

# **Entwicklung und Validierung eines Gesamtsystems zur Verkehrserfassung basierend auf Luftbildsequenzen**

## **DISSERTATION**

**zur Erlangung des akademischen Grades**

**doctor rerum naturalium  
(Dr. rer. nat.)  
im Fach Informatik**

**eingereicht an der  
Mathematisch-Naturwissenschaftlichen Fakultät II  
Humboldt-Universität zu Berlin**

**von  
M. Sc. Karsten Kozempel**

**Präsident der Humboldt-Universität zu Berlin:  
Prof. Dr. Jan-Hendrik Olbertz**

**Dekan der Mathematisch-Naturwissenschaftlichen Fakultät II:  
Prof. Dr. Elmar Kulke**

**Gutachter:**

1. Prof. Dr. rer. nat. Ralf Reulke
2. Prof. Dr.-Ing. Stefan Hinz
3. Prof. Dr. sc. nat. Hans-Dieter Burkhard, i. R.

**eingereicht am:** 20.10.2011

**Tag der mündlichen Prüfung:** 10.02.2012

## **Abstract**

This dissertation should make a contribution to the further development of airborne traffic detection. The used hardware is an airborne camera system combined with an inertial measurement unit for orientation determination. Mainly computer vision algorithms are presented, which are applied afterwards the image acquisition up to the determination of the most important traffic data.

After a short presentation of the used hardware the calibration of the camera's alignment angles during test flights is explained and its accuracy is analyzed. It is shown that the orientation data doesn't reach the specified accuracy, which is fortunately less important for traffic detection.

After the image preparation, which contains the ortho image generation as well as the clipping of traffic areas, a two-stage vehicle detection algorithm is implemented, which at first rapidly creates hypotheses based on edge filters. In the second stage those hypotheses are verified by a *Support Vector Machine* which rejects most of the False Positives. At good conditions the detection reaches completeness rates of up to 90 percent with a low contingent of *FP* detections.

Subsequently a tracking algorithm based on *singular value decomposition* is applied to associate vehicle hypotheses in adjacent images and determine the average speed. The achieved velocities differ less than ten kph from the manually obtained data.

Concluding an alternative orientation method is presented, that automatically determines the airplane's attitude based on GPS and image information. This is realized by extraction and matching of street segments and additional tracking of ground control points. The results have accuracies of around 0.1 to 0.2 degrees.

The innovation of this thesis refers to the combination of rapid pre-selection and more reliable verification of object hypotheses. Additionally the automatic extraction of streets as landmarks for orientation determination and its fusion with relative orientation parameters as stabilization are novel.

## **Zusammenfassung**

Die vorliegende Dissertation soll einen Beitrag zur Weiterentwicklung der luftgestützten Verkehrslage erfassung leisten. Als Plattform dafür dient ein flugzeugge tragenes Kamerasystem, welches mit einem Fluglage erfassungssystem gekoppelt ist. Vorgestellt werden hauptsächlich bildverarbeitende Algorithmen, welche an die Bildaufnahme anschließend bis hin zur Ermittlung der wichtigsten verkehrstechnischen Kenngrößen zum Einsatz kommen.

Nach kurzer Skizzierung der verwendeten Hardware wird zunächst die Kalibrierung der Kameraeinbauwinkel durch Testflüge erläutert und auf ihre Genauigkeit hin untersucht. Es wird gezeigt, dass die Orientierungsdaten nicht die vom Hersteller angegebene Genauigkeit erreichen, was jedoch für die Verkehrslage erfassung nur von geringer Bedeutung ist.

Anschließend an die Bildaufbereitung, welche die Orthobildgenerierung sowie die Eingrenzung der verkehrsaktiven Flächen beinhaltet, wird zur Ermittlung der Fahrzeudichte ein zweistufiger Fahrzeugerkennungs algorithmus entwickelt, welcher zunächst auf Kantenfilterbasis möglichst schnell Hypothesen erstellt. Diese werden in einer zweiten Phase durch eine *Support Vector Machine* überprüft, wobei ein Großteil der Fehlhypothesen verworfen wird. Die Erkennung erreicht bei guten Voraussetzungen Vollständigkeiten bis zu 90 Prozent bei sehr geringem Anteil von Fehldetections.

Anschließend wird ein auf Singulärwertzerlegung basierender Tracking-Algorithmus verwendet, um Fahrzeughypothesen in benachbarten Bildern zu assoziieren und die mittleren Geschwindigkeiten zu ermitteln. Die erhaltenen Geschwindigkeiten unterscheiden sich um weniger als zehn km/h von den manuell erhobenen.

Abschließend wird eine alternative Orientierungsmethode vorgestellt, welche auf Basis von GPS-Positionen und Bildinformationen automatisch die Fluglage ermittelt. Dies geschieht durch die Extraktion und das Matching von Straßensegmenten sowie zusätzliche Passpunktverfolgung. Die Ergebnisse weisen Genauigkeiten von etwa 0,1 bis 0,2 Grad auf.

Der Anspruch der Arbeit besteht bei der Fahrzeugerkennung in der Kombination aus schneller Vorselektion und zuverlässiger Verifizierung der Objekthypothesen. Weiterhin sind die automatische Extraktion von Straßen als Landmarken zur Orientierungsbestimmung und die Fusion mit relativen Orientierungsparametern zur Stabilisierung neuartig.



# Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>1</b>
<b>2 Aufnahmesysteme</b>	<b>5</b>
2.1 <i>ANTAR</i> . . . . .	5
2.2 <i>ARGOS</i> . . . . .	5
<b>3 Kalibrierung</b>	<b>9</b>
3.1 Koordinatensysteme . . . . .	9
3.2 Innere Orientierung . . . . .	11
3.3 Bestimmung der Einbauwinkel . . . . .	13
3.4 Alternative Kalibration bei <i>ARGOS</i> . . . . .	20
<b>4 Vorverarbeitung</b>	<b>21</b>
4.1 Georeferenzierung . . . . .	21
4.1.1 GeoTIFF . . . . .	24
4.1.2 Genauigkeit der Orthobilder . . . . .	25
4.2 Straßenprojektion . . . . .	25
4.2.1 Straßenbreite . . . . .	26
4.2.2 Genauigkeit der Straßenprojektion . . . . .	26
4.2.3 Straßenkorrektur . . . . .	27
<b>5 Fahrzeugerkennung</b>	<b>29</b>
5.1 Ansatz . . . . .	30
5.1.1 Phase eins - Erstellung von Objekthypothesen . . . . .	30
5.1.2 Phase zwei - Validierung der Hypothesen . . . . .	34
5.2 Fehlerursachen . . . . .	38
5.2.1 False Negatives . . . . .	38
5.2.2 False Positives . . . . .	38
5.3 Ergebnisse . . . . .	40
5.3.1 Datensatz 1 - Autobahn A8 . . . . .	41
5.3.2 Datensatz 2 - Autobahn A8, Stau . . . . .	41
5.3.3 Datensatz 3 - Autobahn A8, Schnee . . . . .	42
5.3.4 Datensatz 4 - Stadtgebiet München, Nebenstraßen . . . . .	42
5.3.5 Datensatz 5 - Stadtgebiet München, Nebenstraßen . . . . .	42
5.4 Zusammenfassung . . . . .	43

## *Inhaltsverzeichnis*

<b>6 Tracking</b>	<b>51</b>
6.1 Ansatz . . . . .	51
6.2 Ergebnisse . . . . .	53
6.2.1 Erste Tests . . . . .	53
6.2.2 Ergebnisse . . . . .	56
6.3 Zusammenfassung . . . . .	57
<b>7 Optische Orientierung</b>	<b>63</b>
7.1 Absolute optische Orientierung . . . . .	64
7.1.1 Extraktion von Straßenbereichen . . . . .	65
7.1.2 Straßenmatching . . . . .	65
7.1.3 Ergebnisse . . . . .	68
7.1.4 Fehlerquellen . . . . .	69
7.2 Relative optische Orientierung . . . . .	73
7.2.1 Interest Points . . . . .	73
7.2.2 Ausgleichung . . . . .	74
7.2.3 Ergebnisse . . . . .	76
7.3 Navigationsfilter . . . . .	77
7.3.1 Kalman-Filter . . . . .	78
7.3.2 Anpassung des Filters . . . . .	82
7.3.3 Ergebnisse . . . . .	87
7.3.4 Schwachstellen und Postprocessing . . . . .	88
7.4 Zusammenfassung . . . . .	91
<b>8 Zusammenfassung und Ausblick</b>	<b>95</b>

# Abbildungsverzeichnis

2.1	Kameras und IMU des <i>ANTAR</i> -Systems . . . . .	6
2.2	Schematischer Aufbau des <i>ANTAR</i> -Systems (Quelle: [53]) . . . . .	6
2.3	3K-Kamera des <i>ARGOS</i> . . . . .	7
2.4	Schematischer Aufbau des <i>ARGOS</i> . . . . .	7
3.1	Verwendete Koordinatensysteme . . . . .	10
3.2	Verwendete Koordinatensysteme . . . . .	11
3.3	Kalibrierwand und innere Orientierung . . . . .	13
3.4	ClickTool mit Aufnahme aus dem <i>ANTAR</i> -System . . . . .	14
3.5	Software zur Ermittlung der Boresightwinkel (RMS-Fehler rot markiert) .	17
3.6	Disparitäten zweier aufeinanderfolgender Aufnahmen (weiße Striche) . .	18
3.7	Ermittelte Boresightwinkel für verschiedene Aufnahmen . . . . .	18
3.8	RMS-Werte bei verschiedenen zeitlichen Offsets . . . . .	19
3.9	Kalibrierung mit drei Aufnahmen (rot - ohne Boresightwinkel, schwarz - mit korrekten Boresightwinkeln, Quelle: [63]) . . . . .	20
4.1	Iterative Höhenbestimmung (Quelle: [73]) . . . . .	22
4.2	Projektion bei der GeoTIFF-Erzeugung . . . . .	23
4.3	GeoTIFF der DLR-Messstrecke „Ernst-Ruska-Ufer“ in Berlin-Adlershof (Aufgenommen mit <i>ANTAR</i> ) . . . . .	24
4.4	Passpunkte in GeoTIFF-Bildern, grün - wahre Position, rot - projizierte Position, links - RMS = 1,6 Pixel (30 cm), rechts - RMS = 7,5 Pixel (150 cm) . . . . .	25
4.5	Fehler in der Straßendatenbank (links falsche Spuranzahl, rechts falsche Position) . . . . .	27
4.6	Straßenprojektion unkorrigiert (Straße - rot, Umgebung - türkis) . . . .	28
4.7	Straßenprojektion korrigiert . . . . .	28
5.1	GeoTIFF mit Straßensegment . . . . .	31
5.2	Straßensegment . . . . .	32
5.3	Filter für die vier Fahrzeugkanten . . . . .	32
5.4	Die vier Filterantworten . . . . .	33
5.5	Die vier Filterantworten (nach Schwellwert) . . . . .	33
5.6	UND-Verknüpfung der Filterantworten . . . . .	34
5.7	Kennzahlen des <i>SURF</i> -Deskriptors, Quelle [13] . . . . .	35
5.8	Merkmalsraum der SVM, Quelle [71] . . . . .	36
5.9	Merkmalsraum der SVM, Quelle [71] . . . . .	37

