



universität
wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis:

"Berechnung von Key Performance Indicators aus Positionsdaten im Bereich Fußball"

verfasst von / submitted by:

Alexander Mikosch

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2019/ Vienna, 2019

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:

UA 066 821

Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:

Masterstudium Mathematik

Betreut von / Supervisor:

Univ.-Prof. Dipl.-Ing. Dr. Arnold Baca

Mitbetreut von / Co-Supervisor:

Mag. Dr. Roland Leser

Abstract

Das Ziel der Masterarbeit ist eine Methodik zu entwickeln um Key Performance Indikatoren berechnen zu können. Diese KPI's verändern die Spielanalyse enorm, da Positionsdaten im Vergleich zu anderen Analysemethoden neu sind. In dieser Masterarbeit findet man zwei KPI's:

- 1.) das Defensives Packing, welches die überspielten Gegner bei einem Pass zählt
- 2.) die Raumkontrolle, in der die Fläche, die ein Team kontrolliert, berechnet wird.

Durch Methodiken aus dem Mathematikstudium habe ich die KPI's in ihre Einzelteile zerlegt und diese mit abstrakten Methoden beschrieben.

Damit man auch mit den sehr umfangreichen Positionsdaten umgehen kann, wurden zwei Programme erstellt, welche die Methodiken darauf anwenden.

Die Ergebnisse dieser Masterarbeit sind somit die erstellten Programme und die entwickelten Methodiken zur Berechnung der KPI's.

Diese Masterarbeit ist für alle Sportwissenschaftler sowie Mathematiker und allen anderen Interessenten aus dem Bereich Sportanalyse lohnenswert.

The goal of this master thesis is to develop a methodology to calculate key performance indicators. These KPI's change the game analysis enormously, because position data are new compared to other analysis methods. In this master thesis you will find two KPI's:

- 1) the Defensive Packing, which counts the overplayed opponents on a pass
- 2) the room control, in which the area that a team controls is calculated.

Using methods from my mathematics courses, I have broken down the KPI's into their individual parts and described them using abstract methods.

In order to be able to deal with the very extensive position data, two programs were created which apply the methodologies to those position data.

The results of this master thesis are thus the created programs and the developed methods for calculating the KPI's.

This Master's thesis is intended for sports scientists, mathematicians and all other Interested individuals from the field of sports analysis.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Erklärung Datensätze	2
1.1.1	Positionsdaten	2
1.1.2	Eventdaten	3
1.1.3	Extra Daten	3
1.2	Definitionen und Theoreme	4
1.3	Mathematische Notation	5
2	Defensives Packing	6
2.1	Erklärung	6
2.2	Methode	7
2.2.1	Allgemeines	7
2.2.2	relevante Pässe	7
2.2.3	Überspielte Gegner	7
2.2.4	Gütefunktion 1	8
2.2.5	Gütefunktion 2	8
2.3	Beispiel	8
3	Raumkontrolle	10
3.1	Erklärung	10
3.2	Methode	11
3.2.1	Kurze Zusammenfassung	11
3.2.2	Allgemeines	11
3.2.3	Mittelsenkrechte	11
3.2.4	Schnittpunkte	12
3.2.5	Rangbedingungen	12
3.2.6	Löschen von überflüssigen Schnittpunkten	12
3.2.7	Zusammenfügen von doppelten Punkten	13
3.2.8	Die richtigen Schnittpunkte	14
3.2.9	Flächenberechnung	14
3.2.10	Iteration	15
3.3	Beispiel	16
4	Zusammenfassung	17
	Literaturverzeichnis	19
	Anlagen	20
	Ergebnisdatei Packing	20
	Ergebnisdatei Raumkontrolle	25

Kapitel 1

Einleitung

Ich zitiere hier einmal Ralf Rangnick: „Die moderne Spielanalyse umfasst mittlerweile mehr als die Auswertung von Zweikampfquoten oder zurückgelegten Kilometern- denn es hat sich herausgestellt, dass diese Daten nicht über Sieg oder Niederlage entscheiden. Vielmehr sind es deutlich komplexere Analysen mit neuartigen Leistungsindikatoren, welche neue Erkenntnisse im Spitzensfußball liefern werden “[[Memmert](#), p.IX]

In einem Review [[Review](#)] verfasst von Daniel Memmert, werden ein paar solcher Leistungsindikatoren beschrieben. Er beschreibt dort die ersten Ansätze für die möglichen Auswirkungen von den Indikatoren auf das Spiel. Zwei von diesen Leistungsindikatoren

möchte ich in dieser Masterarbeit berechnen. Diese Indikatoren werden üblicherweise als KPI´s (Key Performance Indikator) bezeichnen.

Der erste Wert nennt sich „Defensives Packing “oder auch „Outplayed Opponents “. Dabei betrachtet man zwei Zeitpunkte bei einem offensiven Pass: den Zeitpunkt des Spielens vom Ball sowie den Zeitpunkt wenn ein Mitspieler den Ball empfängt. Es werden jeweils die Spieler aus der gegnerischen Mannschaft, die zwischen dem gegnerischen Tor und dem Passspieler bzw. Passempfänger gezählt und anschließend voneinander abgezogen. Also vereinfacht gesagt, wie viele Gegenspieler wurden mit einem Pass überspielt.

Der zweite Wert nennt sich „Space control “bzw Raumkontrolle. Hier berechnet man die Fläche die einer Mannschaft gehört. Genauer gesagt, berechnet man die Fläche die ein Spieler vor allen anderen Spielern erreicht und wenn man dann alle Spielerflächen von einem Team addiert bekommt man die Teamfläche.

In dieser Masterarbeit beschreibe ich die Methodik für die Berechnung der KPI's. Zusätzlich habe ich noch jeweils ein Programm in Java für die zwei KPI's geschrieben. Mit diesen Programmen habe ich meine Auswertungsdaten erstellt. Bei Interesse an den Programmen können Sie sich gerne bei mir unter folgender Mail melden:
alex.mik@gmx.net

Natürlich benötige für die Berechnung der KPI's einen Rohdatensatz. Dieser besteht aus Positionsdaten sowie Eventdaten. Diese Daten bekomme ich von Rapid Wien gestellt. Für die gute Zusammenarbeit mit Herrn Oesen, dem Video-Analysten von Rapid Wien, sowie der Bereitstellung der Daten bedanke ich mich hiermit.

Aufgrund von Auflagen des ÖFB zur Anonymisierung werden Vereine und Spieler nicht namentlich genannt sonder nur deren Identifikationsnummern.

1.1 Erklärung Datensätze

Positionsdaten können durch drei Arten aufgenommen werden. Die drei möglichen Systeme sind: GPS basierende Systeme, Video basierende Systeme mit Hilfe von Bildverarbeitung, Radar und Mikrowellen basierende Systeme.

Häufig werden die Video basierte Systeme eingesetzt wie auch in meinem Datensatz. Mit diesen Systemen können verschiedene Daten aufgenommen werden. Diese Daten werden dann in drei verschiedenen Datensätzen abgespeichert. Folgend werde ich die verschiedenen Datensätze erklären

1.1.1 Positionsdaten

Die erstellte TXT-Datei von einem Spiel enthält die Koordinaten der Spieler und Informationen um die Koordinaten auch den richtigen Spieler zuzuordnen. Im folgenden wird anhand einer Beispielszeile erklärt, welche Informationen abgespeichert werden. Folgend ist eine Beispielzeile aus einer TXT-Datei, bei der Zahlen die zur gleichen Kategorie zugehören gleich eingefärbt sind. Eine Beispielzeile aus einer TXT-Datei:

```
1566147600000;0,1,0:0,572641,5,72.74,28.054;  
0,951820,10,52.033,33.742;0,922840,8,67.081,14.396;  
0,734193,23,69.191,39.348;0,565551,9,51.971,43.514;  
0,518045,21,61.157,37.798;0,507915,4,61.729,56.048;  
0,499543,18,50.079,21.915;0,421600,13,52.255,52.89;  
0,401914,24,57.76,25.778;1,521238,8,43.807,29.582;  
1,831306,14,41.907,31.666;1,1093729,3,33.778,44.9;  
1,620031,9,52.316,20.817;1,660020,10,45.963,43.466;  
1,660022,13,41.803,54.218;1,831307,7,53.214,42.41;  
1,1061263,31,37.399,13.844;1,600908,17,32.296,35.304;  
1,734284,20,33.559,22.06;2,581275,92,73.107,69;  
2,458704,90,61.039,46.615;2,1093716,91,32.069,-1;  
3,507906,27,99.658,29.929;4,582279,1,4.251,33.771::52.5,34,0;
```

Der erste Eintrag ist die Systemzeit (1566147600000) Genauer gesagt, die Zeit seit dem die Systemzeit eingeführt wurde. Die Einheit der Angabe ist Millisekunden.

Danach folgt die Zeit ab Start des Spiels (0), ebenfalls in Millisekunden.

Die anschließende Zahl (1) gibt Aufschluss über die Halbzeit, denn diese zeigt 1 für die erste sowie 2 für zweite Halbzeit an, bei einer möglichen Verlängerung erhöhen sich die Zahlen dazu.

Der vierte Eintrag sagt etwas über den Status des Spiels aus (0 für es läuft gerade ,1 es pausiert).

Ab dem Doppelpunkt starten dann die Positionsdaten der Spieler. Jeder nachfolgende Spieler ist mit einem Semikolon getrennt.

