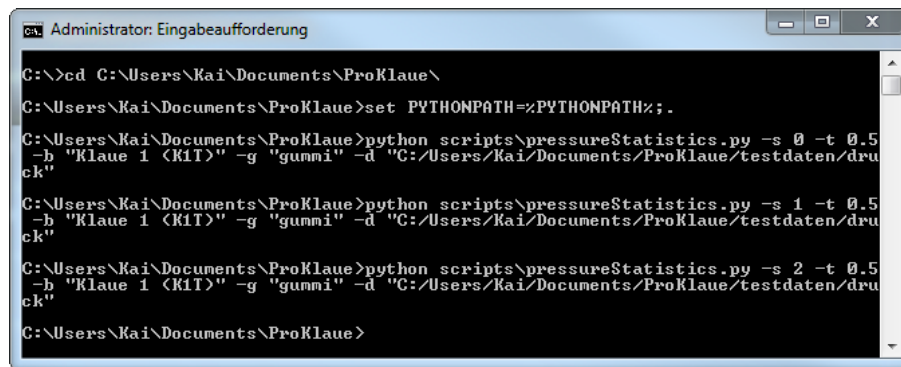


Abbildung 2.5: Ausführen des Python Skripts

Bibliothek "Dokumente"

Klaue 1 (KIT)

Anordnen nach:

Name	Änderungsdatum	Typ	Größe
rigid body matrix KIT	23.05.2017 12:23	Dateiordner	
pressure_data_gummi.csv	21.06.2017 14:42	Microsoft Excel-C...	40 KB
pressure_metadata_gummi.csv	21.06.2017 14:42	Microsoft Excel-C...	1 KB
transformation_imprint_gummi.csv	21.06.2017 14:42	Microsoft Excel-C...	1 KB
transformation_zones_left_gummi.csv	21.06.2017 14:42	Microsoft Excel-C...	1 KB
transformation_zones_right_gummi.csv	21.06.2017 14:42	Microsoft Excel-C...	1 KB
segments_imprint_left.csv	21.06.2017 14:38	Microsoft Excel-C...	48 KB
segments_imprint_right.csv	21.06.2017 14:38	Microsoft Excel-C...	52 KB
tsf_right.csv	21.06.2017 14:38	Microsoft Excel-C...	24,788 KB
tsf_left.csv	21.06.2017 13:46	Microsoft Excel-C...	25,702 KB
Klaue 1 (KIT)_Gummi.csv	26.04.2017 15:41	Microsoft Excel-C...	6 KB
Klaue 1 (KIT)_Gummi.jpg	26.04.2017 15:41	JPEG-Bild	56 KB
Klaue 1 (KIT)_Beton.csv	26.04.2017 15:31	Microsoft Excel-C...	6 KB
Klaue 1 (KIT)_Beton.jpg	26.04.2017 15:29	JPEG-Bild	51 KB
KIT_Hornkapsel rechts_filtered.obj	03.04.2017 13:10	OBJ-Datei	19,099 KB
KIT_Hornkapsel links_filtered.obj	03.04.2017 13:09	OBJ-Datei	19,811 KB

Abbildung 2.6: Klauenordner nach step=1

## 2.2 Anpassen

Nun kann man im R Skript (`scripts\druck.R`) mittels folgender Befehle die Transformation anpassen. Erst die Parameter im SETTINGS-Block anpassen, also Basisverzeichnis, Bodentyp (kleingeschrieben) und Klauen(verzeichnis)-Name angeben:

```

setwd("~/ProKlaue/testdaten/druck")
ground_type = "beton"
bone = "Klaue 3 (K3T)"

```

Und nach Ausführen des SETTINGS-Code-Blocks und des TRANSFORM FIT - Code-Blocks lassen sich auch entsprechende Teile des PLOTS-Code-Blocks ausführen der Fit wird angezeigt.

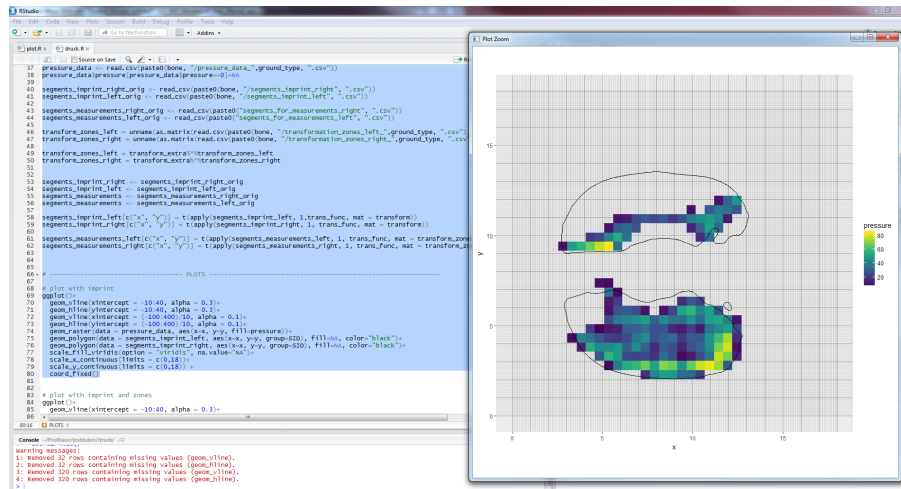


Abbildung 2.7: Anpassen der Transformation

---

```

rot = rotation_matrix(4, radian=F)
rot_pivot = matrix(c(7.4,9), nrow=2)
displace = matrix(c(0,0.3), nrow=2)