## Abbildungsverzeichnis

5.10 Ursachen für Erkennungsfehler (blau - wirkliche Fahrzeuge, rot und gelb - erkannte Fahrzeuge) . . . . .	39
5.11 Ursachen für Erkennungsfehler (blau - wirkliche Fahrzeuge, rot und gelb - erkannte Fahrzeuge) . . . . .	39
5.12 Kontrastproblem bei der Fahrzeugerkennung . . . . .	40
5.13 Detektionsraten Datensatz 1 . . . . .	45
5.14 Detektionsraten Datensatz 1 . . . . .	45
5.15 Detektionsraten Datensatz 2 . . . . .	46
5.16 Detektionsraten Datensatz 2 . . . . .	46
5.17 Detektionsraten Datensatz 3 . . . . .	47
5.18 Detektionsraten Datensatz 3 . . . . .	47
5.19 Detektionsraten Datensatz 4 . . . . .	48
5.20 Detektionsraten Datensatz 4 . . . . .	48
5.21 Detektionsraten Datensatz 5 . . . . .	49
5.22 Detektionsraten Datensatz 5 . . . . .	49
5.23 Flug- und Rechenzeiten der Fahrzeugerkennung . . . . .	50
 6.1 Vollständigkeit des Trackings in Abhängigkeit von $\sigma$ und $\gamma$ . . . . .	53
6.2 Vollständigkeit des Trackings in Abhängigkeit von $\sigma$ und $\gamma$ . . . . .	54
6.3 Zuordnungen nach Singulärwertzerlegung ( $\sigma=30$ , $\gamma=1,0$ ) und Korrelationsschwellwert (CC=0,9), rot - PKW, gelb - LKW . . . . .	58
6.4 Trackingraten bei verschiedenen Korrelationsschwellwerten (Autobahn) . .	59
6.5 Trackingraten bei verschiedenen Korrelationsschwellwerten (Stadt) . . .	59
6.6 Geschwindigkeitsabweichung in Abhängigkeit des Korrelationsschwellwertes (Autobahn) . . . . .	60
6.7 Geschwindigkeitsabweichung in Abhängigkeit des Korrelationsschwellwertes (Stadt) . . . . .	61
6.8 Oben - Wahrscheinlichkeiten der Geschwindigkeitswerte, Unten - Verteilungsdichtefunktion mit Grenzwert (rot), Quelle: [23] . . . . .	62
 7.1 Autobahn A96 in München . . . . .	66
7.2 Mittels Farbwinkel extrahierte Straßen . . . . .	67
7.3 Mittels Sättigung extrahierte Straßen . . . . .	68
7.4 Projektion der dreidimensionalen Straßenelemente in die Aufnahme . . .	69
7.5 Exrahierte (schwarz) und projizierte (rot) Straßen überlagert . . . . .	70
7.6 Projektionsgüte in Abhängigkeit der Winkel $\omega$ und $\phi$ . . . . .	70
7.7 Exrahierte (schwarz) und verbreiterte projizierte (rot) Straßen überlagert	71
7.8 Projektionsgüte in Abhängigkeit der Winkel $\omega$ und $\phi$ (verbreiterte Straßen)	71
7.9 Richtig konvergierte Aufnahmen . . . . .	72
7.10 Winkelfehler aller 400 Aufnahmen . . . . .	73
7.11 $\omega$ -Winkel der absoluten optischen Orientierung (Überflug A) . . . . .	74
7.12 $\omega$ -Winkel der absoluten optischen Orientierung (Überflug B) . . . . .	75
7.13 Winkeldifferenzen der absoluten optischen Orientierung (Überflug A) . .	76
7.14 Winkeldifferenzen der absoluten optischen Orientierung (Überflug B) . .	77

7.15	Falsch konvergierte Aufnahme aufgrund von zu wenig Straßeninformation	78
7.16	Falsch konvergierte Aufnahme aufgrund von zu viel Straßeninformation . . . . .	79
7.17	Falsch konvergierte Aufnahme aufgrund von falscher Straßenextraktion . . . . .	80
7.18	Epipolarebene zwischen zwei Aufnahmen . . . . .	81
7.19	Vektoren der <i>Interest Points</i> zwischen zwei Aufnahmen . . . . .	82
7.20	$\omega$ -Winkel der relativen optischen Orientierung (Überflug A) . . . . .	83
7.21	$\omega$ -Winkel der relativen optischen Orientierung (Überflug B) . . . . .	84
7.22	Winkeldifferenzen der relativen optischen Orientierung (Überflug A) . . . . .	85
7.23	Winkeldifferenzen der relativen optischen Orientierung (Überflug B) . . . . .	86
7.24	Ablaufdiagramm des Kalman-Filters (Quelle [101]) . . . . .	86
7.25	$\omega$ -Winkel des Navigationsfilters (Überflug A) . . . . .	88
7.26	$\omega$ -Winkel des Navigationsfilters (Überflug B) . . . . .	89
7.27	Winkeldifferenzen des Navigationsfilters (Überflug A) . . . . .	90
7.28	Winkeldifferenzen des Navigationsfilters (Überflug B) . . . . .	91
7.29	$\omega$ -Winkel des Navigationsfilters (Überflug 9) . . . . .	92
7.30	RANSAC-korrigierter $\phi$ -Winkel des Navigationsfilters (Überflug 8) . . . . .	93
1	$\omega$ -Winkel der absoluten Orientierung (Überflug 1) . . . . .	100
2	$\phi$ -Winkel der absoluten Orientierung (Überflug 1) . . . . .	100
3	$\kappa$ -Winkel der absoluten Orientierung (Überflug 1) . . . . .	101
4	Winkelfehler der absoluten Orientierung (Überflug 1) . . . . .	101
5	$\omega$ -Winkel der absoluten Orientierung (Überflug 2) . . . . .	102
6	$\phi$ -Winkel der absoluten Orientierung (Überflug 2) . . . . .	102
7	$\kappa$ -Winkel der absoluten Orientierung (Überflug 2) . . . . .	103
8	Winkelfehler der absoluten Orientierung (Überflug 2) . . . . .	103
9	$\omega$ -Winkel der absoluten Orientierung (Überflug 3) . . . . .	104
10	$\phi$ -Winkel der absoluten Orientierung (Überflug 3) . . . . .	104
11	$\kappa$ -Winkel der absoluten Orientierung (Überflug 3) . . . . .	105
12	Winkelfehler der absoluten Orientierung (Überflug 3) . . . . .	105
13	$\omega$ -Winkel der absoluten Orientierung (Überflug 4) . . . . .	106
14	$\phi$ -Winkel der absoluten Orientierung (Überflug 4) . . . . .	106
15	$\kappa$ -Winkel der absoluten Orientierung (Überflug 4) . . . . .	107
16	Winkelfehler der absoluten Orientierung (Überflug 4) . . . . .	107
17	$\omega$ -Winkel der absoluten Orientierung (Überflug 5) . . . . .	108
18	$\phi$ -Winkel der absoluten Orientierung (Überflug 5) . . . . .	108
19	$\kappa$ -Winkel der absoluten Orientierung (Überflug 5) . . . . .	109
20	Winkelfehler der absoluten Orientierung (Überflug 5) . . . . .	109
21	$\omega$ -Winkel der absoluten Orientierung (Überflug 6) . . . . .	110
22	$\phi$ -Winkel der absoluten Orientierung (Überflug 6) . . . . .	110
23	$\kappa$ -Winkel der absoluten Orientierung (Überflug 6) . . . . .	111
24	Winkelfehler der absoluten Orientierung (Überflug 6) . . . . .	111
25	$\omega$ -Winkel der absoluten Orientierung (Überflug 7) . . . . .	112
26	$\phi$ -Winkel der absoluten Orientierung (Überflug 7) . . . . .	112
27	$\kappa$ -Winkel der absoluten Orientierung (Überflug 7) . . . . .	113

## Abbildungsverzeichnis

28	Winkelfehler der absoluten Orientierung (Überflug 7) . . . . .	113
29	$\omega$ -Winkel der absoluten Orientierung (Überflug 8) . . . . .	114
30	$\phi$ -Winkel der absoluten Orientierung (Überflug 8) . . . . .	114
31	$\kappa$ -Winkel der absoluten Orientierung (Überflug 8) . . . . .	115
32	Winkelfehler der absoluten Orientierung (Überflug 8) . . . . .	115
33	$\omega$ -Winkel der absoluten Orientierung (Überflug 9) . . . . .	116
34	$\phi$ -Winkel der absoluten Orientierung (Überflug 9) . . . . .	116
35	$\kappa$ -Winkel der absoluten Orientierung (Überflug 9) . . . . .	117
36	Winkelfehler der absoluten Orientierung (Überflug 9) . . . . .	117
37	$\omega$ -Winkel der absoluten Orientierung (Überflug 10) . . . . .	118
38	$\phi$ -Winkel der absoluten Orientierung (Überflug 10) . . . . .	118
39	$\kappa$ -Winkel der absoluten Orientierung (Überflug 10) . . . . .	119
40	Winkelfehler der absoluten Orientierung (Überflug 10) . . . . .	119
41	$\omega$ -Winkel der absoluten Orientierung (Überflug 11) . . . . .	120
42	$\phi$ -Winkel der absoluten Orientierung (Überflug 11) . . . . .	120
43	$\kappa$ -Winkel der absoluten Orientierung (Überflug 11) . . . . .	121
44	Winkelfehler der absoluten Orientierung (Überflug 11) . . . . .	121
45	$\omega$ -Winkel der relativen Orientierung (Überflug 1) . . . . .	123
46	$\phi$ -Winkel der relativen Orientierung (Überflug 1) . . . . .	123
47	$\kappa$ -Winkel der relativen Orientierung (Überflug 1) . . . . .	124
48	Winkelfehler der relativen Orientierung (Überflug 1) . . . . .	124
49	$\omega$ -Winkel der relativen Orientierung (Überflug 2) . . . . .	125
50	$\phi$ -Winkel der relativen Orientierung (Überflug 2) . . . . .	125
51	$\kappa$ -Winkel der relativen Orientierung (Überflug 2) . . . . .	126
52	Winkelfehler der relativen Orientierung (Überflug 2) . . . . .	126
53	$\omega$ -Winkel der relativen Orientierung (Überflug 3) . . . . .	127
54	$\phi$ -Winkel der relativen Orientierung (Überflug 3) . . . . .	127
55	$\kappa$ -Winkel der relativen Orientierung (Überflug 3) . . . . .	128
56	Winkelfehler der relativen Orientierung (Überflug 3) . . . . .	128
57	$\omega$ -Winkel der relativen Orientierung (Überflug 4) . . . . .	129
58	$\phi$ -Winkel der relativen Orientierung (Überflug 4) . . . . .	129
59	$\kappa$ -Winkel der relativen Orientierung (Überflug 4) . . . . .	130
60	Winkelfehler der relativen Orientierung (Überflug 4) . . . . .	130
61	$\omega$ -Winkel der relativen Orientierung (Überflug 5) . . . . .	131
62	$\phi$ -Winkel der relativen Orientierung (Überflug 5) . . . . .	131
63	$\kappa$ -Winkel der relativen Orientierung (Überflug 5) . . . . .	132
64	Winkelfehler der relativen Orientierung (Überflug 5) . . . . .	132
65	$\omega$ -Winkel der relativen Orientierung (Überflug 6) . . . . .	133
66	$\phi$ -Winkel der relativen Orientierung (Überflug 6) . . . . .	133
67	$\kappa$ -Winkel der relativen Orientierung (Überflug 6) . . . . .	134
68	Winkelfehler der relativen Orientierung (Überflug 6) . . . . .	134
69	$\omega$ -Winkel der relativen Orientierung (Überflug 7) . . . . .	135
70	$\phi$ -Winkel der relativen Orientierung (Überflug 7) . . . . .	135
71	$\kappa$ -Winkel der relativen Orientierung (Überflug 7) . . . . .	136

72	Winkelfehler der relativen Orientierung (Überflug 7) . . . . .	136
73	$\omega$ -Winkel der relativen Orientierung (Überflug 8) . . . . .	137
74	$\phi$ -Winkel der relativen Orientierung (Überflug 8) . . . . .	137
75	$\kappa$ -Winkel der relativen Orientierung (Überflug 8) . . . . .	138
76	Winkelfehler der relativen Orientierung (Überflug 8) . . . . .	138
77	$\omega$ -Winkel der relativen Orientierung (Überflug 9) . . . . .	139
78	$\phi$ -Winkel der relativen Orientierung (Überflug 9) . . . . .	139
79	$\kappa$ -Winkel der relativen Orientierung (Überflug 9) . . . . .	140
80	Winkelfehler der relativen Orientierung (Überflug 9) . . . . .	140
81	$\omega$ -Winkel der relativen Orientierung (Überflug 10) . . . . .	141
82	$\phi$ -Winkel der relativen Orientierung (Überflug 10) . . . . .	141
83	$\kappa$ -Winkel der relativen Orientierung (Überflug 10) . . . . .	142
84	Winkelfehler der relativen Orientierung (Überflug 10) . . . . .	142
85	$\omega$ -Winkel der relativen Orientierung (Überflug 11) . . . . .	143
86	$\phi$ -Winkel der relativen Orientierung (Überflug 11) . . . . .	143
87	$\kappa$ -Winkel der relativen Orientierung (Überflug 11) . . . . .	144
88	Winkelfehler der relativen Orientierung (Überflug 11) . . . . .	144
89	$\omega$ -Winkel des Navigationsfilters (Überflug 1) . . . . .	146
90	$\phi$ -Winkel des Navigationsfilters (Überflug 1) . . . . .	146
91	$\kappa$ -Winkel des Navigationsfilters (Überflug 1) . . . . .	147
92	Winkelfehler des Navigationsfilters (Überflug 1) . . . . .	147
93	$\omega$ -Winkel des Navigationsfilters (Überflug 2) . . . . .	148
94	$\phi$ -Winkel des Navigationsfilters (Überflug 2) . . . . .	148
95	$\kappa$ -Winkel des Navigationsfilters (Überflug 2) . . . . .	149
96	Winkelfehler des Navigationsfilters (Überflug 2) . . . . .	149
97	$\omega$ -Winkel des Navigationsfilters (Überflug 3) . . . . .	150
98	$\phi$ -Winkel des Navigationsfilters (Überflug 3) . . . . .	150
99	$\kappa$ -Winkel des Navigationsfilters (Überflug 3) . . . . .	151
100	Winkelfehler des Navigationsfilters (Überflug 3) . . . . .	151
101	$\omega$ -Winkel des Navigationsfilters (Überflug 4) . . . . .	152
102	$\phi$ -Winkel des Navigationsfilters (Überflug 4) . . . . .	152
103	$\kappa$ -Winkel des Navigationsfilters (Überflug 4) . . . . .	153
104	Winkelfehler des Navigationsfilters (Überflug 4) . . . . .	153
105	$\omega$ -Winkel des Navigationsfilters (Überflug 5) . . . . .	154
106	$\phi$ -Winkel des Navigationsfilters (Überflug 5) . . . . .	154
107	$\kappa$ -Winkel des Navigationsfilters (Überflug 5) . . . . .	155
108	Winkelfehler des Navigationsfilters (Überflug 5) . . . . .	155
109	$\omega$ -Winkel des Navigationsfilters (Überflug 6) . . . . .	156
110	$\phi$ -Winkel des Navigationsfilters (Überflug 6) . . . . .	156
111	$\kappa$ -Winkel des Navigationsfilters (Überflug 6) . . . . .	157
112	Winkelfehler des Navigationsfilters (Überflug 6) . . . . .	157
113	$\omega$ -Winkel des Navigationsfilters (Überflug 7) . . . . .	158
114	$\phi$ -Winkel des Navigationsfilters (Überflug 7) . . . . .	158
115	$\kappa$ -Winkel des Navigationsfilters (Überflug 7) . . . . .	159