Aufgezählt werden:

Teamzugehörigkeit (0 Heim, 1 Auswärts, 2 Schiedsrichter , 3 Heim Torhüter, 4 Auswärts Torhüter), **Identifikationsnummer** , **Trikotnummer**, **X-Achsenwert**, **Y-Achsenwert**.

Wenn alle Spieler aufgezählt wurden, folgen noch die **X-,Y,Z-Koordinaten des Balles**. Die Koordinaten sind in der unteren linken Ecke genullt. Die Länge des Spielfeldes ist 105 Meter und die Breite 68 Meter. Der Nullpunkt (0,0) in diesem Koordinatensystem liegt also bei der unteren linken Ecke des Spielfeldes.

Jede 100 Millisekunden wird eine weitere Zeile erfasst. Bei einem Spiel entstehen so mindestens $90[\text{min}] \cdot 60[\text{sek}] \cdot 100/1000[\text{millisek}] = 54000$ Zeilen.

1.1.2 Eventdaten

Die Eventdaten sind eine XML-Datei in der jede Ballaktion genau gespeichert wird. Eine Beispielzeile:

```
<Event Time="1333" IdActor1="831883" IdActor2="" EventName="Reception"
BodyPart="foot" Behaviour="" DuelType="" DuelBrutality=""
DuelInitiative="" DuelWinner="" LocationX="-1151" LocationY="-373"
LocationZ="" TargetX="" TargetY="" TargetZ="" HitsPost="False"
Blocked="" PhaseType="OpenPlay" PhaseStartTime="125"
PhaseEndTime="86596" PhaseSubType="" StartOfPlay=""
BreakThroughBall="False" AssistType="" BallCurve=""
TechnicalCharacteristics="" FoulReason="" GKHeightOfIntervention=""
DuelOutcome="" ScoreHomeTeam="0" ScoreAwayTeam="0" RedCardsHomeTeam="0"
RedCardsAwayTeam="0" />
```

Die Begriffe sind selbsterklärend. Hier wird die Zeit in Millisekunden genau gemessen. Der Nullpunkt (0,0) liegt hier am Anstoßkreis. Somit können auch negative Koordinaten entstehen. Die Spannweite der x-Achse (Länge des Feldes) reicht von -52,5 bis 52,5 und bei der y-Achse (Breite) von -34 bis 34.

1.1.3 Extra Daten

Um beide Dateien (TXT, XML) zu kombinieren, braucht man noch eine weitere Datei (CSV). Denn die beiden Dateitypen haben jeweils unterschiedliche Identifikationsnummern für die Spieler. In der CSV-Datei finden man alle Spieler mit ihren beiden Identifikationsnummern aufgezählt. Eine Beispielzeile:

```
7,1,Philipp,Schobesberger,1,10-DEC-93,1564941600000,1564944661800,
589406,831307,2150250,2150250;
```

Die viertletzte Zahl (589406) ist die XML-ID und die drittletzte Zahl (831307) ist die TXT-ID. Zusätzlich kann man hier dann auch den Namen der Spieler mit den ID's verknüpfen. Man findet hier ebenfalls die Trikotnummer (7) und das Geburtsdatum. Alle anderen Daten sind für diese Arbeit nicht von Interesse.

1.2 Definitionen und Theoreme

Definition 1. Die Indikator-Funktion $\mathbb{1}_{\{\}}$ hat den Wert 1 falls die gesamte Bedingung in der geschweiften Klammer erfüllt ist, 0 sonst.

Definition 2. Intervalle können offen, geschlossen oder halb offen/halb geschlossen sein. Offen bedeutet, dass der Randpunkt nicht mehr in der Menge enthalten ist, geschlossen bezeichnet das Gegenteil. Offen Intervalle werden mit runden Klammern $()$ geschrieben, geschlossen mit eckigen $[]$.

Satz 1. Hat man zwei Punkte $P = (p_x, p_y)$ $Q = (q_x, q_y)$ gegeben, dann kann man mit deren Koordinaten eine Gerade erstellen, welche durch beide Punkte verläuft und die Geradengleichung berechnet man wie folgt:

$$m = \frac{q_y - p_y}{q_x - p_x} \quad (1.1)$$

$$b = \frac{q_y - p_y}{q_x - p_x} * p_x - p_y \quad (1.2)$$

$$g(x) = y(x) = m * x + b \quad (1.3)$$

Satz 2. Sind zwei Geraden f, g orthogonal zueinander so gilt:

$$m_f * m_g = -1 \quad (1.4)$$

Außer für eine oder beide Steigungen gilt $m_{\{f, g\}} \in \{0, \infty, -\infty\}$

Satz 3. Die Fläche eines Dreiecks mit Eckpunkten $A = (a_x, a_y), B, C$ lässt sich folgendermaßen berechnen.

Mit $\overrightarrow{AB} = \begin{pmatrix} b_x - a_x \\ b_y - a_y \end{pmatrix}; \overrightarrow{AC} = \begin{pmatrix} c_x - a_x \\ c_y - a_y \end{pmatrix}$ bekommen wir:

$$\begin{aligned} \text{Fläche}_{\Delta ABC} &= \frac{1}{2} \left| \overrightarrow{AB} \quad \overrightarrow{AC} \right| \\ &= \frac{1}{2} \left| \begin{matrix} b_x - a_x & c_x - a_x \\ b_y - a_y & c_y - a_y \end{matrix} \right| \\ &= \frac{1}{2} ((b_x - a_x) * (c_y - a_y) - (b_y - a_y) * (c_x - a_x)) \end{aligned} \quad (1.5)$$

1.3 Mathematische Notation

Die jetzt eingeführte Notation wird in den nachfolgenden Kapiteln benutzen.

Es gibt zwei Teams: das angreifende Team A , das verteidige Team V .

Spieler dieser Teams werden mit A_i bzw. V_i bezeichnet wobei i von 1 bis 14 sein kann. Da es die elf startenden Spieler gibt und zusätzlich noch 3 Spieler eingewechselt werden können. Allgemeine Spieler ,wenn das Team irrelevant ist, werden S_i genannt. Wird von einem Spieler nur die x-Achse betrachtet, so wird x indiziert (Beispielhaft A_{i_x}). Ebenso für die y-Achse.

Da das Tor des verteidigen Team eine wichtige Rolle spielt, wird dies auch $VTor$ genannt.

Es gibt noch zwei besondere Spieler, diese werden für „überspielte “ Gegner besonders herausgehoben. Es sind folgende Spieler: der Pass abgebender Spieler Apa und der Pass empfangender Spieler Ape .

Zeitangaben werden mit t bezeichnet.

Kapitel 2

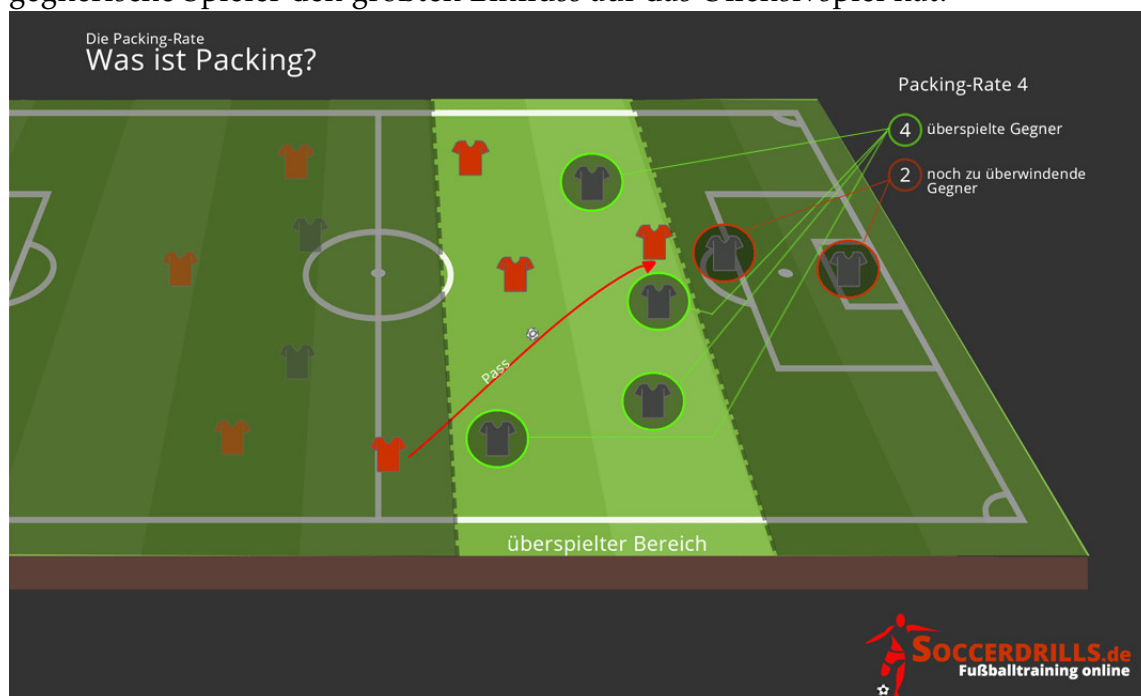
Defensives Packing

2.1 Erklärung

Defensives Packing ist ein Kriterium für die Verteidigung bei einem Gegnerischen Angriff. Sie zeichnet die Differenz der Anzahl der eigene Spieler, welche sich zwischen dem Ball und dem eigenen Tor befinden, vor und nach einer gelungenen Offensivaktion auf. Die Offensivaktion könnte eine Flanke, ein Pass oder ein Dribbling sein. **[Packing]** In dieser Arbeit betrachtet man nur offensive Pässe des angreifenden Teams, dh. der Pass muss von Team A gespielt werden und sich in der x-Achsen-Richtung zum verteidigenden Tor bewegen und auch ankommen. Alle anderen Fälle werden nicht betrachtet.