```

---

Hier wurde der Fit um 4° um den Punkt (7.4, 9) entgegen dem Uhrzeigersinn gedreht worden und dann um 0.3 Einheiten nach oben verschoben. Das Ergebnis zeigt Abb. 2.7.

Um den Fit zu speichern führt man den SAVE TRANSFORMED FIT - Code - Block aus, um einen gespeicherten Fit zu lesen, führt man den READ TRANSFORMATION - Code - Block aus.

## 2.3 Statistik und Plots

Nachdem man den Fit gespeichert hat (SAVE TRANSFORMED FIT-Block), kann man das Python Skript erneut ausführen, diesmal mit step=2:

---

```

C:\Users\Kai\Documents\ProKlaue\scripts>python pressureStatistics.py -s
2 -t 0.5 -b "Klaue 1 (K1T)" -g "gummi" -d "C:/Users/Kai/Documents/
ProKlaue/testdaten/druck"

```

---

Nun findet sich auch ein `statistics_<boden>.csv` im Klauenverzeichnis mit Daten für die Messzonen. Dies lässt sich nun auch wieder in R laden und darstellen. Dort sind u.a. für jede Zone die Fläche und der errechnete Druck, sowie das Verhältnis (sowohl zur Fläche, als auch zum Gesamtdruck der Seite, etc.) aufgeführt. Die Bezeichnungen der Spalten und eine Erklärung für die jeweiligen Daten:



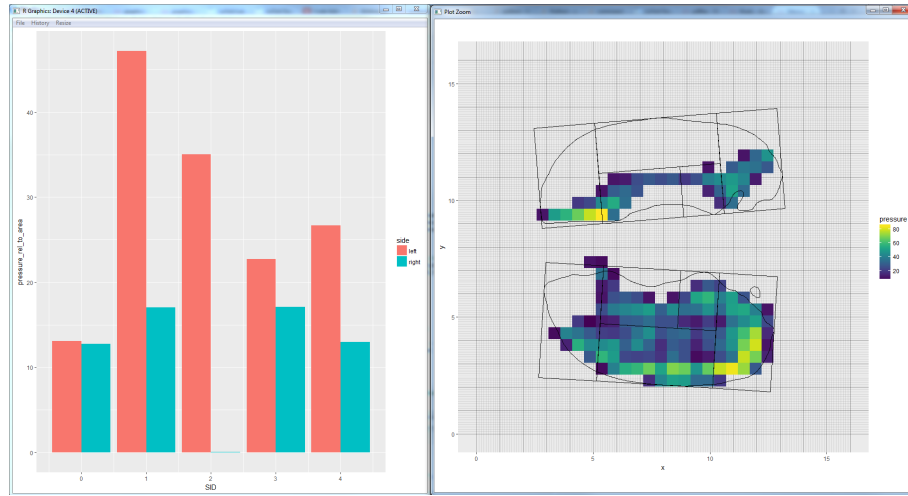


Abbildung 2.8: Plots

Bezeichnung	Bedeutung
side	left oder right, je nachdem ob es sich um ein Segment der linken oder rechten Seite handelt
SID	Segment-ID, also ID der Zone
force	Kraft, die auf dieser Zone gemessen wurde, Berechnung erfolgt nach Formel 2.1
force_rel_to_all	Kraft, die auf dieser Zone wirkt in Relation zur Kraft die auf der gesamten Sensor-Fläche wirkt
force_rel_to_side	Kraft, die auf dieser Zone wirkt in Relation zur Kraft die auf allen Zonen der Seite wirkt
force_rel_to_area	Kraft, die auf dieser Zone wirkt in Relation zur Fläche des Abdrucks in dieser Zone
area	Fläche des Abdrucks in dieser Zone
area_unclipped	Fläche der Zone
area_with_pressure	belastete Abdruckfläche in dieser Zone
area_with_pressure_unclipped	Summe der Fläche aller belasteten Sensoren in der Abdruckfläche dieser Zone

Die Kraft, die auf eine Zone  $i$  wirkt (mit insgesamt  $n$  Zonen), wird wie folgt berechnet aus den Kräften die auf die Sensoren  $S$  wirken:

$$F_i = \sum_{s \in S} \frac{A_s \cap A_i}{\sum_{j=1}^n A_s \cap A_j} F(A_s) \quad (2.1)$$

## WEITERE MÖGLICHKEITEN

## 3.1 Einzelnes Anpassen der Zonen

Angenommen, man hat bereits den Fit angepasst, gespeichert und möchte nun noch eine kleinere Rotation an den Zonen vornehmen. Dies ist wie folgt möglich, man sollte jedoch im Hinterkopf behalten, dass man hiermit nicht die Skalierung der Zonen anpasst.