## Abbildungsverzeichnis

116	Winkelfehler des Navigationsfilters (Überflug 7) . . . . .	159
117	$\omega$ -Winkel des Navigationsfilters (Überflug 8) . . . . .	160
118	$\phi$ -Winkel des Navigationsfilters (Überflug 8) . . . . .	160
119	$\kappa$ -Winkel des Navigationsfilters (Überflug 8) . . . . .	161
120	Winkelfehler des Navigationsfilters (Überflug 8) . . . . .	161
121	$\omega$ -Winkel des Navigationsfilters (Überflug 9) . . . . .	162
122	$\phi$ -Winkel des Navigationsfilters (Überflug 9) . . . . .	162
123	$\kappa$ -Winkel des Navigationsfilters (Überflug 9) . . . . .	163
124	Winkelfehler des Navigationsfilters (Überflug 9) . . . . .	163
125	$\omega$ -Winkel des Navigationsfilters (Überflug 10) . . . . .	164
126	$\phi$ -Winkel des Navigationsfilters (Überflug 10) . . . . .	164
127	$\kappa$ -Winkel des Navigationsfilters (Überflug 10) . . . . .	165
128	Winkelfehler des Navigationsfilters (Überflug 10) . . . . .	165
129	$\omega$ -Winkel des Navigationsfilters (Überflug 11) . . . . .	166
130	$\phi$ -Winkel des Navigationsfilters (Überflug 11) . . . . .	166
131	$\kappa$ -Winkel des Navigationsfilters (Überflug 11) . . . . .	167
132	Winkelfehler des Navigationsfilters (Überflug 11) . . . . .	167
133	RANSAC-korrigierte $\omega$ -Winkel (Überflug 8) . . . . .	169
134	RANSAC-korrigierte $\phi$ -Winkel (Überflug 8) . . . . .	169
135	RANSAC-korrigierte $\kappa$ -Winkel (Überflug 8) . . . . .	170
136	RANSAC-korrigierte Winkelfehler (Überflug 8) . . . . .	170
137	RANSAC-korrigierte $\omega$ -Winkel (Überflug 9) . . . . .	171
138	RANSAC-korrigierte $\phi$ -Winkel (Überflug 9) . . . . .	171
139	RANSAC-korrigierte $\kappa$ -Winkel (Überflug 9) . . . . .	172
140	RANSAC-korrigierte Winkelfehler (Überflug 9) . . . . .	172
141	RANSAC-korrigierte $\omega$ -Winkel (Überflug 10) . . . . .	173
142	RANSAC-korrigierte $\phi$ -Winkel (Überflug 10) . . . . .	173
143	RANSAC-korrigierte $\kappa$ -Winkel (Überflug 10) . . . . .	174
144	RANSAC-korrigierte Winkelfehler (Überflug 10) . . . . .	174

# **Tabellenverzeichnis**

3.1	Genauigkeit des Applanix-Inertialsystems (Quelle: [76]) . . . . .	17
7.1	RMS-Werte der elf Überflüge. (links ungefiltert, rechts Kalman-gefiltert, falsch initialisierte Überflüge sind kursiv dargestellt) . . . . .	90
7.2	RMS-Werte der elf Überflüge (links ungefiltert, rechts Kalman- und RAN- SAC-gefiltert) . . . . .	91



# Symbolverzeichnis

Symbol	Erklärung
$\mathbf{B}_k$	Matrix A
$B_1, B_2$	Koeffizienten der Affinität und Scherung
$b$	Verschiebung einer Hyperebene zum Ursprung
$c$	Kammerkonstante, Brennweite
$\mathbf{C}$	Korrelationsmatrix
$\cos(\omega)$	Kosinus des Winkels $\omega$
cols	horizontale Anzahl der Pixel
$D$	Verkehrsdichte
$\mathbf{D}$	Diagonalmatrix bei der Singulärwertzerlegung
$d_x$	Pixelbreite
$d_y$	Pixelhöhe
$\Delta(x, y)$	Verzeichnungsdifferenz
$e$	eulersche Zahl (2,718281828459)
$E[\mathbf{x}]$	Erwartungswert des Vektors $\mathbf{x}$
$\exp(x)$	Exponentialfunktion mit Basis $e$ und Exponent $x$
$\mathbf{F}_k$	Zustandsübergangsmatrix im Zeitschritt $k$
$\mathbf{G}$	Nachbarschaftsmatrix
$\mathbf{G}_k$	Einflussmatrix des Systemrauschen im Zeitschritt $k$
$\mathbf{H}_k$	Beobachtungs- oder Messmatrix im Zeitschritt $k$
$\mathcal{H}$	Hyperebene
$\mathbf{I}$	Einheitsmatrix (identity matrix)
$I, J$	aufeinanderfolgende Aufnahmen mit zu assoziierenden Objekten
$I_i, J_j$	Objekte in den Aufnahmen $I$ und $J$
$\ I_i - J_j\ $	euklidischer Abstand der Objekte $I_i$ und $J_j$
$\mathbf{K}_k$	Kalman-Verstärkungsmatrix im Zeitschritt $k$
$K_1 - K_3$	Koeffizienten der radialsymmetrischen Verzeichnung
$\mathcal{K}$	Kernel-Funktion
$\kappa$	Rotationswinkel der $z$ -Achse (photogrammetrische Orientierung)
$N_f(x, t)$	Fahrzeuganzahl in bestimmtem Ortsintervall zu bestimmter Zeit
$\omega$	Rotationswinkel der $x$ -Achse (photogrammetrische Orientierung)
$\mathbf{P}_k$	Kovarianzmatrix des Schätzfehlers im Zeitschritt $k$
$P_1, P_2$	Koeffizienten der radial-asymmetrischen Verzeichnung
$\langle p, q \rangle$	Skalarprodukt der Vektoren $p$ und $q$
$\phi$	Rotationswinkel der $y$ -Achse (photogrammetrische Orientierung), Rotationswinkel der $x$ -Achse (IMU-Orientierung)

## Tabellenverzeichnis

$\Phi$	Projektionsfunktion
$\psi$	Rotationswinkel der $z$ -Achse (IMU-Orientierung)
$Q$	Verkehrsstärke
$\mathbf{Q}_k$	Kovarianzmatrix des Systemrauschen im Zeitschritt $k$
$\mathbf{R}_k$	Kovarianzmatrix des Messwertes im Zeitschritt $k$
$r$	Radius (Abstand eines Projektionsstrahles zur optischen Achse)
$r', r''$	Projektionsstrahlen in der Epipolarebene
$r_{ij}$	euklidischer Abstand der Objekte $I_i$ und $J_j$ , Element der Rotationsmatrix $\mathbf{R}$ mit dem Index $i, j$
$\mathbf{R}_x, \mathbf{R}_y, \mathbf{R}_z,$	Rotationsmatrizen um die Achsen $x, y, z$
$\mathbb{R}^g$	Merkmalsraum reeller Zahlen der Dimension $g$
rows	vertikale Anzahl der Pixel
$\sigma$	Standardabweichung der Gauß-Verteilung
$sign(x)$	Vorzeichen von $x$
$\sin(\omega)$	Sinus des Winkels $\omega$
$\mathbf{T}, \mathbf{U}$	Orthogonalmatrizen (bei der Singulärwertzerlegung)
$\theta$	Rotationswinkel der $y$ -Achse (IMU-Orientierung)
$\bar{v}$	mittlere Geschwindigkeit der Fahrzeuge an einem Querschnitt (zeitlicher Mittelwert)
$\bar{v}_m$	mittlere Geschwindigkeit der Fahrzeuge im Verkehrsstrom (räumlicher Mittelwert)
$w$	Normalenvektor einer Ebene
$\mathbf{x}_k$	Zustandsvektor eines Kalman-Filters im Zeitschritt $k$
$\hat{\mathbf{x}}_k^-$	a-priori-Schätzung des Zustandes im Zeitschritt $k$
$\hat{\mathbf{x}}_k$	a-posteriori-Schätzung des Zustandes im Zeitschritt $k$
$x$	Ort auf der Strecke
$X_c, Y_c, Z_c$	Kamerakoordinaten
$x_i, y_i$	Bildkoordinaten
$x, x_i$	Trainingsdatenvektor der $SVM$ (Index $i$ )
$x_{neu}$	zu klassifizierender Vektor der $SVM$
$\bar{x}_i, \bar{y}_i$	verzerrte Bildkoordinaten
$x_p, y_p$	Pixelkoordinaten
$X_p, Y_p$	Bildhauptpunkteverschiebung (in Pixel)
$\mathcal{X}$	Menge der Trainingsbeispiele einer $SVM$
$y_i$	Klasse des Vektors $x_i$
$\mathbf{y}_k$	Messwert im Zeitschritt $k$

# Abkürzungen

Abkürzung	Erklärung
AdV	Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland
ANTAR	Airborne Traffic Analyzer
AO	Absolute Orientierung
ATKIS	Amtliches Topographisch-Kartographisches Informationssystem
ARGOS	Airborne Wide Area High Altitude Monitoring System
CC	Correlation Coefficient
CCD	Charge Coupled Device (Lichtempfindlicher Sensor)
FN	False Negatives
FP	False Positives
GCP	Ground Control Point
GPS	Global Positioning System
HOG	Histogram of Oriented Gradients
HOS	Histogram of Oriented Gradients
HSV	Hue Saturation Value
IMU	Inertial Measurement Unit
NAVTEQ	Navigation Technologies
RoI	Region of Interest
RANSAC	Random Sample Consensus
RAS-Q	Richtlinie für die Anlage von Straßen - Querschnitt
RMS	Root Mean Square (Mittlerer quadratischer Fehler)
RO	Relative Orientierung
SAR	Synthetic Aperture Radar
SBET	Smoothed Best Estimate of Trajectory
SIFT	Scale-Invariant Feature Transform
SURF	Speeded Up Robust Features
SVD	Singular Value Decomposition (Singulärwertzerlegung)
SVM	Support Vector Machine
UTM	Universal Transverse Mercator



# 1 Einleitung

Digitale Bildverarbeitung ist ein wichtiges Werkzeug in vielen Bereichen der Wissenschaft [74]. Ob in der Astronomie, der Elektronenmikroskopie, der Medizin oder wie im vorliegenden Fall der terrestrischen Fernerkundung, die digitale Bildverarbeitung automatisiert bzw. erleichtert die Analyse der gewonnenen Bilddaten. Die Luftbildfotografie geht bis in die Mitte des 19. Jahrhunderts zurück [86], als der Franzose Gaspar Tournachon von einem Heißluftballon über Paris die ersten Luftbilder aufnahm. Heutzutage findet die Luftbildfotografie auf den verschiedensten Gebieten Anwendung, sei es bei der Kartografie, der militärischen Aufklärung, der Stadtplanung, der Luftbildarchäologie oder der Verkehrslage erfassung. Letzterem Themengebiet widmet sich die vorliegende Arbeit.

Eine funktionierende Infrastruktur ist für eine moderne Gesellschaft unerlässlich. Das bedeutet, dass ein der Nachfrage angemessenes Verkehrsnetz existiert, in diesem Falle das Straßennetz. Leider sind jedoch Nachfrage und Angebot nicht immer ausgewogen. Wenn die Nachfrage aus unterschiedlichsten Gründen das Verkehrsangebot übersteigt, kommt es schnell zur Überlastung des Netzes. Um dies zu vermeiden, wird versucht, aktiv in den Verkehr einzutreten. Bevor solch eingreifende Maßnahmen vorgenommen werden können, muss wie bei jedem Regelkreis zunächst gemessen werden. Die wichtigsten Größen sind dabei die Geschwindigkeiten der Fahrzeuge sowie die *Verkehrsstärke*  $Q$  und die *Verkehrsdichte*  $D$  [65].