Man fixiert dabei die Teams im Vorhinein als angreifend oder verteidigend und macht dies nicht von der Spielsituation abhängig. Die überspielten Spieler können ihr Tor nicht mehr verteidigen. Sie sind daher „gepackt“ und aus dem Spiel genommen.

Deshalb ist dieses Kriterium besonders interessant um herauszufinden, welcher gegnerische Spieler den größten Einfluss auf das Offensivspiel hat.



[Packing Bild]

Mit der Auswertung der „Packing“ Methode sind noch viele weitere Berechnungen möglich. Mögliche interessante Fragestellungen wären, falls man sich den Zielspieler oder den Ballverlust näher anschaut:

Wer wird wie häufig angespielt? An welchem Ort steht der Zielspieler.

Wie schwer wiegend ist ein Ballverlust, anhand der Mitspieler die noch zwischen Ball und Tor stehen?

Oder man kann daraus auch noch Häufigkeiten berechnen, wie z.B. wie häufig wurden mehr als x Spieler überspielt? etc.

2.2 Methode

2.2.1 Allgemeines

Wie in der Einleitung beschrieben scheint die Berechnung recht einfach zu sein. Trotzdem ergeben sich ein paar Probleme. Wie in den Erklärungen der Datensätze beschrieben, haben die Rohdaten unterschiedliche Identifikationsnummern sowie die Koordinaten. Diese muss man zunächst einmal normieren. Dies ist durch Überschreiben der ID's sowie umrechnen der Koordinaten gelöst.

In keiner Datei ist beschrieben, welches Team auf welches Tor spielt. Um das Problem zu beheben, liest das Programm zunächst die Position von beiden Torhütern ein und vergleicht die x-Achsenabschnitte mit Kontrollwerten, das heißt, wenn der x-Wert Torhüter von Team A zwischen 0-52,5 liegt spielt das Team von Links nach Rechts, bei 52,5-105 genau andersherum.

2.2.2 relevante Pässe

Da nur die Pässe vom angreifenden Team von Interesse sind, durchsucht das Programm die Eventdaten nach Pässe von ID's der angreifenden Mannschaft. Anschließend betrachtet das Programm die Koordinaten des Pass abgebenden Spielers Apa sowie des Pass empfangenden Spielers Ape . Ist nun Ape_x näher als Apa_x zum verteidigenden Tor ($VTor_x$) so wird dieser Pass gespeichert. Näher bedeutet in diesem Zusammenhang, dass die Distanz in x von Ape zu VTor kleiner ist als von Apa zu VTor. Wenn der Pass gespeichert wird, werden zusätzlich noch die ID's von Apa und Ape gespeichert sowie die genaue Zeit des Passes.

Da in der XML-Datei die Zeit genauer gemessen wird als in der TXT-Datei muss man zunächst die Zeit auf die vorherigen vollen 100 Millisekunden runden. Dies wird bei beiden Zeitpunkten gemacht (Passabgabe t_1 , Passempfang t_2).

2.2.3 Überspielte Gegner

Nun liest man alle x-Achsen Wert von den verteidigenden Spielern (V_i) ein.

Mit folgender Formel erhält man die Anzahl der überspielten verteidigenden Spieler falls $VTor_x=105$:

$$\sum_{i=1}^{11} \mathbb{1}_{\{t=t_1, V_{i_x} \in [Apa_x, 105]\}} - \mathbb{1}_{\{t=t_2, V_{i_x} \in [Ape_x, 105]\}} \quad (2.1)$$

Falls $VTor_x = 0$ benutzt man diese :

$$\sum_{i=1}^{11} \mathbb{1}_{\{t=t_1, V_{i_x} \in [0, Apa_x]\}} - \mathbb{1}_{\{t=t_2, V_{i_x} \in [0, Ape_x]\}} \quad (2.2)$$

Ein Spieler kann nur überspielt werden wenn er zum ersten Zeitpunkt t_1 zwischen dem Ball und dem verteidigenden Tor (VTor) steht. Verteidigende Spieler, die beim zweiten Zeitpunkt t_2 zwischen dem Ball und dem verteidigenden Tor (VTor) stehen, wurden nicht überspielt. Somit muss man diese noch von der ersten Anzahl abziehen.

2.2.4 Gütefunktion 1

Zusätzlich gibt das Programm noch 2 Gütefunktionen aus. Die erste wird folgendermaßen berechnet:

$$G_1 = \mathbb{1}_{\{Ape_y \in [13.5, 54.5], Ape_x \in [0, 21], VTor_x = 0\}} * \mathbb{1}_{\{Ape_y \in [13.5, 54.5], Ape_x \in [84, 105], VTor_x = 105\}}. \quad (2.3)$$

Diese Gütefunktion beschreibt ob ein Pass in den 16-Meterraum gespielt wird. Da Spielsituation im 16-Meterraum gefährlicher sind als außerhalb, ist dies ein guter Hinweis auf die Wichtigkeit des Passes.

2.2.5 Gütefunktion 2

Da man Pässe gerne vergleichen möchte, sollten diese auch eine Wertung bekommen. So eine Wertung bekommt man mit der zweiten Gütefunktion hin. Die zweite Gütefunktion schaut wie folgt aus:

$$G_2 = (\mathbb{1}_{\{Ape_y \in [0, 34]\}} \frac{|Ape_y|}{34} * \frac{|Ape_x - Apa_x|}{105} * \frac{|Ape_y - Apa_y|}{68} + \mathbb{1}_{\{Ape_y \in (34, 68]\}} \frac{|Ape_y - 68|}{34} * \frac{|Ape_x - Apa_x|}{105} * \frac{|Ape_y - Apa_y|}{68}) * 100 \quad (2.4)$$

Diese Gütefunktion gibt Werte zwischen 0 und 100 aus. Der Wert steigt unter verschiedenen Bedingungen:

- a) je geringer die Distanz vom Zweiten Spieler in der y-Achse zum wert $y_{mitte}=34$ ist
- b) je größer die Distanz vom Pass abgebenden und Pass empfangenden Spieler in der x-Achse ist
- c) je größer die Distanz vom Pass abgebenden und Pass empfangenden Spieler in der y-Achse ist

Der erste Bruch in der Formel beschreibt a), der Zweite b), der Dritte c). Jeder Bruch soll für sich selbst stehend nur einen Wert zwischen 0 und 1 haben. Somit sind die Nenner passend dafür gewählt.

Zusammengefasst sagt die Funktion viel über die Distanz des Passes und über den Ort des angespielten Spielers in y-Richtung aus.

2.3 Beispiel

Im folgenden Beispiel wird ein Fußballspiel aus der Österreichischen Bundesliga betrachtet. Der komplette ausgewertete Datensatz liegt in den Anlagen. Aus diesem Datensatz erhält man dann mit dem Auslesen der Datei und passender Addition, Mittelwertberechnung und Zählen die folgenden Ergebnisse in den Tabellen. Um daraus passende Rückschlüsse auf Spielsituationen, der Effektivität einzelner Spieler, Schlüsselspieler etc zu bekommen, muss man viel Erfahrung damit haben und sich zusätzlich mehr Dateien, Spiele und Analysen anschauen. Die Berechnung solcher Rückschlüsse wäre einer eigenen Masterarbeit gleichzusetzen, deshalb folgt hier nur ein Auszug. Wenn man diesen Auszug für mehrere Spiele berechnet, kann aber schon versuchen passende Rückschlüsse zu ziehen.

Tabelle 2.1: Anzahl von Pässen in Abhängigkeit von Spielern

SpielerID	Anzahl der Passabgabe von	Anzahl der empfangenden Pässe von
9608977	24	0
582760	24	8
573158	21	10
369331	29	12
9611576	11	22
349309	28	26
375478	10	34
977414	4	26
362784	38	24
606463	3	12
1061409	24	19
828351	0	2
973044	1	6
596555	0	16

Tabelle 2.2: Anzahl von bestimmten Pässen

	Anzahl der Pässe mit	Anzahl der Pässe mit
Anzahl der ...	überspielten Gegner	noch zu überwindenden Gegnern
0	57	0
1	44	4
2	45	3
3	26	11
4	16	19
5	10	20
6	7	22
7	5	38
8	4	32
9	1	33
10	1	15
11	0	20

Tabelle 2.3: Mittelwerte

	Überspielte Gegner	Noch zu überwindende Gegner	Gütefunktion
Summe aller	453	1540	1616,4557
Mittelwert aller	2,0779816514	7,0642201835	7,4149344037