Zunächst sollte man seinen gespeicherten Fit wieder auslesen, durch den READ TRANSFORMATION Codeblock im R Script:

---

```
# ----- READ TRANSFORMATION
# -----
transform = unname(as.matrix(read.csv(file = paste0(bone, "/
  transformation_r_imprint_",ground_type, ".csv"), header=F)))
transform_zones_left = unname(as.matrix(read.csv(paste0(bone, "/
  transformation_r_zones_left_",ground_type, ".csv"), header = F)))
transform_zones_right = unname(as.matrix(read.csv(paste0(bone, "/
  transformation_r_zones_right_",ground_type, ".csv"), header = F)))

segments_imprint_right_orig <- read_csv(paste0(bone, "/segments_imprint_
  right", ".csv"))
segments_imprint_left_orig <- read_csv(paste0(bone, "/segments_imprint_
  left", ".csv"))

segments_measurements_right_orig <- read_csv(paste0("segments_for_
  measurements_right", ".csv"))
segments_measurements_left_orig <- read_csv(paste0("segments_for_
  measurements_left", ".csv"))

segments_imprint_right <- segments_imprint_right_orig
segments_imprint_left <- segments_imprint_left_orig
segments_measurements <- segments_measurements_right_orig
segments_measurements <- segments_measurements_left_orig

segments_imprint_left[c("x", "y")] = t(apply(segments_imprint_left, 1,
  trans_func, mat = transform))
segments_imprint_right[c("x", "y")] = t(apply(segments_imprint_right, 1,
  trans_func, mat = transform))

segments_measurements_left[c("x", "y")] = t(apply(segments_measurements_
  left_orig, 1, trans_func, mat = transform_zones_left))
segments_measurements_right[c("x", "y")] = t(apply(segments_measurements
  _right_orig, 1, trans_func, mat = transform_zones_right))
```

---

Danach sollte man eine neue Transformation für eine der Zonen anlegen, (siehe TRANSFORM FIT Block):

Listing 3.1: Definition einer neuen Transformation

---

```

rot = rotation_matrix(-20, radian=F)
rot_pivot = matrix(c(8,11), nrow=2)
displace = matrix(c(0,0), nrow=2)

transform_extra = rbind(cbind(c(1,0),c(0,1),displace),c(0,0,1))%%
  rbind(cbind(c(1,0),c(0,1),rot_pivot),c(0,0,1))%%
  rbind(cbind(rot,c(0,0)),c(0,0,1))%%
  rbind(cbind(c(1,0),c(0,1),-rot_pivot),c(0,0,1))

```

---

Dann wird diese Transformation wie folgt auf die Transformation der jeweiligen Zone angewandt:

---

```
transform_zones_left = transform_extra%%transform_zones_left
```

---

Bzw. für die rechten Zonen:

---

```
transform_zones_right = transform_extra%%transform_zones_right
```

---

Und letztendlich das Segment transformiert mit

---

```
segments_measurements_left[c("x", "y")] = t(apply(segments_measurements_
  left_orig, 1, trans_func, mat = transform_zones_left))
```

---

Bzw. für die rechten Zonen:

---

```
segments_measurements_right[c("x", "y")] = t(apply(segments_measurements
  _right_orig, 1, trans_func, mat = transform_zones_right))
```

---

Anschließend kann man sich den Fit plotten lassen und die Schritte wiederholen, indem man wieder anfängt und eine weitere zusätzliche Transformation definiert, siehe Listing 3.1. Oder von neuem beginnt und dadurch den Fit wieder zurücksetzt.

Wenn man zufrieden ist führt man den SAVE TRANSFORMED FIT Block aus, um den Fit zu speichern:

---

```

# ----- SAVE TRANSFORMED FIT -----
write.table(file = paste0(bone, "/transformation_r_imprint_",ground_type
  , ".csv"), x= transform, row.names = FALSE, col.names=FALSE, sep=",
  ")
write.table(file = paste0(bone, "/transformation_r_zones_left_",ground_
  type, ".csv"), x= transform_zones_left, row.names = FALSE, col.
  names=FALSE, sep=",")
write.table(file = paste0(bone, "/transformation_r_zones_right_",ground_
  type, ".csv"), x= transform_zones_right, row.names = FALSE, col.
  names=FALSE, sep=",")

```

---

Dann wiederholt man Schritt 2 des Python Scripts, um die Statistik zu aktualisieren:

---

```
C:\Users\Kai\Documents\ProKlaue\scripts>python pressureStatistics.py -s  
2 -t 0.5 -b "Klaue 1 (KIT)" -g "gummi" -d "C:/Users/Kai/Documents/  
ProKlaue/testdaten/druck"
```

---