Diese Größen beziehen sich auf einen Verkehrsstrom. Von einem Verkehrsstrom spricht man, wenn sich mehrere Fahrzeuge auf derselben Spur in dieselbe Richtung bewegen. Die Stromstärke dieses Stromes wird als Verkehrsstärke  $Q$  bezeichnet [88]. Sie leitet sich aus der Anzahl der Fahrzeuge her, die einen Querschnitt am Ort  $x$  in einem Zeitintervall  $\Delta t$  passieren, ist also eine querschnittsbezogene Messung [65]. Wenn die Zahl der am Querschnitt  $x$  zum Zeitpunkt  $t$  gezählten Fahrzeuge als  $N_f(x, t)$  bezeichnet wird, ergibt sich die Verkehrsstärke bezogen auf das Zeitintervall  $[t_1, t_2]$  aus:

$$Q(x, t_1, t_2) = \frac{N_f(x, t_2) - N_f(x, t_1)}{t_2 - t_1} \quad (1.1)$$

Die *Verkehrsdichte*  $D$  hingegen ist eine streckenbezogene Größe und bezieht sich auf die Anzahl der Fahrzeuge auf einem bestimmten Streckenabschnitt  $[x_1, x_2]$  und ist wie folgt definiert:

$$D(x_1, x_2, t) = \frac{N_f(x_2, t) - N_f(x_1, t)}{x_2 - x_1} \quad (1.2)$$

Es gibt die verschiedensten Arten von Detektoren für die Verkehrserfassung. Am weitesten verbreitet sind wohl die Induktionsschleifen [55]. Sie werden in den Boden der

## 1 Einleitung

Straße eingelassen und haben den Vorteil, sehr zuverlässige Detektionsraten zu liefern, solange sie regelmäßig gewartet werden, was mit großem Aufwand verbunden ist ([36] S.31). Ein weiterer Nachteil sind die Kosten und der Aufwand, die Schleife in die Straßoberfläche einzulassen. Andere Systeme detektieren mit Laser, Radar oder Kameras und sind meist auf Brücken, Schilderbrücken oder an Ampeln angebracht.

Nachteil dieser punktuellen Erfassungsmethoden (Kameras ausgenommen) ist, dass sich mit ihnen nur querschnittsbezogene Daten (Verkehrsstärken) erfassen lassen. Angehert lsst sich die Verkehrsdichte uber die Flussrelation aus

$$Q = \bar{v}_m D \approx \bar{v} D \quad (1.3)$$

herleiten, wobei  $\bar{v}_m$  die mittlere momentane Geschwindigkeit aller Fahrzeuge im Streckenabschnitt ist und  $\bar{v}$  die mittlere Geschwindigkeit aller Fahrzeuge, die den Querschnitt  $x$  passieren. Unter der Annahme einer Gleichverteilung der Fahrzeugpositionen und einer Normalverteilung der Geschwindigkeiten sind  $\bar{v}_m$  und  $\bar{v}$  gleich [88], was jedoch in der Praxis nicht immer der Fall ist. Daher ist  $\bar{v}$  allenfalls als Schatzwert fur die mittlere momentane Geschwindigkeit zu betrachten.

Durch luftgesttzte Verkehrserfassung ist es jedoch mglich, die Anzahl, den Ort und die Geschwindigkeit aller (erfassten) Fahrzeuge in einem nahezu beliebig groen rumlichen Intervall zu bestimmen. Dadurch lassen sich Verkehrsstarke und -dichte sowie deren zeitliche Mittelwerte berechnen [65].

Gerade bei Katastrophen oder Groereignissen, bei denen das Straßenverkehrsnetz um ein Vielfaches uberlastet wird, ist an vielen Stellen keine Dichte messbar. In diesen Fallen ist die luftgesttzte Erfassung von groem Vorteil. Zum einen knnen Einsatzzeit und -ort individuell gewhlt werden, zum anderen werden groe Gebiete komplett erfasst und mussen nicht interpoliert werden. Die Werkzeuge und Algorithmen, mit denen ein solches luftgesttztes Erfassungssystem realisiert werden kann, sollen in dieser Arbeit aufgezeigt und neu entwickelt beziehungsweise verbessert werden.

Die Arbeit beschtigt sich mit folgenden Fragestellungen:

- Kalibrierung der aueren Orientierung einer Luftbildkamera und Untersuchung der angegebenen Genauigkeit des Fluglagemesssystems
- Entwicklung von Algorithmen zur Fahrzeigerkennung und -verfolgung unter Bercksichtigung sehr groer Datenmengen
- Entwicklung einer Fluglagebestimmung auf Basis optischer Informationen

In den folgenden drei Kapiteln 2 bis 4 soll dem Leser ein Uberblick uber den Stand der Technik gegeben werden. Es werden Werkzeuge und Algorithmen vorgestellt, die die Grundlage fr die in dieser Art entwickelten Neuerungen bilden. Im zweiten Kapitel wird auf die verwendete Hardware eingegangen. Die Erfassungssysteme *ANTAR* und *ARGOS* werden vorgestellt, welche im Laufe der letzten Jahre fr den Zweck der luftgesttzten Verkehrserfassung am DLR entwickelt wurden. Kapitel 3 befasst sich allgemein mit der Kalibrierung solcher Systeme und zeigt deren Realisierung bei *ANTAR* und *ARGOS*. In

Kapitel 4 werden Algorithmen zur Vorverarbeitung der erfassten Aufnahmen vorgestellt, um diese für die in späteren Kapiteln implementierten Algorithmen aufzubereiten.

Die eigentlichen Innovationen dieser Arbeit werden in Kapitel 5 bis 7 dargelegt. Kapitel 5 befasst sich mit der Entwicklung eines hybriden Erkennungsalgorithmus, welcher die Vorteile aus schnellen aber unzuverlässigen und rechenintensiven aber genauen Algorithmen vereinen soll. Die ungenaue Vorauswahl mit einfachen Modellen erstellt in möglichst kurzer Zeit Fahrzeughypothesen, welche durch eine *Support Vector Machine* verifiziert werden. Die Kombination aus Vorauswahl und Verifizierung führt zu einem Algorithmus, welcher nicht nur zuverlässig detektiert, sondern es schafft, Aufnahmen von 20 bis 30 Megapixel während des Überfluges in Quasi-Echtzeit auszuwerten. Im Vergleich zu anderen Ansätzen werden sowohl Geschwindigkeit als auch Zuverlässigkeit optimiert.

In Kapitel 6 wird ein Trackingalgorithmus vorgestellt, der die detektierten Fahrzeuge in Folgebildern miteinander assoziiert. Da die Aufnahmen nicht kontinuierlich übertragen werden, können Fahrzeuge nicht über längere Bildsequenzen getrackt werden. Dafür wird ein Algorithmus verwendet, der zur Zuordnung von *Interest Points* in Stereobildern entwickelt wurde. Er basiert auf der Singulärwertzerlegung von Matrizen und berücksichtigt die relative Position sowie die Ähnlichkeit der zu assoziierenden Objekte. Dadurch lassen sich die mittleren Geschwindigkeiten der erfassten Fahrzeuge zuverlässig ableiten.

Kapitel 7 enthält eine alternative optische Orientierungsbestimmung, welche entwickelt wurde, um das Fluglagemesssystem IMU zu ersetzen. Da das Fluglagesystem eine der kostspieligsten Komponenten im ganzen System ist, soll die Messung der Flugzeug- bzw. Kameraorientierung durch den optischen Sensor übernommen werden. Mit Hilfe der GPS-Position wird die Fluglage durch Matching von bekannten Straßensegmenten geschätzt. Gestützt wird das Ganze durch eine zusätzliche relative Orientierung, um falsche Messungen zu kompensieren. Mittels eines Kalman-Filters und eines *RANSAC*-Algorithmus wurde in dieser Arbeit die korrekte Fluglage für sämtliche elf Testdatensätze auf durchschnittlich 0,17 Grad genau ermittelt.

Ein Ausblick auf weitere Forschungsthemen ist letztlich in Kapitel 8 zusammengefasst.



## 2 Aufnahmesysteme

Der Ansatz der luftgestützten Verkehrsdatenerfassung wird im Deutschen Zentrum für Luft- und Raumfahrt schon länger verfolgt. Zwei zu diesem Zweck entwickelte Systeme zur Verkehrslageerfassung bilden die Datengrundlage für die vorliegende Arbeit. Zum einen das System *ANTAR*, welches im Rahmen des LUMOS-Projektes bereits 2003 auf der ITSC-Konferenz in Shanghai vorgestellt wurde (vgl. [27] und [28]) und bisher in Flugzeugen, Hubschraubern und Zeppelinen zum Einsatz kam. Zum anderen befasst sich die Arbeit mit dem System *ARGOS*, mit dessen Entwicklung im Jahre 2007 im gleichnamigen Projekt am Institut für Methodik der Fernerkundung begonnen wurde. Die grundsätzliche Idee bestand darin, ein flexibles, auf kommerziellen Baugruppen basierendes Aufnahmesystem zu entwickeln, welches im Falle von Großereignissen oder Katastrophenszenarien in der Lage ist, im Überflug großflächig die Verkehrslage zu erfassen. Die Aufnahmen sollen als Entscheidungshilfe für Einsatzkräfte wie Feuerwehr, Polizei oder das Technische Hilfswerk (THW) dienen. Grundsätzlich arbeitet *ARGOS* nicht anders als das *ANTAR*-System, nur soll aufgrund größerer Flughöhe und höherer Auflösung die Bodenabdeckung um ein Vielfaches erhöht und somit die Aufnahmezeit verringert werden. Im Folgenden sollen beide Systeme kurz vorgestellt werden.

### 2.1 ANTAR

Der „Airborne Traffic Analyzer“ (kurz *ANTAR*) besteht aus einer monochromen CCD-Kamera (siehe Abb. 2.1) mit einer Auflösung von 1024 x 1024 Pixeln. Diese liefert Luftbilder mit einer Frequenz von fünf Hertz. Eine zweite Kamera mit 320 x 240 Pixeln liefert zusätzlich Aufnahmen im thermalen Infrarotbereich. Diese kann aufgrund der geringeren Auflösung eine Bildrate von bis zu 25 Hertz erreichen. Zusätzlich liefert das Inertialsystem POS-AV 410 von Applanix (siehe [76] und Abb. 2.1) die aktuelle Flugzeugposition und -orientierung zu jeder Aufnahme.

Die Luftbilder werden im JPEG-Format gespeichert und erhalten Zusatzinformationen wie Fluglagedaten (GPS und IMU) sowie einen Zeitstempel (Abb. 2.2 gelb). Die Aufnahmen werden per Funkverbindung (Downlink) mit einer Datenrate von bis zu fünf MBit/s an die Bodenstation weitergeleitet und dort in Echtzeit verarbeitet (Abb. 2.2 blau).

### 2.2 ARGOS

Das „Airborne Wide Area High Altitude Monitoring System“ (kurz *ARGOS*) ist ähnlich aufgebaut wie das *ANTAR*-System.

## 2 Aufnahmesysteme



Abbildung 2.1: Kameras und IMU des *ANTAR*-Systems

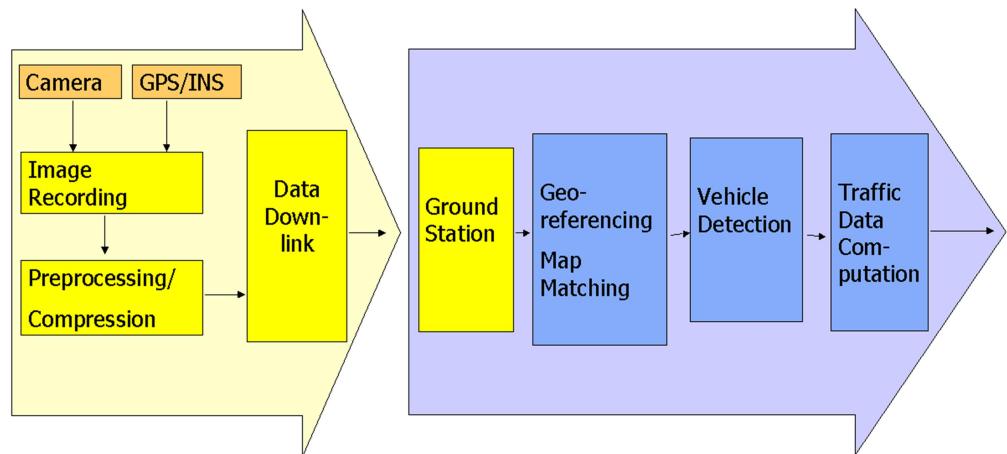
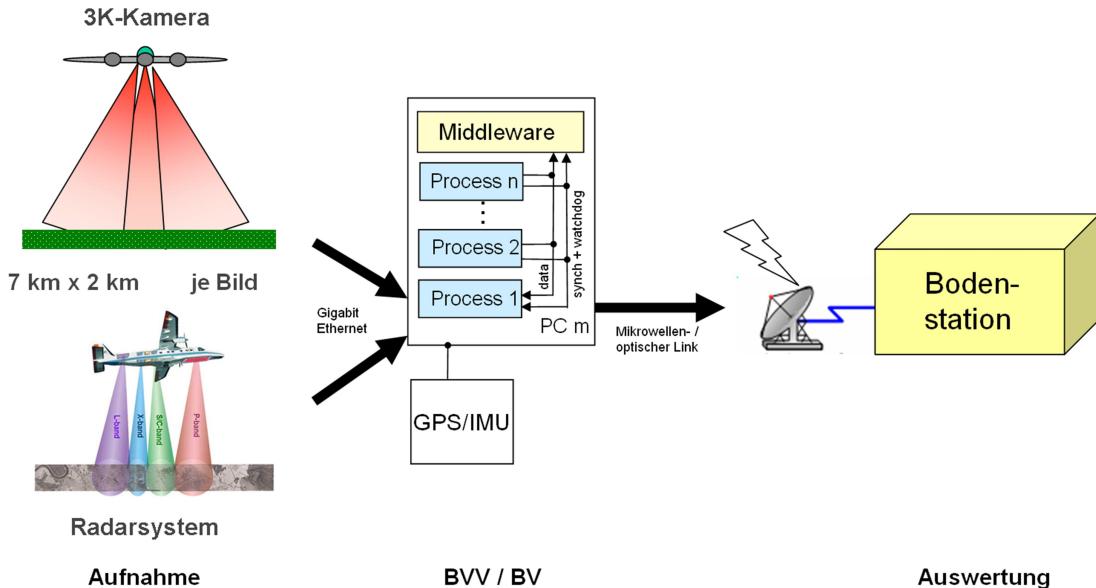