Tabelle 2.4: Gefährliche Zone

	in der gefährlichen Zone	außerhalb der gefährlichen Zone
Pässe	8	210

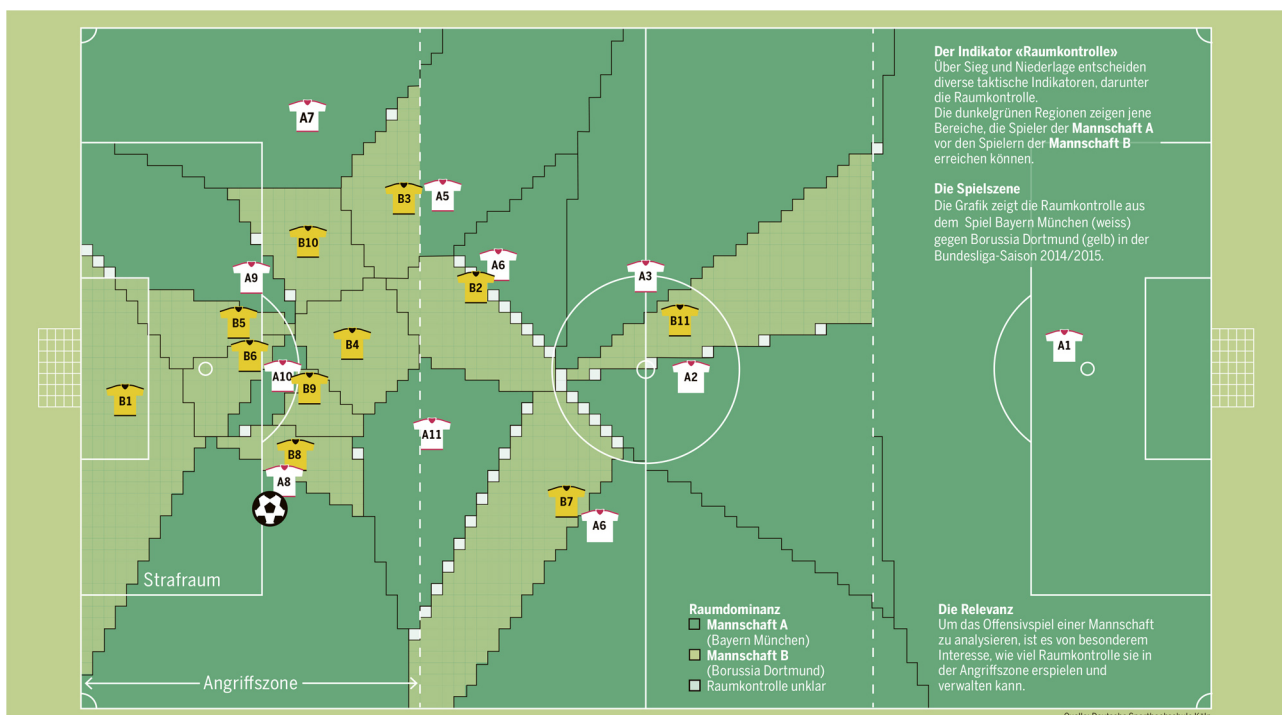
Kapitel 3

Raumkontrolle

3.1 Erklärung

"Ballbesitzstatistik, Passquoten, Laufdistanzen oder Heatmaps mit den Aufenthaltsorten der Spieler – heute vergeht kein professioneller Fußballmatch, bei dem das Geschehen nicht haufenweise mit Daten aufbereitet wird." [Raumkontrolle]

Immer größerer Beliebtheit bekommt die Raumkontrolle. Sie beschreibt die Herrschaft einer Mannschaft über die Räume auf dem Fußballfeld und ist durch jene Rasenfläche definiert, die Spieler einer Mannschaft vor den Spielern der gegnerischen Mannschaft erreichen können.



[Raumkontrolle]

Das Bild zeigt die Methodik von dem Tool „Soccer“ welches von Daniel Memmert und Jürgen Perl entwickelt wurde.

Wie man aber erkennen kann, ist dieses noch ein wenig ungenau. Mit meiner Methodik beseitige ich diese Ungenauigkeit. Man fixiert die Teams als angreifend oder verteidigend. Der Raum, welcher aus den letzten $\frac{2}{3}$ der gesamten Fläche besteht, somit vom Verteidigenden Tor bis etwas weiter als die Mittellinie, wird nur berechnet. Da der restliche Raum für den Angriff irrelevant scheint. Diese $\frac{2}{3}$ Grenze kann man aber auch auf $\frac{1}{3}$ oder auf die gesamte Fläche verändern um noch mehr nützliche Daten zu erstellen.

Man berechnet genau den Raum, den ein Spieler vor allen anderen erreichen kann. Danach addiert man die Fläche von allen Spielern von einem Team und bekommen so die Fläche die von dieser Mannschaft kontrolliert wird.

Die Flächen berechnet man jede $\frac{1}{10}$ Sekunde . Folglich haben man eine fast „kontinuierlich“ verändernde Raumkontrolle. Mit der Auswertung meiner "Raumkontrolle" Methode sind noch viele weitere Berechnungen möglich.

Mögliche interessante Fragestellungen wären:

Welcher Spieler im Ballbesitz verändert den Raum am stärksten?

Welcher Pass/Lauf/Bewegung bringt einem wie viel neuen Raum?

Wo sind Lücken in der Raumkontrolle?

3.2 Methode

3.2.1 Kurze Zusammenfassung

Um die Genauigkeit zu bekommen, von der in der Erklärung gesprochen wurde, benutzte ich eine andere Methodik als die von Memmert. Memmert hat im Prinzip die Abzählmethode benutzt, dh. das Spielfeld in $1m * 1m$ Kästchen aufgeteilt und dann die Kästchen, den nächststehenden Spielern zugeordnet.

Grob beschrieben erstellt man eine Mittelsenkrechte von einem Spieler zu jedem anderen Spieler, lässt diese Mittelsenkrechten mit einander schneiden und so bekommt man den Raum für diesen Spieler.

3.2.2 Allgemeines

Da man hier nur die TXT-Datei benutzt fällt die Normierung der Datensätze weg.

Wie beim Packing ist in keiner Datei beschrieben, welches Team auf welches Tor spielt. Um das Problem zu beheben, liest das Programm zunächst die Position von beiden Torhütern ein und vergleicht die x-Achsenabschnitte mit Kontrollwerten, das heißt, wenn der x-Wert Torhüter von Team A zwischen 0-52,5 liegt spielt das Team von Links nach Rechts , bei 52,5-105 genau andersherum.

3.2.3 Mittelsenkrechte

Nun ausführlich im Detail: Man fixiert einen Zeitpunkt t. Man fixiert einen Spieler S_1 .

Nun berechnet man die Mittelsenkrechte von S_1 zu S_i wobei i alle anderen Spieler auf dem Spielfeld sind (somit $i \in \{2, 3, \dots, 24\}$). Die Mittelsenkrechten trennen den Raum auf in 2 Teile, jeweils der Teil indem der Spieler 1 oder i liegt, kontrolliert dieser dann auch. Die Gleichung erstellt man durch die Kombination von Satz 1 und 2.

$$M_{1,i} = \left(\frac{S_{i_x} + S_{1_x}}{2}, \frac{S_{i_y} + S_{1_y}}{2} \right) \quad (3.1)$$

$$m_{1,i} = -\frac{S_{i_x} - S_{1_x}}{S_{i_y} - S_{1_y}} \quad (3.2)$$

$$b_{1,i} = M_{1,i_y} - M_{1,i_x} * m_{1,i} \quad (3.3)$$

Wobei man mit $M_{1,i}$ den Mittelpunkt von S_1 und S_i bezeichnet. Die Geradengleichung ist somit gegeben:

$$g_{1,i} = x * m_{1,i} + b_{1,i} = x * -\frac{S_{i_x} - S_{1_x}}{S_{i_y} - S_{1_y}} + M_{1,i_y} - M_{1,i_x} * \left(-\frac{S_{i_x} - S_{1_x}}{S_{i_y} - S_{1_y}} \right) \quad (3.4)$$

3.2.4 Schnittpunkte

Anschließend berechnet man die Schnittpunkte von allen Geraden, die man gerade erstellt habe. Indem man zwei Geraden $g_{1,i}$ und $g_{1,j}$ mit einander gleichsetzt und dann nach x auflöst. Zusätzlich fügt man das x wieder in eine der beiden Gleichungen ein und erhält y . Diesen Schnittpunkt wird als $INT_{i,j}$ definiert, für den englischen Begriff Intersection.

3.2.5 Rangbedingungen

Da die Fußballspielfläche beschränkt ist braucht man noch Randbedingungen. Für die Randbedingungen unten und oben lautet die Geradengleichung $g_o = 68$ $g_u = 0$ Für die Randbedingung links und rechts ist die Vorgehensweise mit dem Erstellen von Geraden nicht mehr komplett möglich. Denn zum Beispiel die Randbedingung links würde durch den Punkt $(0,0)$ sowie $(0,68)$ und allen Punkten dazwischen gehen. Eine Geradengleichung kann dafür nicht erstellt werden, da zu einem x -Wert kein eindeutiger y -Wert berechnet werden kann.

Man benutzt einen kleinen Trick um dieses Problem zu umgehen.

Für die Randbedingung links (also wenn $x=0$ ist) kann man die y -Koordinate des Schnittpunktes einfach von dem b von der anderen Geraden auslesen und bekommt den Schnittpunkt $(0,b)$.

Für die Randbedingung rechts (also wenn $x=68$ ist) kann man die y -Koordinate des Schnittpunktes mit der anderen Geradengleichung berechnen und bekommt den Schnittpunkt $(0,m*68+b)$.

Nun berechnet man noch alle Schnittpunkte der Geraden mit den 4 Randbedingungen.

Zusätzlich fügt man noch die Schnittpunkte an den Ecken hinzu $(0,0)(0,68)(105,0)(105,68)$.

Ab jetzt wird wenn man von Geraden spricht, die Mittelsenkrechten sowie die Randbedingungen meinen.

3.2.6 Löschen von überflüssigen Schnittpunkten

Jetzt löscht man alle Schnittpunkte $INT_{i,j}$ welche hinter einer Geraden $g_{1,k}$ liegen, wobei $k \notin \{i, j\}$. Das bedeutet man schaut sich jede Gerade an, orientieren diese zum Spieler und löschen alle Schnittpunkte die auf der anderen Seite liegen. Man teilt mit diesen Geraden sozusagen den Raum auf in 2 Teilräume, in einem Raum liegt der Spieler S_1 und in dem anderen liegt er folgerichtig nicht. Anschließend löscht man also alle Schnittpunkte in dem „anderen“ Raum.

Man geht folgendermaßen vor: Man nimmt die Steigung $m_{1,k}$ berechnet mit dieser und dem Koordinaten von Spieler 1 einen Referenzwert.

$$S_{1_y} - S_{1_x} * m_{1,k} = Ref_1 \quad (3.5)$$

Anschließend vergleicht man Ref_1 mit $b_{1,k}$.

Fall 1 bzw. 2 kann man umgangssprachlich mit der Spieler S_1 liegt „unter“ bzw „über“ der Geraden $g_{1,k}$ bezeichnen.

Entweder Fall 1 : $Ref_1 < b_{1,k}$ oder Fall 2: $Ref_1 > b_{1,k}$ entsteht.

$$INT_{i,j_y} - INT_{i,j_x} * m_{1,k} = Ref_{i,j} \quad (3.6)$$

Löschen: Tritt Fall 1 ein, dann löscht man den Schnittpunkt falls $Ref_{i,j} > b_{1,k}$.

Tritt Fall 2 ein, dann löscht man den Schnittpunkt falls $Ref_{i,j} < b_{1,k}$.

Dies wiederholt man mit allen Schnittpunkten und mit allen Geraden. Also iteriert man erst über i, j und dann springt man zum nächsten k und dort iteriert man wieder über i, j und so weiter.

Das heißt man überprüft zunächst alle Schnittpunkte mit der ausgewählten Geraden.

Dann wählt man eine neue Gerade aus und überprüft die restlichen Schnittpunkt usw. bis man bei allen Geraden die Schnittpunkte überprüft hat.

Anschließend bleiben nur noch Schnittpunkte übrig, die auf richtigen Geraden liegen.

Diese Geraden grenzen mit ihren Schnittpunkten den Spieler in jede Richtung ab.

Warum das Löschen richtig funktioniert.

Beweisidee:

Annahme: Es gibt einen Schnittpunkt B , der nicht gelöscht wird und nicht mindestens auf einer richtigen Gerade liegt.

Fallunterscheidung:

a) *der Schnittpunkt liegt außerhalb der eingegrenzten Fläche von den richtigen Geraden*

Dies kann nicht passieren, denn dann gibt es mindestens eine Gerade, die den Schnittpunkt vom Spieler trennt und somit würde der Schnittpunkt gelöscht werden.

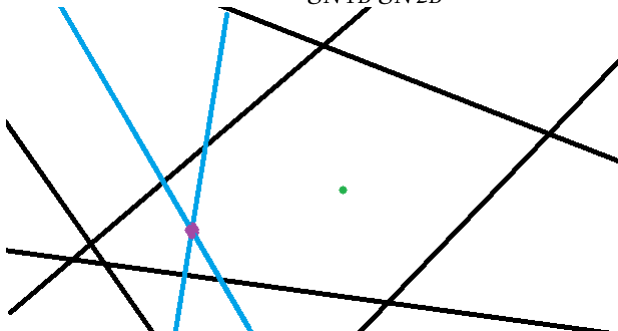
b) *der Schnittpunkt liegt innerhalb der eingegrenzten Fläche von den richtigen Geraden*

Somit existieren mindestens zwei Geraden, die sich in dem Schnittpunkt schneiden. Diese Geraden müssen sich aber auch mit den richtigen Geraden schneiden. Da die Fläche beschränkt ist und die Geraden unendlich lang sind.

Zwei Geraden in einem zweidimensionalen Raum schneiden sich einmal, sofern die Steigungen der Geraden nicht gleich sind. Da aber mehr als eine Gerade die Fläche definieren muss, gibt es bei parallelen Geraden mit den anderen Geraden somit auch Schnittpunkte. Somit existieren zwei Schnittpunkte N_1, N_2 , die jeweils einem Index von dem Schnittpunkt B und einen Index von einer richtigen Gerade haben.

Die zwei Geraden g_{N_1B}, g_{N_2B} , die N_1, N_2 und B verbinden, trennen somit etwas von der relevanten Fläche ab, da B innerhalb der relevanten Fläche liegt. Somit würde man mindestens einen Schnittpunkt außerhalb haben. Es entsteht wieder Fall a). Widerspruch

An einem Beispiel sieht man es eigentlich direkt. Hier ist der violette Punkt B und die blauen Geraden sind g_{N_1B}, g_{N_2B} .



3.2.7 Zusammenfügen von doppelten Punkten

Es kann noch vorkommen, dass es Schnittpunkte gibt, welche die selben Koordinaten haben aber mehrere Kombinationen von Geraden ID's. Meistens tritt dieser Fall im Doppelpack auf. Dafür erstellt man eine Liste für den Punkt und fügt dieser Liste alle Geraden ID's hinzu.

3.2.8 Die richtigen Schnittpunkte

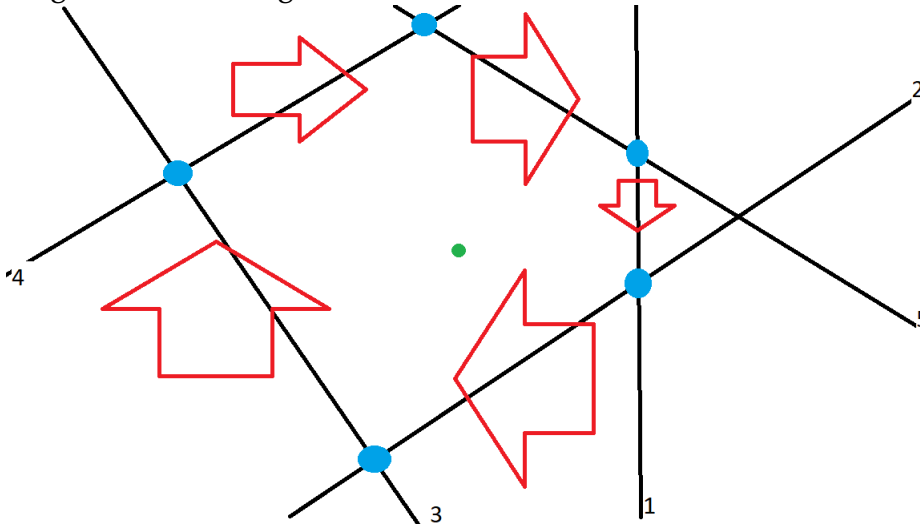
Nun wählt man einen beliebigen Schnittpunkt.

Bei dem Punkt starten wir. Der Punkt hat noch die Informationen von den Indizes auf welchen Geraden er liegt. Man wählt einen Index des Start Schnittpunktes aus und schaut nach ob dieser Index bei einem anderen Punkt existiert.

Falls nicht, geht man zu einem anderen Index.

Falls ein Schnittpunkt mit dem selben Index existiert wählt man dann diesen aus. Dieses Verfahren iteriert man so lange bis man zum Start Schnittpunkt wieder ankommen ist.

Folgende Zeichnung sollte die Methodik verständlich machen.



Es existieren nur noch die blau eingefärbten Schnittpunkte. Der grüne Punkt stellt den Spieler 1 dar.

3.2.9 Flächenberechnung

Nun zerlegt man dieses n-Eck in n Dreiecke, indem man jeweils ,mit zwei durch einer Geraden verbunden, Schnittpunkten und den Spieler 1 („Mittelpunkt“ des n-Ecks) ein Dreieck bildet.

Mit Satz 3 kann man dann sehr einfach die Fläche jedes Dreiecks berechnen und somit die Fläche des n-Ecks.

Einschränkung der Spielfläche mit einer Trennlinie

Wie aber schon in dem Kapitel Erklärung genannt, möchte man nur die letzten $\frac{2}{3}$ in der x-Achse von der Gesamtfläche betrachten. Die vorgegebene Fläche beinhaltet somit das verteidigende Tor und von dort aus gesehen die an grenzende $\frac{2}{3}$ Fläche der Gesamtfläche. Die relevante Fläche ist im Bereich $[0,70]$ oder $[35,105]$, je nachdem auf welche Seite gespielt. Für die Berechnung mit Trennlinie erstellt man eine Fallunterscheidung.

Fall 1: Das n-Eck liegt nicht in der vorgegebenen Fläche. Dies passiert falls folgendes gilt:

$$\forall INT_{i,j} \in \{\text{Lösungsmenge des n-Ecks}\} \text{ gilt } INT_{i,j_x} > 70 \text{ oder } INT_{i,j_x} < 35 \quad (3.7)$$

In diesem Fall kann man die Fläche ignorieren.

Fall 2: Das n-Eck liegt komplett in der vorgegebenen Fläche. Dies passiert falls folgendes gilt:

$$\forall INT_{i,j} \in \{\text{Lösungsmenge des n-Ecks}\} \text{ gilt } INT_{i,j_x} < 70 \text{ oder } INT_{i,j_x} > 35 \quad (3.8)$$

Hier kann man die Fläche mit der vorherigen Methode berechnen.