Abbildung 2.2: Schematischer Aufbau des *ANTAR*-Systems (Quelle: [53])

Das Kernstück der optischen Erfassung bildet die 3K-Kamera (3K = 3-Kopf) (siehe Abb. 2.3). Die in dieser Arbeit verwendete Version ist eine Kombination aus drei Spiegelreflexkameras vom Typ „Canon EOS 1Ds Mark II“ mit CMOS-Vollformatsensor und einer Auflösung von je 4992 x 3328 Pixeln (16,7 MPix). Die Bildrate der Kameras liegt bei höchstens drei Aufnahmen pro Sekunde. Erschwerend für den Echtzeitbetrieb kommt hinzu, dass die Kamera nach zwei bis vier Aufnahmen den internen Puffer leert und die Daten auf die Speicherplatte schreibt. Dabei wird die Aufnahme für sieben bis zehn Sekunden unterbrochen. Erst bei einer Bildrate von 2 Hertz wäre der kontinuierliche Betrieb möglich. Daher wird in Erwägung gezogen, den Ringpuffer der Kamera

Abbildung 2.3: 3K-Kamera des *ARGOS*

## Aufbau ARGOS

Abbildung 2.4: Schematischer Aufbau des *ARGOS*

direkt auszulesen, anstatt die Daten auf die Speicherplatte zu übertragen, was jedoch zur Anfertigung dieser Arbeit noch nicht realisiert war. Daneben enthält *ARGOS* ein SAR-System (Siehe [45] und 2.4 links unten), welches zusätzlich Radar-Daten erheben soll, die jedoch hier nicht weiter berücksichtigt werden.

Die vorliegende Arbeit konzentriert sich jedoch ausschließlich auf die Aufnahme und Verarbeitung optischer Daten. Die *ARGOS*-Daten werden, anders als bei *ANTAR*, direkt an Bord verarbeitet (Abb. 2.4 Mitte). Der Downlink soll nur dazu dienen, bereits ermit-

## *2 Aufnahmesysteme*

telte Verkehrskenngrößen und Kartenbilder an die Bodenstation zu übertragen (Abb. 2.4 rechts). Die Informationen können von dort je nach Bedarf an verschiedene Interessenten wie Feuerwehr, THW oder Polizei weitergeleitet werden.

Für die Erfassung der Fluglagedaten ist ein Inertialsystem (GPS/IMU) der „Ingenieur-Gesellschaft für Interfaces mbH“ (IGI) integriert [8]. Dies besteht aus drei orthogonal angeordneten Beschleunigungssensoren, welche die lineare Beschleunigung in  $x$ -,  $y$ -, und  $z$ -Richtung messen, sowie drei ebenfalls orthogonal angeordneten Drehratensensoren, welche die Winkelbeschleunigungen um die drei Achsen bestimmen. In Kombination mit einem GPS-Receiver können alle sechs Parameter der äußeren Orientierung erfasst werden.

# 3 Kalibrierung

Die im folgenden Kapitel beschriebene Kalibrierung bezieht sich auf das Inertialsystem Applanix POS-AV 410 des *ANTAR*-Systems. Um den im Bild erfassten Objekten, in diesem Fall den Fahrzeugen, eine geografische Position zuordnen zu können, sie zu georeferenzieren, müssen die äußere (Position und Richtung) und innere Orientierung (Abbildungseigenschaften) der Kamera zum Aufnahmepunkt bekannt sein. Die je nach Modell bis zu zehn oder mehr Parameter der inneren Orientierung werden im Vorfeld bestimmt und ändern sich während eines Einsatzes normalerweise nicht. Die sechs Parameter der äußeren Orientierung hingegen, drei für die Translation (Position) und drei für die Rotation (Richtung), werden während des Fluges ständig von einem Inertialsystem erfasst und gespeichert. Nachfolgend wird das Inertialsystem auch als IMU (Inertial Measurement Unit) bezeichnet. Diese Art der direkten Georeferenzierung wird heutzutage für fast jede erdenkliche Art von luftgestützten Sensoren verwendet und hat die klassische Aerotriangulation, d.h. das Ausrichten jeder einzelnen Aufnahme durch Referenzpunkte, weitestgehend ersetzt (siehe [21] und [89]).

Da jedoch die äußere Orientierung der Kamera nicht mit der der IMU übereinstimmt, muss diese durch zusätzliche Rotation und Translation ausgeglichen werden. Die Winkel, welche den Rotationsoffset zwischen dem IMU-Koordinatensystem und dem Kameraprojektionszentrum beschreiben, werden als Boresightwinkel bezeichnet. Diese Winkel werden im Vorfeld in Testflügen bestimmt.

Für gewöhnlich sind die Translationen zwischen Kamera, Inertialsystem und GPS-Antenne ebenfalls von Belang, doch können diese in unserem Fall vernachlässigt werden. Der „lever arm“ also der Abstand zwischen Inertial-System und GPS-Antenne (siehe 3.1), wurde vorher gemessen und wird vom Inertialsystem kompensiert. Der Abstand zwischen Inertialsystem und Kamera ist so gering (ca. 10 cm), dass er bei der zu erwarteten Genauigkeit der Kalibrierung nicht ins Gewicht fällt. Somit werden die Nullpunkte des Projektionszentrums und des Inertialsystems als gleich angenommen.

## 3.1 Koordinatensysteme

Zuerst wird festgelegt, auf welche Koordinatensysteme sich die Orientierungsangaben beziehen. Die Koordinatensysteme sind mit Index und Bezeichnung in den Abbildungen 3.1 und 3.2 skizziert. Zusätzlich stellen die Pfeile die Umrechnungen zwischen den Systemen mit den zugehörigen Rotationsmatrizen  $M_y^x$  dar.

Zum einen gibt es das Kartenkoordinatensystem (Index  $m$ , Abb. 3.1) mit der Ausrichtung  $X_m = \text{Osten}$ ,  $Y_m = \text{Norden}$  und  $Z_m = \text{Höhe}$  (Angaben in Meter). Dieses System ist ein UTM-Koordinatensystem (siehe [2]), in welches die GPS-Koordinaten projiziert werden, und ist somit als Weltkoordinatensystem zu verstehen.

### 3 Kalibrierung

Als zweites existiert das Kamerakoordinatensystem (Index  $c$ , Abb. 3.1) mit der Ausrichtung ( $X_c = \text{Rechts}$ ,  $Y_c = \text{Oben}$ ,  $Z_c = \text{Negative Blickrichtung}$ ). Da die Kamera im Flugzeug mit dem Blick nach unten gerichtet ist, zeigt die  $Y$ -Achse im Weltkoordinatensystem in Flugrichtung und die  $Z$ -Achse nach oben.

Das dritte Referenzsystem ist das Bildkoordinatensystem (Index  $i$ ), welches die zweidimensionale Sensorebene der Kamera darstellt und somit auch nur zwei Koordinaten hat. Dieses System ist in Abbildung 3.2 skizziert. Es befindet sich im Abstand der Brennweite vor dem Projektionszentrum, lotrecht zur optischen Achse und hat die Ausrichtung  $X_i = \text{Rechts}$ ,  $Y_i = \text{Oben}$  (Angaben in Millimeter), ist also achsparallel zum Kamerakoordinatensystem.

Als vierter wird das Pixelkoordinatensystem verwendet (Index  $p$ ), welches ebenfalls in Abbildung 3.2 dargestellt ist. Es hat die Ausrichtung  $X_p = \text{Rechts}$ ,  $Y_p = \text{Unten}$  (Angaben in Pixel). Es liegt koplanar zum Bildkoordinatensystem, hat seinen Ursprung jedoch nicht auf der optischen Achse sondern in der linken oberen Ecke des Bildes. Die Entfernung zwischen Pixelursprung und optischer Achse wird durch die Größe und Anzahl der Sensorelemente definiert.

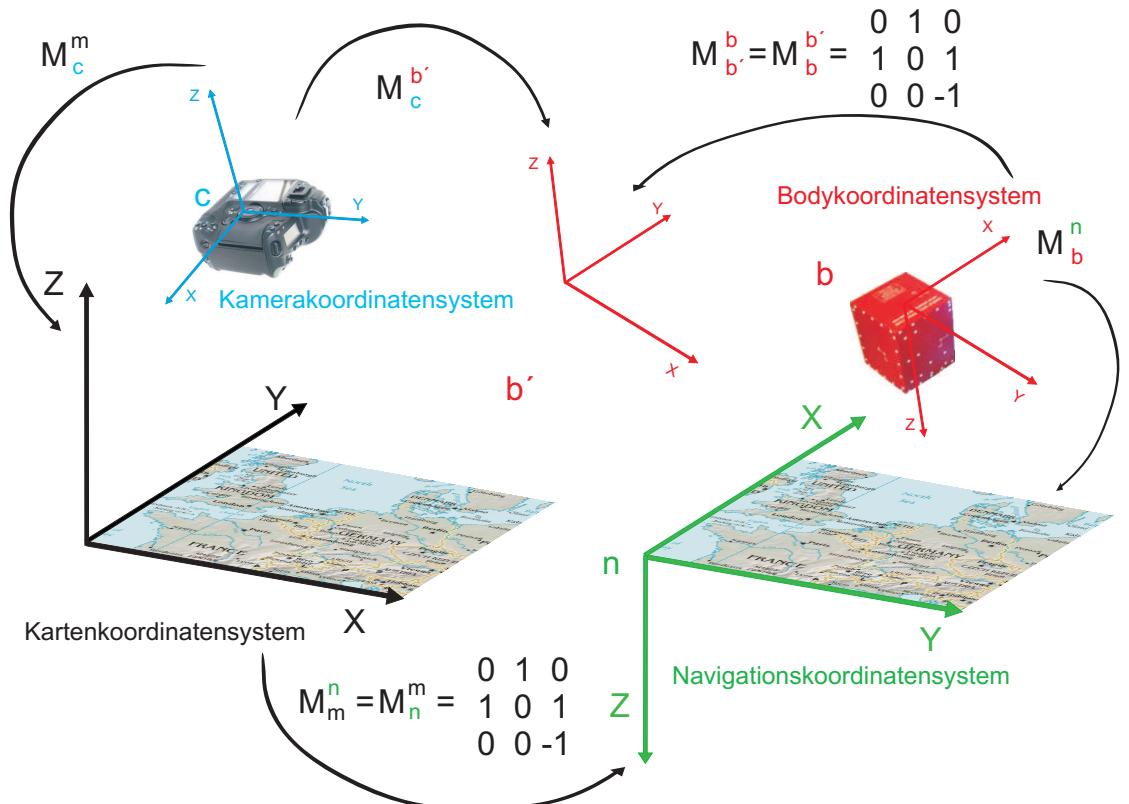


Abbildung 3.1: Verwendete Koordinatensysteme

Eine zweites Weltkoordinatensystem, das Navigationskoordinatensystem (Index  $n$ , Abb. 3.1) wird vom Inertialsystem benutzt. Es ist zwar achsparallel zum Kartenko-