Fall 3: Dieser Fall wird nur für die vorgegebene Fläche von $[0,70]$ dargestellt, für $[35,105]$ geht man analog vor.

a) Falls ein n-Eck auf der Trennlinie liegt und der Spieler sich innerhalb der vorgegebenen Fläche befindet. Dies passiert, falls folgendes gilt:

$$\begin{aligned} &\exists INT_{i,j} \in \{\text{Lösungsmenge des n-Ecks}\} \text{ mit } INT_{i,j_x} \in [0,70] \text{ und} \\ &\exists INT_{i,j} \in \{\text{Lösungsmenge des n-Ecks}\} \text{ mit } INT_{i,j_x} \in (70,105] \text{ und} \\ &S_{1_x} \in [0,70] \end{aligned}$$

Nun sucht man zwei Geraden. Auf diesen Geraden sollen zwei Schnittpunkte existieren, jeweils ein Schnittpunkt liegt innerhalb der relevanten Fläche und einer außerhalb der relevanten Fläche. Mit den gefundenen zwei Geraden berechnet man neue Schnittpunkte. Diese Schnittpunkte sind $(70, Y\text{-Wert von der Gerade IA1})$ und $(70, Y\text{-Wert von der Gerade IA2})$ wobei der Y-Wert durch $70 * m_{1,s} + b_{1,s} = y_{Wert}$ berechnet werden. IA soll nur bezeichnen das die Gerade einen Punkt innerhalb und einen außerhalb hat.

Jetzt schließt man alle Eckpunkte, die im x-Wert außerhalb von $[0,70]$ liegen, aus. Folglich definiert man $InnerINT_{i,j} = \{INT_{i,j} : INT_{i,j} \in [0,70]\}$

Mit den Punkten aus $\{(70, Y\text{-Wert von der Gerade IA}_q) ; \text{mit } q=1,2\}$ und $InnerINT_{i,j}$ erstellt man ein n-Eck, dieses teilt man wieder in Dreiecke auf und berechnet die Fläche dieser Dreiecke mit Satz 3.

b) Falls ein n-Eck auf der Trennlinie liegt und der Spieler sich außerhalb der vorgegebenen Fläche befindet. Dies passiert falls folgendes gilt:

$$\begin{aligned} &\exists INT_{i,j} \in \{\text{Lösungsmenge des n-Ecks}\} \text{ mit } INT_{i,j_x} \in [0,70] \text{ und} \\ &\exists INT_{i,j} \in \{\text{Lösungsmenge des n-Ecks}\} \text{ mit } INT_{i,j_x} \in (70,105] \text{ und} \\ &S_{1_x} \notin [0,70] \end{aligned}$$

Hier geht man ähnlich zu a) vor. Nur berechnet man nicht direkt die richtige Fläche sondern geht den Umweg über die gesamte Fläche des n-Ecks minus der Fläche des n-Ecks im falschen Bereich.

$$\text{n-Eck Gesamtfläche} = \text{n-Eck in } [0,70] + \text{n-Eck in } (70,105] \quad (3.9)$$

Die n-Eck Gesamtfläche berechnen wie vor der Fallunterscheidung erklärt. Die Fläche des n-Eck in $(70,105]$ berechnet man mit a). Man muss nur darauf achten, dass man den Bereich von $[0,70]$ zu $(70,105]$ ändern und zusätzlich dreht man die Ungleichungen in Fall1 und Fall2 um. Anschließend benutzt man Gleichung 3.9 und bekommen die relevante Fläche im richtigen Bereich.

3.2.10 Iteration

Alle Vorherigen Schritten wiederholt man noch mit allen anderen Spielern, d.h. man tauscht Spieler 1 erst gegen 2, dann 3, usw. bis man alle Spieler abgehandelt hat. Nach dem man alle Flächen der Spieler berechnet hat, addiert man diese teamweise. Zusätzlich wiederholen man alle Schritte für jede $\frac{1}{10}$ Sekunde.

3.3 Beispiel

Die Ergebnisdatei ist so groß, dass es keinen Sinn macht diese komplett in den Anlagen hinzuzufügen. Diese wäre 54000 Zeilen lang. Trotzdem findet sich ein kleiner Auszug in den Anlagen.

Um die richtigen Rückschlüsse aus diesen Daten zu ziehen, muss man die Daten individuell für jeden Spieler oder jeden Verein über einen längeren Zeitraum ansehen. Folgend wird nur eine kleine möglich Auswertungen aus der Raumkontrolle aufgezeigt. Zukunftsgerichtet wäre eine Erweiterung des Systems um Laufrichtung sowie der Geschwindigkeit der Spieler interessant.

Im folgenden Beispiel betrachte man ein Fußballspiel aus der Österreichischen Bundesliga.

Tabelle 3.1: Halbzeitunterschied

	Verteidigend	Angreifend
durchschnitt.Fläche H1 in %	69,5182959521	30,4817040479
durchschnitt.Fläche H2 in %	71,0960805365	28,9039194635
durchschnitt.Fläche Gesamt in %	70,2901944965	29,7098055035

Kapitel 4

Zusammenfassung

Die Datensätze von den Positionsdaten und den Eventdaten sind die Grundlage der Berechnung von den Key Performance Indicators. Die Positionsdaten sind eine mindestens 54000 Zeilen lange TXT-Datei in der jede Zehntelsekunde die Position von jedem Spieler genau erfasst wird. Die Eventdaten werden in einer XML-Datei gespeichert und erfassen jede Ballaktion im Spiel.

Für das Defensive Packing vereinigt man beide Dateien, um die Pässe aus der XML-Datei auszulesen und die Koordinaten der überspielten Gegner aus der TXT-Datei.

Nun definiert man sich die relevanten Pässe. Es sind die Pässe vom angreifenden Team in Richtung des Tors des verteidigenden Teams.

Man zählt alle verteidigenden Spieler und vergleicht deren Koordinaten mit den Ballkoordinaten beim Passabspiel und Passempfang, um die überspielten Gegner zu erhalten.

Aus den Koordinaten des Pass spielenden Spielers und den des Pass empfangenden Spielers berechnet man dann noch eine Gütefunktion, die die Wichtigkeit des Passes widerspiegelt.

Um potenziell gefährlichere Pässe von Anderen abzuheben, wurde die zweite Gütefunktion eingefügt. Diese zeigt an ob der angespielte Spieler in unserem definierten etwas größeren 16-Meter-Raum steht.

Bei der Raumkontrolle betrachten man nur die TXT-Datei. Man fixiert einen Spieler 1.

Anschließend erstellt man Mittelsenkrechten mit allen anderen Spielern. Diese Mittelsenkrechten schneidet man miteinander sowie mit den Randbedingungen, welche den Spielfeldrand definieren. Daraus erhält man alle Schnittpunkte. Man orientiert nun alle Geraden Richtung Spieler 1 und teilt somit den Raum in zwei Hälften. Alle Schnittpunkte in der Hälfte, in der der Spieler 1 nicht liegt, werden gelöscht.

Anschließend fügt man den Schnittpunkten die Indices der Geraden, auf denen sie liegen, hinzu. Man wählt sich einen Schnittpunkt als Startpunkt aus. Nun sucht man einen weiteren Schnittpunkt, welcher einen gleichen Index mit dem Startpunkt hat, aus. Dies iteriert man so lange bis man wieder beim Startpunkt angekommen ist.

Da man aber häufig die Raumkontrolle nicht für das gesamte Spielfeld berechnen möchte, fügt man nun eine Trennlinie, orthogonal zur x-Achse ein. Diese ist bei $66,6\%$ also beim x-Wert 70 oder 35 je nachdem auf welcher Seite gespielt wird. Man sucht zwei Geraden bei denen ein Schnittpunkt links und ein Schnittpunkt rechts von der Trennlinie liegen, diese schneiden wir noch mal mit der Trennlinie und bekommen so neue Schnittpunkte. Je nachdem wo der Spieler steht und wo die Schnittpunkte liegen machen wir eine Fallunterscheidung und erstellen verschiedene Methoden zur Berechnung der Fläche.

Das gesamte Verfahren iteriert man für jeden Spieler und danach für jede Zeitsequenz.

Ich erstellte für die Masterarbeit auch zwei Programme in Java, diese wenden die Methodik auf die Datensätze an. Die daraus gewonnen neuen Daten sind in den Anlagen angefügt. Mit den neuen Daten bekommt man mit Excel Spiel analytische interessante Daten. Einen kleinen Ausschnitt davon findet man bei den Beispielen.

Literaturverzeichnis

[Memmert] Memmert D, Raabe D

Revolution Im Profifußball Mit Big Data zur Spielanalyse 4.0 2017.

[Review] Memmert D, Rein R

R. Match analysis, Big Data and tactics: current trends in elite soccer Dtsch Z Sportmed. 2018; 69: 65-72.

[Packing] <https://www.fussballtraining.de/allgemein/packing-oder-wie-wichtig-effizientes-passspiel-im-fussball-ist/21487/>

[Packing Bild] <https://www.soccerdrills.de/magazin/trainer-wissen/artikel/packing-im-fussball/>

[Raumkontrolle] <https://www.tagesanzeiger.ch/wissen/technik/entschluesselung-des-fussballs-dank-big-data/story/30186152>

Anlagen

Ergebnisdatei Packing

Die Ausgabe für ein Spiel vom Programm Packing sieht folgendermaßen aus.

ID=PassID,Half=die Halbzeit 1 oder 2, ITime= Initialtime vom Pass,RTime=Receivingtime vom Pass, OO=Ouplayed Opponents,RO= Remaining Opponents, QF= Gütefunktion,In= Insidebox.

Tabelle 4.1: Packing

ID	Half	ITime	RTime	PlayerID1	PlayerID2	OO	RO	QF	In
1	1	42800	44160	582760	362784	2	9	3.0634	0
2	1	81080	83000	573158	582760	0	11	10.3953	0
3	1	84400	85640	582760	349309	4	7	3.9572	0
4	1	86120	86600	349309	375478	0	7	0.9799	0
5	1	90320	92000	375478	9611576	1	8	13.0656	0
6	1	148720	149680	375478	349309	0	11	0.2511	0
7	1	184600	185440	582760	362784	1	9	0.0225	0
8	1	271560	273480	369331	977414	6	3	4.8834	0
9	1	279720	280240	349309	375478	3	7	1.8876	0
10	1	381120	383000	369331	977414	5	3	1.7283	0
11	1	442960	443840	362784	375478	0	9	0.7023	0
12	1	445880	446360	375478	977414	2	6	0.2705	0
13	1	469440	470800	369331	9611576	2	8	1.7019	0
14	1	481840	485240	9608977	977414	7	4	51.4192	0
15	1	488840	489880	362784	375478	0	6	1.3034	0
16	1	495520	496480	375478	362784	0	6	0.0391	0
17	1	515000	516360	9608977	582760	0	11	14.3034	0
18	1	519440	520360	582760	1061409	1	10	7.9953	0
19	1	522480	523680	1061409	9611576	0	9	7.1743	0
20	1	564360	565720	375478	349309	5	6	1.0751	0
21	1	600358	601120	349309	9611576	2	5	7.7402	0
22	1	619160	620600	349309	362784	2	7	1.4204	0
23	1	663280	664080	349309	977414	5	4	0.2088	0
24	1	670160	670720	977414	9611576	0	8	0.0042	0
25	1	672040	674480	349309	977414	5	4	1.0148	0
26	1	675440	678160	977414	606463	2	3	1.139	0
27	1	766840	768000	362784	977414	2	8	10.3556	0

Tabelle 4.2: Packing

ID	Half	ITime	RTime	PlayerID1	PlayerID2	OO	RO	QF	In
28	1	798840	799240	9611576	369331	0	7	-0.0033	0
29	1	802320	803240	9611576	375478	6	3	14.8098	0
30	1	842960	844120	1061409	375478	3	6	7.4957	0
31	1	848000	849200	375478	369331	2	6	0.8973	0
32	1	870560	871120	1061409	375478	2	7	1.3613	0
33	1	872160	873520	1061409	349309	0	9	2.0337	0
34	1	875120	875960	349309	362784	5	4	0.1079	0
35	1	882040	883080	573158	977414	6	3	32.7598	0
36	1	893080	894080	349309	362784	1	8	0.613	0
37	1	899800	900720	582760	1061409	1	10	5.6872	0
38	1	904200	905360	582760	1061409	2	9	6.0748	0
39	1	907960	908960	1061409	9611576	3	8	14.0323	0
40	1	912640	913680	9611576	977414	4	3	2.2117	1
41	1	982200	982880	369331	606463	2	7	0.7189	0
42	1	1056458	1057498	9608977	1061409	1	10	7.6332	0
43	1	1073640	1074360	362784	375478	1	7	0.2159	0
44	1	1192898	1193858	9608977	573158	0	11	0.6851	0
45	1	1203440	1205160	9608977	573158	0	11	6.7663	0
46	1	1223760	1224280	1061409	9611576	2	6	0.214	0
47	1	1225880	1226520	349309	375478	1	7	6.7901	0
48	1	1239320	1240320	582760	375478	1	8	0.0973	0
49	1	1321440	1322960	369331	977414	5	3	1.7177	0
50	1	1325600	1326240	1061409	375478	2	6	1.0499	0
51	1	1331600	1334360	349309	369331	8	1	9.039	1
52	1	1369040	1369880	369331	1061409	2	9	1.0142	0
53	1	1395480	1396760	362784	375478	2	8	0.0976	0
54	1	1398320	1400280	349309	606463	5	4	34.6693	0
55	1	1413240	1414520	362784	606463	4	7	4.0333	0
56	1	1449120	1450880	9608977	582760	0	11	1.5218	0
57	1	1453960	1455280	362784	977414	4	6	4.7631	0
58	1	1469080	1469960	362784	349309	1	8	0.9766	0
59	1	1543458	1547638	9608977	977414	6	5	42.5092	0
60	1	1552400	1553520	349309	375478	1	7	0.5329	0
61	1	1555760	1556640	362784	349309	1	8	2.8212	0
62	1	1566880	1569400	573158	582760	1	10	2.3246	0
63	1	1594018	1595360	9608977	573158	0	9	4.172	0
64	1	1601120	1604120	573158	375478	1	7	24.6267	0
65	1	1626800	1627520	1061409	375478	3	6	3.1273	0
66	1	1629160	1629600	1061409	9611576	0	9	1.9128	0
67	1	1678040	1679080	362784	349309	0	9	2.9546	0
68	1	1681560	1683240	582760	362784	2	9	11.2424	0
69	1	1684640	1685680	362784	977414	4	5	21.5102	0
70	1	1746560	1749560	573158	362784	1	10	27.4959	0
71	1	1752320	1753040	362784	9611576	2	8	0.3986	0
72	1	1759320	1759720	582760	9611576	2	9	0.1436	0
73	1	1765920	1768120	9608977	573158	0	11	37.083	0
74	1	1769480	1771400	573158	369331	0	11	8.9284	0

Tabelle 4.3: Packing

ID	Half	ITime	RTime	PlayerID1	PlayerID2	OO	RO	QF	In
75	1	1776000	1777440	369331	349309	1	9	6.7542	0
76	1	1781240	1782680	1061409	582760	0	11	4.838	0
77	1	1784920	1785880	582760	349309	2	9	14.7896	0
78	1	1794560	1795720	582760	362784	2	9	1.0265	0
79	1	1808080	1808880	362784	9611576	1	7	0.4707	0
80	1	1825720	1828080	9608977	573158	0	11	6.3178	0
81	1	1835320	1836440	369331	1061409	1	9	3.8463	0
82	1	1840040	1841160	1061409	9611576	0	10	3.8233	0
83	1	1842160	1843520	9611576	362784	4	6	2.0021	0
84	1	1850480	1851040	369331	375478	3	4	0.9488	0
85	1	1943760	1944600	349309	977414	3	5	1.5167	0
86	1	1972800	1973440	606463	375478	-1	5	0.1265	0
87	1	1976400	1977320	349309	9611576	3	6	8.9584	0
88	1	1982000	1982680	1061409	375478	2	7	4.0282	0
89	1	1986160	1986920	1061409	349309	2	8	6.4668	0
90	1	1992760	1994200	582760	362784	0	9	3.7531	0
91	1	2051520	2052040	349309	977414	4	5	0.1189	0
92	1	2258520	2259480	977414	606463	1	4	14.3179	0
93	1	2361298	2363720	9608977	369331	1	10	6.6875	0
94	1	2374480	2376240	582760	349309	0	11	9.2462	0
95	1	2379960	2381000	349309	362784	3	6	0.1411	0
96	1	2434360	2436120	582760	349309	2	8	0.1168	0
97	1	2467858	2471398	9608977	977414	7	4	89.5219	0
98	1	2481520	2483400	573158	362784	1	10	9.4759	0
99	1	2484560	2486840	362784	9611576	8	2	2.5184	0
100	1	2497560	2498440	362784	977414	4	5	0.0456	0
101	1	2499520	2500720	349309	9611576	3	4	2.3172	0
102	1	2502040	2502720	9611576	375478	1	4	2.6584	1
103	1	2564640	2565520	369331	1061409	0	9	0.1646	0
104	1	2573840	2575520	9608977	362784	0	11	14.4772	0
105	1	2590840	2591840	349309	977414	2	4	6.8013	0
106	1	2604600	2605240	362784	349309	0	8	0.097	0
107	1	2706240	2708000	9608977	582760	1	10	1.6231	0
108	1	2768000	2769120	369331	977414	4	5	0.3028	0
109	1	2782360	2783120	362784	606463	3	4	4.5465	1
110	1	2786520	2788760	9611576	369331	1	8	9.6932	0
111	1	2790160	2790720	369331	375478	0	7	4.3976	1
112	1	2866800	2868400	573158	375478	2	8	0.4312	0
113	1	2872960	2873960	349309	9611576	1	7	9.598	0
114	1	2894520	2895880	9611576	606463	0	5	14.2844	0
115	1	2911280	2912720	362784	349309	1	8	0.0876	0
116	1	2914280	2915800	349309	1061409	0	8	2.1216	0
117	2	40	240	606463	977414	2	9	0.075	0
118	2	960	2360	977414	9611576	0	9	4.6142	0
119	2	2880	4280	9611576	606463	3	6	2.8537	0
120	2	23500	26380	9608977	573158	1	10	5.7735	0
121	2	54340	55860	573158	362784	2	9	0.6363	0
122	2	146620	148420	362784	1061409	2	8	3.7475	0