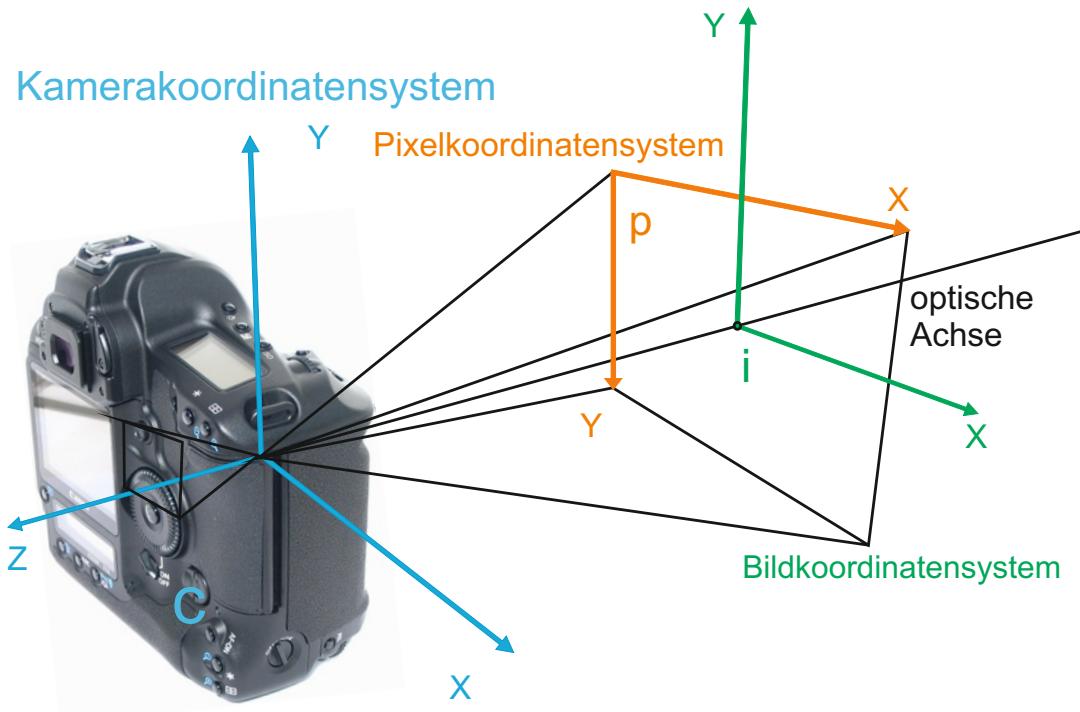


Abbildung 3.2: Verwendete Koordinatensysteme

ordinatensystem, hat jedoch die Ausrichtung  $X_n = \text{Nord}$ ,  $Y_n = \text{Ost}$ ,  $Z_n = \text{Unten}$ .

In diesem Koordinatensystem wiederum ist die Orientierung des Bodykoordinatensystems (Index  $b$ , Abb. 3.1) definiert, welches das Inertialsystem selbst darstellt. Es hat die Ausrichtung  $X_b = \text{Flugrichtung}$ ,  $Y_b = \text{Rechts}$ ,  $Z_b = \text{Unten}$ .

## 3.2 Innere Orientierung

Um die äußere Orientierung der Kamera auf Grundlage von Bildinformationen bestimmen zu können, müssen ihre Abbildungseigenschaften, muss also die innere Orientierung bekannt sein. Die Abbildung umfasst die Transformation der Koordinaten vom Kamerakoordinatensystem ( $X_c, Y_c, Z_c$ ) über das Bildkoordinatensystem mit normalisierten  $(x_i, y_i, -c)$  und verzeichneten Bildkoordinaten  $(\bar{x}_i, \bar{y}_i, -c)$  bis hin zum Pixelkoordinatensystem  $(x_p, y_p)$ .

$$(X_c, Y_c, Z_c) \rightarrow (x_i, y_i, -c) \rightarrow (\bar{x}_i, \bar{y}_i, -c) \rightarrow (x_p, y_p) \quad (3.1)$$

Nach dem geometrischen Abbildungsmodell (vgl. [70]) werden Objekte zentralperspek-

### 3 Kalibrierung

tivisch auf die Bildebene projiziert.

$$x_i = \frac{X}{Z} \cdot c \quad (3.2)$$

$$y_i = \frac{Y}{Z} \cdot c \quad (3.3)$$

Aufgrund des vergleichsweise kleinen Öffnungswinkels und der geringen Linsenverzeichnung wurde das Lochkamera-Modell in Verbindung mit einem vereinfachten Verzeichnungsmodell gewählt. Im Falle sehr kurzer Brennweiten und großer Öffnungswinkel (ab  $180^\circ$ ) lassen sich einfallende Strahlen nicht mehr durch Zentralperspektive und Verzeichnungskorrektur abbilden, und andere Abbildungsmodelle müssen angewandt werden (siehe [52] und [69]).

Für die Korrektur der Verzeichnung wurde der Ansatz von Brown (vgl. [17]) gewählt. Dabei werden die abgebildeten Punkte je nach Position und Abstand  $r$  zur optischen Achse um einen bestimmten Vektor  $(\Delta x, \Delta y)$  verschoben. Dieser errechnet sich in drei Schritten:

$$\Delta(x, y) = \Delta(x_r, y_r) + \Delta(x_d, y_d) + \Delta(x_a, y_a) \quad (3.4)$$

Radial-symmetrische Verzeichnung:

$$\Delta x_r = \bar{x}(r^2 K_1 + r^4 K_2 + r^6 K_3) \quad (3.5)$$

$$\Delta y_r = \bar{y}(r^2 K_1 + r^4 K_2 + r^6 K_3) \quad (3.6)$$

Radial-asymmetrische Verzeichnung:

$$\Delta x_d = (r^2 + 2\bar{x}^2)P_1 + 2\bar{x}\bar{y}P_2 \quad (3.7)$$

$$\Delta y_d = 2\bar{x}\bar{y}P_1 + (r^2 + 2\bar{y}^2)P_2 \quad (3.8)$$

Affinität und Scherung:

$$\Delta x_a = \bar{x}B_1 + \bar{y}B_2 \quad (3.9)$$

$$\Delta y_a = 0 \quad (3.10)$$

Die Verzeichnungskoeffizienten  $K_1, K_2, K_3, P_1, P_2, B_1$  und  $B_2$  sind Parameter der inneren Orientierung der Kamera und müssen bei der Kalibrierung bestimmt werden. Allerdings werden mit den Gleichungen 3.4 bis 3.10 die normalisierten Koordinaten  $x$  und  $y$  aus den verzeichneten Koordinaten  $\bar{x}$  und  $\bar{y}$  errechnet. Da die sich ergebende Gesamtgleichung nicht invertierbar ist, müssen die Verzeichnungen iterativ errechnet werden, wobei die idealen Koordinaten  $x$  und  $y$  als Startwert genommen werden können. Experimente haben ergeben, dass beim verwendeten Objektiv nur zwei Iterationen notwendig

### 3.3 Bestimmung der Einbauwinkel

sind, um die Verzeichnung zu korrigieren.

Hinzu kommt, dass die optische Achse nicht exakt durch die Mitte des Bildsensors verläuft und somit bei der Quantisierung eines Bildpunktes neben der Größe und Anzahl der Sensorelemente die Verschiebung des Bildhauptpunktes ( $X_p$  und  $Y_p$ ) berücksichtigt werden muss. Die Pixelkoordinaten im Bild errechnen sich aus den verzeichneten Bildkoordinaten somit wie folgt:

$$x_p = \frac{\text{cols}}{2} + 0.5 + \frac{\bar{x} + X_p}{d_x} \quad (3.11)$$

$$y_p = \frac{\text{rows}}{2} + 0.5 + \frac{\bar{y} + Y_p}{d_y} \quad (3.12)$$

Die Parameter *cols* und *rows* geben die Anzahl der Sensorelemente pro Zeile und Spalte an,  $d_x$  und  $d_y$  stehen für die Größe der Elemente und  $X_p$  und  $Y_p$  für die Bildhauptpunktverschiebung. Die genannten Parameter wurden mithilfe der Kalibriersoftware „Australis“ und bekannter Objektpunkte durch eine Bündelblockausgleichung bestimmt (siehe Abb. 3.3), welche in [70] beschrieben ist. Von den Verzeichnungsparametern  $K_1$  bis  $B_2$  werden jedoch nur die ersten beiden  $K_1$  und  $K_2$  in den weiteren Rechnungen berücksichtigt.

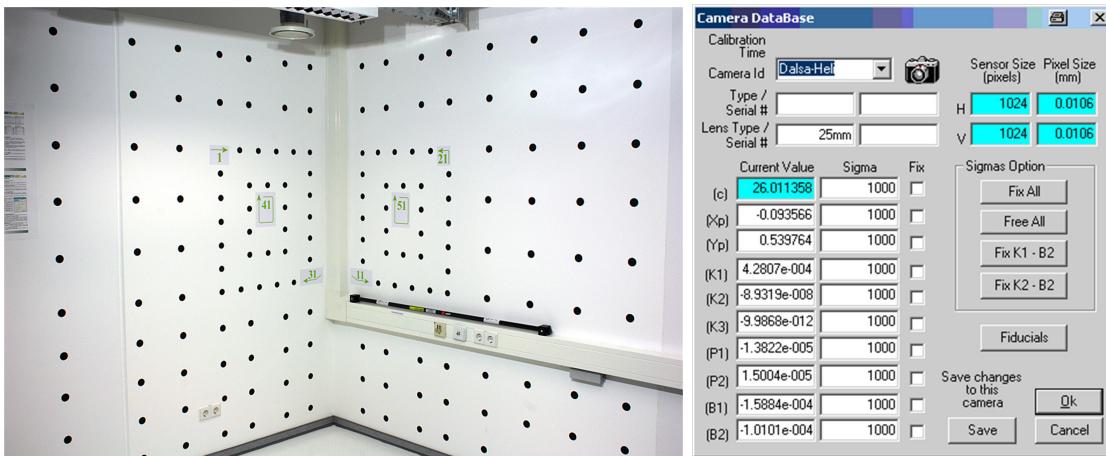


Abbildung 3.3: Kalibrierwand und innere Orientierung

### 3.3 Bestimmung der Einbauwinkel

Während die innere Orientierung die Transformation vom Kamera- (Index  $c$ ) ins Pixelkoordinatensystem (Index  $p$ ) beschreibt, bezieht sich die Äußere Orientierung auf die Transformation vom Navigations- (Index  $n$ ) in das Kamerakoordinatensystem (Index  $c$ ). Was an dieser Stelle jedoch ermittelt werden soll, sind die Einbauwinkel der Kamera im Verhältnis zum Inertialsystem, also der Zusammenhang der äußeren Orientierung beider Systeme (Indizes  $c$  und  $b$ ). Mithilfe der inneren Orientierung der Kamera und bekannter

### 3 Kalibrierung

Objektpunkte können Position und Blickrichtung der Kamera zum Aufnahmezeitpunkt berechnet werden. Mit dem eigens dafür implementierten ClickTool (Abb. 3.4) werden diese Objektpunkte im Bild angewählt. Da die Positionen der Punkte im Vorfeld durch DGPS bestimmt wurden, kann für jede Aufnahme durch Räumlichen Rückwärtsschnitt (siehe [70]) eine äußere Orientierung der Kamera errechnet werden. Diese äußere Orientierung entspricht der Transformation eines Punktes vom Karten- ins Kamerakoordinatensystem, also der Drehmatrix  $M_m^c$  (siehe Abb. 3.1) und dem Translationsvektor  $T_m^c$ . Da die Translation, wie schon erwähnt, vernachlässigt wird, interessieren uns fortan nur die Rotationsmatrizen.



Abbildung 3.4: ClickTool mit Aufnahme aus dem ANTAR-System

Die Aufnahmen werden mit einem weiteren Programm, welches die eigentliche Bore-sight-Bestimmung vornimmt, in eine Liste geladen und die zugehörigen Orientierungsdaten des Inertialsystems aus der Aufnahme extrahiert oder aus einer externen Datei geladen. Die IMU-Orientierungen wurden zum Aufnahmezeitpunkt als Zusatzinformation in die jeweilige JPEG-Datei geschrieben. Werden nachprozessierte IMU/GPS-Daten verwendet, stehen diese in zusätzlichen *SBET*-Dateien. *SBET* steht für „Smoothed Best Estimation of Trajectory“. Das Inertialsystem ist mit zusätzlicher Post-Processing-

### 3.3 Bestimmung der Einbauwinkel

Software in der Lage, die fehlerbehafteten Navigationsdaten zu filtern (siehe [77]) und eine optimale Trajektorie zu schätzen.

Die Position des Bodysystems  $T_b^n$ , welche gleichzeitig die Position der Kamera darstellt, ist in WGS84-Koordinaten angegeben, also zwei Winkel für die geographische Breite und Länge, und muss per UTM-Projektion in das vom GPS verwendete UTM-Koordinatensystem projiziert werden. Dabei muss die Meridiankonvergenz beachtet werden. Da sich die geografische Nordrichtung der IMU von der Nordrichtung des UTM-Koordinatensystems unterscheidet (siehe [49] und [62]), wird diese dementsprechend korrigiert.