Tabelle 4.4: Packing

ID	Half	ITime	RTime	PlayerID1	PlayerID2	OO	RO	QF	In
123	2	156100	157460	362784	9611576	3	7	16.4072	0
124	2	160420	164820	375478	369331	1	6	10.189	0
125	2	240780	241580	362784	1061409	0	10	1.1849	0
126	2	244940	245660	362784	375478	3	5	4.1887	0
127	2	307340	308940	362784	977414	4	4	0.7503	0
128	2	347667	349107	362784	606463	2	6	6.8794	0
129	2	352867	353700	362784	375478	4	5	9.4911	0
130	2	376660	378100	349309	375478	1	7	0.9709	0
131	2	379660	380500	362784	349309	2	7	1.7622	0
132	2	383220	384780	582760	362784	2	9	1.0767	0
133	2	441687	445547	9608977	977414	6	5	56.4243	0
134	2	460740	462420	9608977	573158	0	11	7.7587	0
135	2	466940	468940	573158	606463	5	5	51.8822	0
136	2	476540	479700	582760	349309	0	8	3.4983	0
137	2	491180	492620	375478	362784	1	7	0.862	0
138	2	493820	495260	362784	9611576	0	7	1.0473	0
139	2	525660	527300	582760	362784	1	9	3.2792	0
140	2	543300	543860	1061409	349309	2	8	4.5533	0
141	2	545980	546540	9611576	349309	0	8	1.8606	0
142	2	579220	579940	369331	1061409	0	10	1.777	0
143	2	582180	584580	582760	606463	7	4	10.5951	0
144	2	585420	586540	606463	9611576	3	3	0.6844	0
145	2	723727	725007	362784	349309	2	9	1.0369	0
146	2	743620	746740	9608977	977414	6	5	43.0014	0
147	2	764947	765900	573158	1061409	1	9	4.1035	0
148	2	766700	767660	1061409	349309	1	9	5.9397	0
149	2	774300	775980	573158	606463	4	7	49.3408	0
150	2	778060	779340	349309	362784	0	9	2.1879	0
151	2	780260	781220	362784	375478	4	5	10.2973	0
152	2	832060	833660	369331	362784	3	6	0.0698	0
153	2	874727	876167	369331	1061409	0	10	3.7655	0
154	2	878647	880567	573158	369331	2	9	1.1386	0
155	2	987447	988327	9608977	582760	0	11	1.1879	0
156	2	1009180	1010660	9611576	977414	4	3	12.67	0
157	2	1034140	1035060	362784	596555	1	6	0.3899	0
158	2	1088540	1089660	362784	977414	1	7	1.0373	0
159	2	1091140	1092060	362784	977414	2	6	0.2069	0
160	2	1161300	1162100	1061409	596555	3	5	0.8062	0
161	2	1189340	1190580	362784	596555	8	2	26.7186	0
162	2	1208900	1210780	573158	375478	3	7	16.9916	0
163	2	1377700	1378220	369331	9611576	1	7	0.0357	0
164	2	1392220	1393220	369331	1061409	2	8	0.7662	0
165	2	1501227	1502027	369331	9611576	0	8	0.2736	0
166	2	1502507	1503067	9611576	369331	0	8	0.0065	0
167	2	1523860	1524660	1061409	375478	3	5	1.6325	0
168	2	1590500	1591420	369331	973044	1	8	1.4256	0
169	2	1632380	1633420	362784	375478	0	7	8.6475	0
170	2	1634500	1635060	375478	973044	2	5	5.0314	0

Tabelle 4.5: Packing

ID	Half	ITime	RTime	PlayerID1	PlayerID2	OO	RO	QF	In
171	2	1689820	1690580	349309	375478	2	7	0.5666	0
172	2	1697980	1699620	573158	596555	5	6	32.4875	0
173	2	1703180	1704140	375478	362784	2	6	0.537	0
174	2	1710820	1712580	573158	369331	1	10	14.7386	0
175	2	1715340	1716380	369331	973044	3	7	1.7969	0
176	2	1907380	1908820	9608977	1061409	3	7	2.6307	0
177	2	1917180	1918620	1061409	362784	0	8	0.0881	0
178	2	1930500	1932940	582760	369331	0	11	8.6467	0
179	2	1937220	1938140	369331	596555	0	8	0.2031	0
180	2	1949700	1951140	582760	1061409	0	11	1.9944	0
181	2	1956140	1956660	1061409	375478	1	9	1.4494	0
182	2	1960540	1962220	582760	573158	0	11	18.9829	0
183	2	1964260	1965300	573158	596555	3	8	1.8297	0
184	2	1976860	1978140	582760	362784	0	11	1.1936	0
185	2	1979220	1981780	362784	596555	10	1	0.6264	1
186	2	1986660	1988740	349309	596555	3	4	4.4761	0
187	2	2084340	2087940	9608977	596555	8	3	40.7014	0
188	2	2118687	2119500	369331	1061409	0	9	1.0857	0
189	2	2131660	2132580	369331	1061409	2	7	0.7047	0
190	2	2133740	2134780	1061409	596555	2	5	2.1844	0
191	2	2171060	2171700	369331	596555	3	3	1.1557	0
192	2	2270260	2271540	573158	828351	3	5	0.3429	0
193	2	2289540	2292980	9608977	596555	7	4	110.378	0
194	2	2336460	2337100	362784	375478	2	8	0.4876	0
195	2	2390740	2392980	973044	573158	0	11	8.2215	0
196	2	2404860	2406620	582760	973044	1	10	6.2121	0
197	2	2409340	2410140	1061409	349309	1	7	3.4757	0
198	2	2410860	2412020	349309	973044	0	8	0.8314	0
199	2	2420460	2421300	349309	375478	2	7	1.0578	0
200	2	2501820	2503500	362784	596555	5	4	6.6454	0
201	2	2527620	2528260	582760	1061409	0	7	0.7146	0
202	2	2529500	2530820	1061409	349309	0	7	9.364	0
203	2	2531780	2533860	349309	369331	0	7	2.9189	0
204	2	2539380	2541700	369331	828351	3	2	0.4266	0
205	2	2562660	2564660	9608977	582760	0	11	18.6878	0
206	2	2569500	2571340	1061409	573158	2	9	5.3139	0
207	2	2572540	2576460	573158	596555	9	1	18.0051	1
208	2	2596260	2597780	362784	375478	1	7	9.4922	0
209	2	2605780	2606500	1061409	349309	3	7	1.8884	0
210	2	2660727	2661767	369331	349309	1	8	0.3427	0
211	2	2684100	2685180	369331	596555	4	4	0.8108	0
212	2	2705100	2706100	369331	349309	0	9	0.2008	0
213	2	2711300	2713820	573158	973044	2	9	27.2881	0
214	2	2727260	2727940	349309	375478	7	1	3.3312	1
215	2	2796967	2800587	9608977	596555	6	4	21.831	0
216	2	2851380	2853820	369331	362784	4	6	20.407	0
217	2	2873420	2873860	573158	349309	1	7	0.7402	0

Ergebnisdatei Raumkontrolle

Die Ausgabe für ein Spiel vom Programm Raumkontrolle sieht folgendermaßen aus. AreaHome, AreaAway bezeichnen die Fläche die das Heim- bzw Auswärts-Team kontrolliert. PHome, PAway bezeichnen die prozentuale Fläche die das Heim- bzw Auswärts-Team kontrolliert.

Die nachfolgende Tabelle zeigt nur die Auswertung der ersten 3 Sekunden eines Spiels an. Bei einem gesamten Spiel hätte man mindestens 54000 Zeilen, dies wäre zu viel um es dieser Arbeit hinzuzufügen.

Tabelle 4.6: Raumkontrolle

Time	AreaHome	AreaAway	PHome	PAway
0	3855,71957433	904,2805044447	81,0025107252	18,9974892748
100	3857,780774535	902,219072522	81,0458171951	18,9541828049
200	3859,9595086	900,0404380335	81,0915872243	18,9084127757
300	3862,314222245	897,685984442	81,1410515659	18,8589484341
400	3864,89647374	895,1036657499	81,1953016908	18,8046983092
500	3867,667608005	892,3322166292	81,2535241701	18,7464758299
600	3870,66594562	889,3335388496	81,3165202696	18,6834797304
700	3873,75906595	886,2408569517	81,3814943003	18,6185056997
800	3876,58492924	883,4150618616	81,4408600102	18,5591399898
900	3878,04839433	881,951614305	81,4716047751	18,5283952249
1000	3876,09010068	883,9093307831	81,4304740261	18,5695259739
1100	3863,770884585	896,2286754535	81,1716647418	18,8283352582
1200	3831,38425012	928,616262805	80,4912570853	19,5087429147
1300	3832,9639956	927,0391290468	80,5244008298	19,4755991702
1400	3835,244823815	924,7549078258	80,5723747907	19,4276252093
1500	3838,27135149	921,7285394847	80,6359546093	19,3640453907
1600	3841,509760085	918,4899910822	80,7039907753	19,2960092247
1700	3844,66951957	915,3305332438	80,770367162	19,229632838
1800	3846,66376012	913,3361450229	80,8122654785	19,1877345215
1900	3847,74832076	912,2518711868	80,8350454958	19,1649545042
2100	3846,902808145	913,0973933831	80,8172824638	19,1827175362
2200	3847,02567004	912,9742631946	80,8198681513	19,1801318487
2300	3849,69161043	910,3085310773	80,8758717644	19,1241282356
2400	3854,045642885	905,9544706328	80,9673435078	19,0326564922
2500	3859,4250006206	900,5747589572	81,0803612512	18,9196387488
2600	3862,8136699003	897,1863801557	81,1515468336	18,8484531664
2700	3862,7010347066	897,2988445532	81,1491834598	18,8508165402
2800	3860,9033311319	899,0968152846	81,1114120246	18,8885879754
2900	3859,6819795914	900,3277685018	81,0855898171	18,9144101829
3000	3860,5762632352	899,4237737943	81,1045427143	18,8954572857