Die Orientierung der IMU entspricht der Rotation eines Punktes vom Body- ins Navigationskoordinatensystem, also der Matrix  $M_b^n$  (siehe Abb. 3.1), wobei die Rotationsreihenfolge anders als üblich hier roll ( $\phi$ ), pitch ( $\theta$ ), heading ( $\psi$ ) (also  $X-$ ,  $Y-$ ,  $Z-$ Achse oder Roll-, Nick-, und Gierwinkel) ist (siehe [18] und [48]). Die gesamte Rotationsmatrix definiert sich somit wie folgt:

$$M_b^n = \mathbf{R}_z(\psi) \cdot \mathbf{R}_y(\theta) \cdot \mathbf{R}_x(\phi) \quad (3.13)$$

$$= \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \quad (3.14)$$

Photogrammetrisch (siehe [70]) wird jedoch vorwiegen die Drehreihenfolge  $\kappa, \phi, \omega$  verwendet (also  $Z-, Y-, X-$ Achse). Die Rotationsmatrix  $M_c^m$  eines Punktes vom Kamera- ins Kartenskoordinatensystem definiert sich wie folgt:

$$M_c^m = \mathbf{R}_x(\omega) \cdot \mathbf{R}_y(\phi) \cdot \mathbf{R}_z(\kappa) \quad (3.15)$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\omega) & -\sin(\omega) \\ 0 & \sin(\omega) & \cos(\omega) \end{bmatrix} \cdot \begin{bmatrix} \cos(\phi) & 0 & \sin(\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{bmatrix} \cdot \begin{bmatrix} \cos(\kappa) & -\sin(\kappa) & 0 \\ \sin(\kappa) & \cos(\kappa) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.16)$$

Um die obere Drehung  $M_b^n$  auf einen Punkt aus dem Kamera- ins Kartenskoordinatensystem anzuwenden, müssen die  $x$ - und  $y$ -Achse vor und nach der Drehung vertauscht werden. Die  $z$ -Achse wird gespiegelt.

$$M_n^m = M_m^n = M_b^{\bar{b}} = M_{\bar{b}}^b = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (3.17)$$

Die Rotationsmatrix  $M_n^m$  beschreibt also sowohl eine Drehung vom Karten- ins Navigationskoordinatensystem als auch die gegensätzliche. Die neue Rotationsmatrix  $M_b^m$

### 3 Kalibrierung

ergibt sich dann aus

$$\mathbf{M}_b^m = (\mathbf{M}_n^m)^{-1} \cdot \mathbf{M}_b^n \cdot \mathbf{M}_n^m \quad (3.18)$$

Somit wurde ein zweites Body-Koordinatensystem definiert (Index  $\bar{b}$ ). Dessen Ursprung befindet sich nach wie vor an gleicher Stelle, seine Achsen sind jedoch wie die der Systeme  $m$  und  $c$  ausgerichtet. Um die Rotation des  $\bar{b}$ -Koordinatensystems in das Kamerakoordinatensystem zu ermitteln, welches der eigentlichen Boresight-Rotation entspricht, muss ein Punkt nun die folgenden Rotationen durchlaufen:

$$\mathbf{M}_{\bar{b}}^c = (\mathbf{M}_c^m)^{-1} \cdot \mathbf{M}_{\bar{b}}^m \quad (3.19)$$

Erstens findet eine Rotation vom  $\bar{b}$ -Koordinatensystem ins Kartenkoordinatensystem ( $\mathbf{M}_{\bar{b}}^m$ ) und anschließend von diesem ins Kamerakoordinatensystem statt (siehe [92]). Die erhaltene Boresight-Matrix rotiert einen Punkt vom  $\bar{b}$ - ins Kamerakoordinatensystem. Die eigentlichen Boresightwinkel lassen sich aus der Rotationsmatrix mithilfe der speziellen Arkustangens-Funktion  $\text{atan2}(y, x)$  ermitteln. Da die Eindeutigkeit des Winkels bei der normalen Arkustangens-Funktion nicht gewährleistet ist, d.h. der Quadrant nicht bestimmt wird, gibt es in vielen Programmiersprachen eine Funktion, die mit zwei Argumenten aufgerufen wird. Anstelle ihres Quotienten  $\frac{y}{x}$  werden die kartesischen Koordinaten einzeln übergeben (siehe [5]). Die Boresightwinkel sind demnach aus den Elementen der Matrix wie folgt zu berechnen:

$$\phi = \text{atan2}\left(\mathbf{M}_{1,3}, \sqrt{\mathbf{M}_{2,3}^2 + \mathbf{M}_{3,3}^2}\right) \quad (3.20)$$

$$\omega = \text{atan2}\left(\frac{-\mathbf{M}_{2,3}}{\cos(\phi)}, \frac{\mathbf{M}_{3,3}}{\cos(\phi)}\right) \quad (3.21)$$

$$\kappa = \text{atan2}\left(\frac{-\mathbf{M}_{1,2}}{\cos(\phi)}, \frac{\mathbf{M}_{1,1}}{\cos(\phi)}\right) \quad (3.22)$$

Für mehrere Einzelaufnahmen aus vier Anflugrichtungen wurden so die Boresightwinkel durch eine eigens dafür implementierte Software ermittelt (Abb. 3.5). Zusammen mit der aktuellen IMU-Orientierung wurden Kontrollpunkte in alle Aufnahmen projiziert, deren GPS-Positionen ebenfalls bekannt waren. Die Projektionen wurden mit vorher angeklickten Passpunkten verglichen und die mittleren quadratischen Fehler berechnet. Die Kameraposition wurde vom GPS ermittelt und in der Ausgleichsrechnung als fixe Eingangsgröße behandelt. Für die Bestimmung der Boresightwinkel wurden nur nachprozessierte GPS/IMU-Daten (*SBET*) verwendet, welche nach Angaben des Herstellers „Applanix Corporation“ die folgenden Genauigkeiten aufweisen.

Im Allgemeinen ist bei Inertialsystemen die Genauigkeit des Gierwinkels (heading) um das Zwei- bis Dreifache schlechter als die der Roll- (roll) und Nickwinkel (pitch). Das kommt daher, dass der Gierwinkel nicht durch die Erdbeschleunigung korrigiert werden kann, da er sich in einer Ebene parallel zur Erdoberfläche befindet (siehe [97]).

Die ermittelten Winkel unterliegen jedoch größeren Streuungen als erwartet (Abb. 3.5

### 3.3 Bestimmung der Einbauwinkel

Parameter Accuracy (RMS)	POS/AV <sup>TM</sup> 410
Position (m)	0.05 - 0.30
Velocity (m/s)	0.005
Roll, Pitch (Deg)	0.008
True Heading (Deg)	0.015

Tabelle 3.1: Genauigkeit des Applanix-Inertial-systems (Quelle: [76])

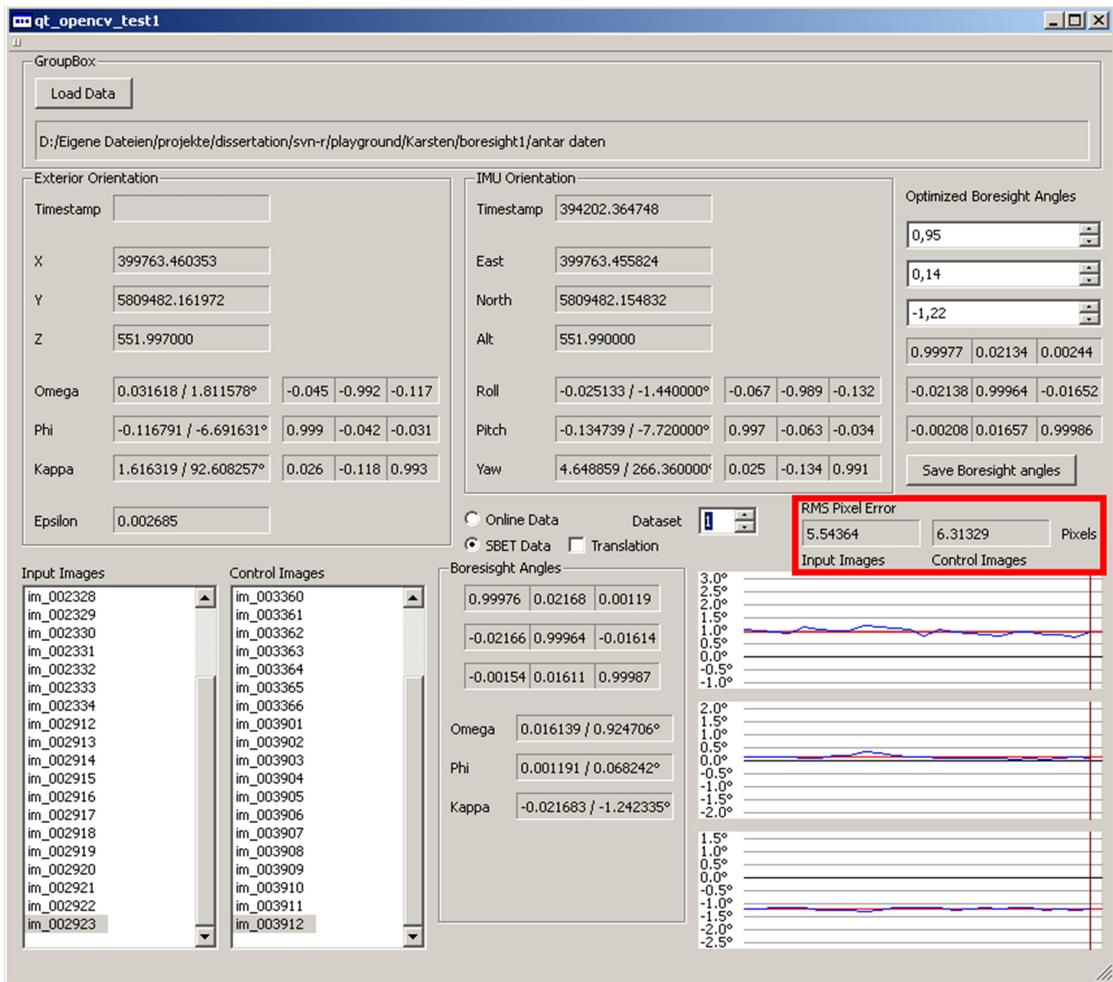


Abbildung 3.5: Software zur Ermittlung der Boresightwinkel (RMS-Fehler rot markiert)

und 3.6). Wie die Abbildung 3.7 zeigt, schwanken die Winkel  $\omega$ ,  $\phi$  und  $\kappa$  mit Standardabweichungen von 0,05 bis 0,12 Grad. Diese Ungenauigkeiten ergeben bei den rückprojizierten Kontrollpunkten einen mittleren quadratischen Fehler (RMS) von 5-6 Pixeln, was bei der gegebenen Auflösung und Flughöhe einem Versatz von 1,0 bis 1,2 Metern

### 3 Kalibrierung



Abbildung 3.6: Disparitäten zweier aufeinanderfolgender Aufnahmen (weiße Striche)

am Boden entspricht.

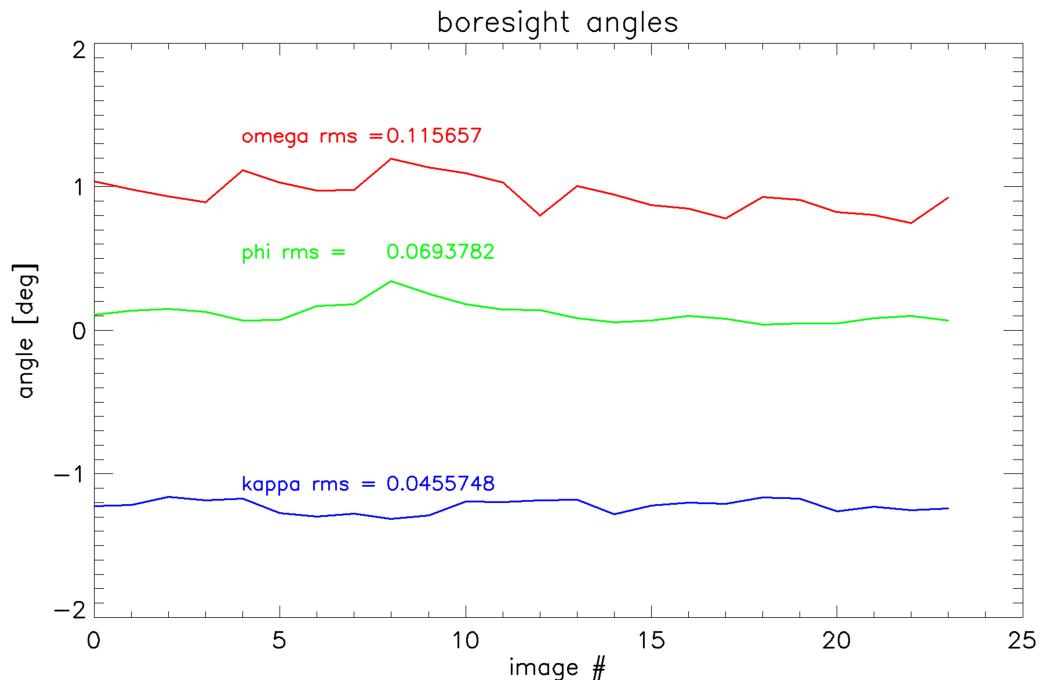


Abbildung 3.7: Ermittelte Boresightwinkel für verschiedene Aufnahmen

Nach den Angaben in Tabelle 3.1 kann der RMS der GPS-Einheit bei nachprozessierten Daten bis zu 30 cm betragen. Bei den gegebenen Parametern entspricht das ca. 1,5 Pixeln im Bild. Die Fehler der IMU von 0,005 und 0,008 Grad beim Roll- und Nickwinkel fallen mit ca. einem drittel Pixel in der Bildmitte wie am Rand nicht ins Gewicht. Der Kurswinkelfehler von 0,015 Grad würde am Bildrand weniger als ein achtel Pixel betragen, ist also auch vernachlässigbar klein. Diese Ungenauigkeit wird auch in [37] bestätigt und mit der unzureichenden Dämpfung der Kamera sowie ungenauer Synchronisierung zwischen IMU und Kamera erklärt.

Aufgrund der Vermutung, dass IMU und/oder GPS-Daten verzögert erzeugt werden, wurden systematisch verschiedene zeitliche Offsets verwendet und die Ergebnisse analysiert (siehe Abbildung 3.8). Auch wenn die minimalen RMS-Werte bei einem Zeitversatz

### 3.3 Bestimmung der Einbauwinkel

von  $-1\text{ms}$  liegen, unterscheiden sich diese nur geringfügig von den ursprünglichen Fehlerwerten bei 0 ms. Es lässt sich also für keinen Offset eine signifikante Verbesserung erkennen.

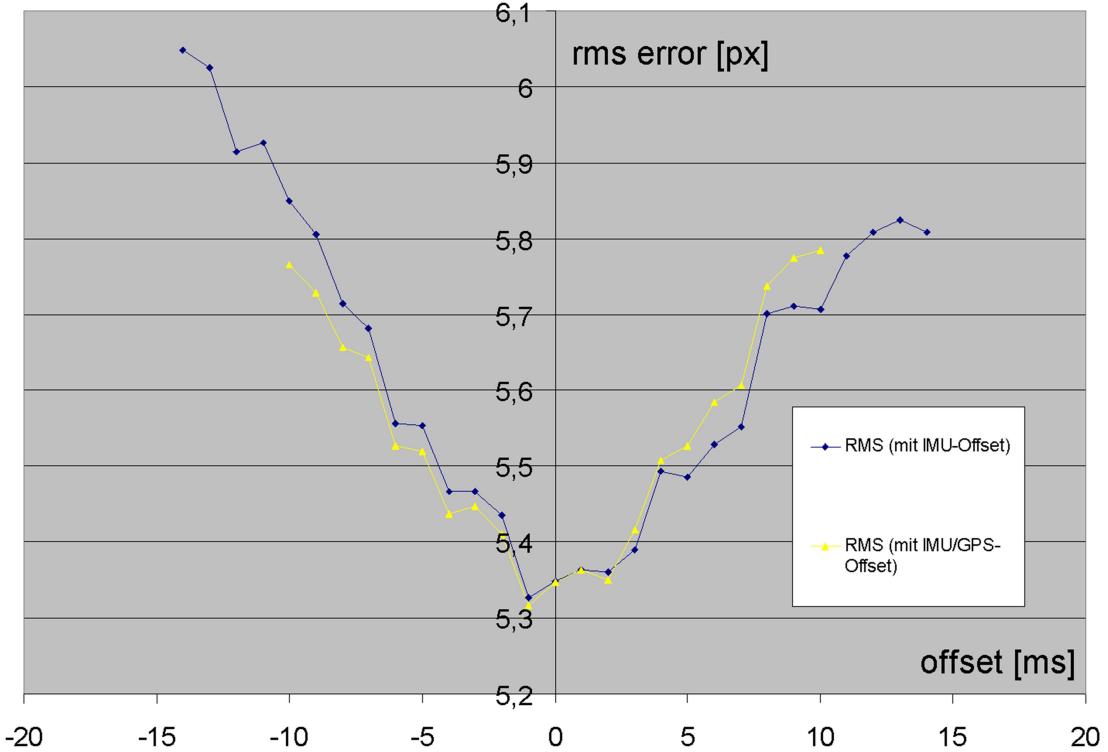


Abbildung 3.8: RMS-Werte bei verschiedenen zeitlichen Offsets

Eine weitere Vermutung war, dass die Kamera nicht exakt in dem Moment auslöst, wenn die IMU von ihr getriggert wird. Wenn sich bei festen Boresightwinkeln für jede Aufnahme ein individueller Zeitoffset finden ließe, bei dem der Fehler signifikant kleiner wird, wäre der Fehler auf die unstetige Synchronisation zwischen IMU und Kamera zurückzuführen. Doch auch der Versuch, die Ungenauigkeit mit einem variablen Synchronisations-Offset auszugleichen, war nicht möglich. Zwar ließ sich in einigen Aufnahmen der Fehler auf 0,5 bis 1,5 Pixel reduzieren, doch lag er bei drei der neun Kalibrerbildern nach wie vor zwischen vier und sieben Pixeln.

Die Experimente zeigen, dass sich die vom Hersteller angegebene Genauigkeit des Inertialsystems nicht ohne Weiteres erreichen lässt. Die nächste in Frage kommende Ursache für die Ungenauigkeiten wäre, wie auch in [37] erwähnt wird, die schlechte Dämpfung des Systems. Weitere Experimente mit *ANTAR* könnten diese Hypothese bestätigen, wurden jedoch für diese Arbeit nicht weiter verfolgt.

Die bisher erreichbare Genauigkeit des *ANTAR*-Systems liegt somit bei einer Flughöhe von 500 Metern zwischen 1,0 und 1,2 Metern am Boden.

### 3.4 Alternative Kalibration bei ARGOS

Im Institut für Methodik der Fernerkundung am DLR-Standort Oberpfaffenhofen wurde eine alternative Bestimmung der Einbauwinkel für die 3K-Kamera des *ARGOS* entwickelt (siehe [63]). Die Boresightwinkel werden während des Fluges ohne bekannte Passpunkte oder Ground Control Points (GCP) ermittelt. Dabei werden je drei Aufnahmen berücksichtigt, in denen beliebige Kontrollpunkte festgelegt und verfolgt werden. Mit den aktuellen Fluglagedaten und den Boresightwinkel müssten die Strahlen durch die Kontrollpunkte aller drei Aufnahmen in einem Raumpunkt konvergieren (siehe Abb. 3.9). Die drei Boresightwinkel werden in einer Ausgleichsrechnung so optimiert, dass die Raumstrahlen in minimalem Abstand aneinander vorbeilaufen.

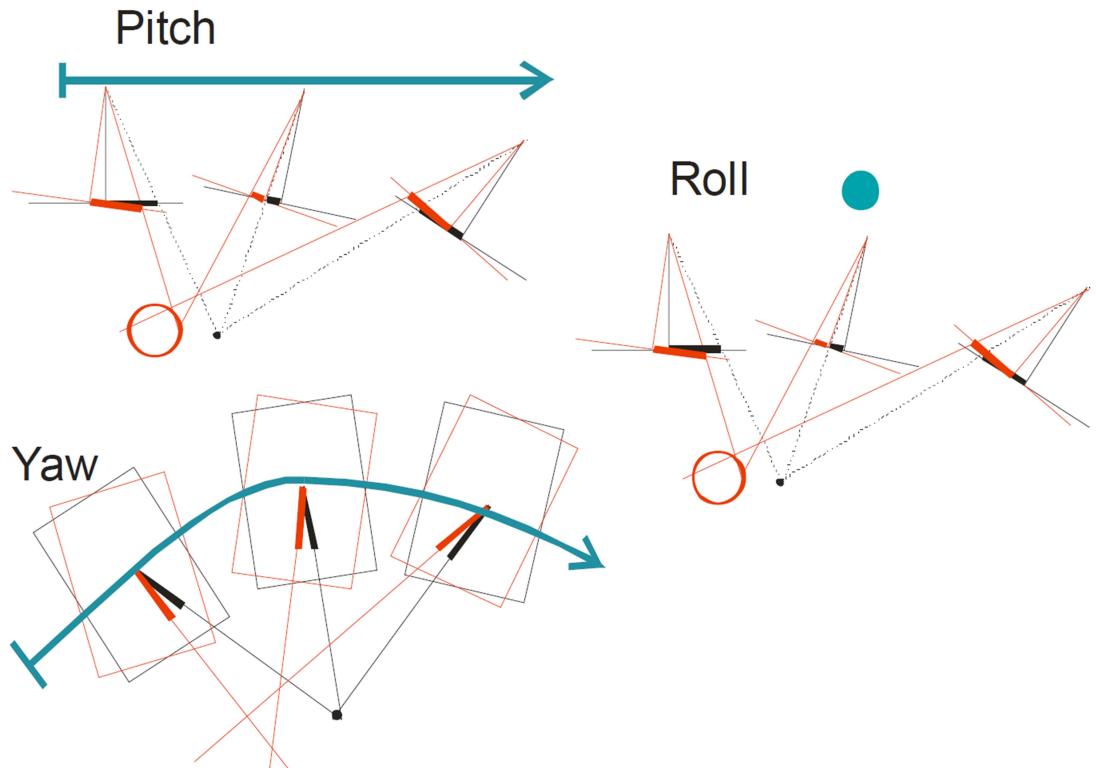


Abbildung 3.9: Kalibrierung mit drei Aufnahmen (rot - ohne Boresightwinkel, schwarz - mit korrekten Boresightwinkel, Quelle: [63])

Die Genauigkeit dieser Kalibrierung ist mit fünf Metern in der Position und einem Meter in der Höhe zwar signifikant geringer als die klassische Verwendung von GCP (1,0 und 0,5 m), doch reicht sie für die Erfassung von Verkehrsdaten aus.

# 4 Vorverarbeitung

## 4.1 Georeferenzierung

Um jedem Pixel im Bild eine geografische Position zuzuordnen, werden die Luftaufnahmen orthorektifiziert und georeferenziert. Bei der Orthorektifizierung werden alle Bildpunkte unter Berücksichtigung sämtlicher Verzerrungen durch Linse, Perspektive oder Gelände senkrecht auf eine Referenzebene projiziert. Bei der Georeferenzierung werden die genordeten Aufnahmen als TIFF-Dateien gespeichert und mit einem geografischen Ankerpunkt und einem Skalierungsfaktor versehen. Die entstandene Format wird als GeoTIFF bezeichnet. Auf diese Weise können die Aufnahmen direkt auf Karten im UTM-Format (vgl. [2]) abgebildet werden.

Bei der im *ANTAR*-System verwendeten Software *TrafficFinder* werden die Bilder nach der sogenannten Ankerpunktmethode rektifiziert ([61] S. 426 ff.). Dafür werden für die vier Bildecken Höhenwerte berechnet und für dazwischenliegende Pixel interpoliert. Auch wird die geringe Linsenverzerrung der Kamera nicht berücksichtigt.

Anders als bei später verwendeten *ARGOS*-Aufnahmen wurden die GeoTIFFs aus den *ANTAR*-Aufnahmen (siehe Abb. 4.3) für diese Arbeit erneut orthorektifiziert. Dies geschah nach der klassischen Pixel-by-Pixel-Methode. Anstatt alle Pixel des Originalbildes mit einem Höhenmodell zu schneiden, wird zuerst das leere Orthobild oder GeoTIFF erstellt, für jedes Pixel die Höhe ermittelt und das Pixel mit dem korrespondierenden Grauwert aus dem Originalbild gefüllt. Die Berechnung geschieht auf der CPU und ist mit ca. 20 s pro Aufnahme nicht echtzeitfähig, soll aber auch nicht in der Prozessierungskette verwendet werden. Als Höhenmodell stehen die Daten der *Shuttle Radar Topography Mission* [7] aus dem Jahre 2000 zur Verfügung. Diese enthalten für nahezu die gesamte Erdoberfläche Höhendaten mit einer Auflösung von 3 Bogensekunden (ca. 90 m). Für die vorliegende Arbeit wurde ein Ausschnitt von  $47^\circ$  bis  $57^\circ$  nördlicher Breite und  $7^\circ$  bis  $17^\circ$  östlicher Länge verwendet. Der Orthorektifizierungs- und Georeferenzierungsprozess dazu verläuft wie folgt:

- Bestimmung des Skalierungsfaktors / der Bodenauflösung

Wichtig für die Balance aus Rechenzeit und Informationsverlust ist der Skalierungsfaktor (Länge und Breite eines Bodenpixels). Zu kleine Pixel führen zu längeren Rechenzeiten, während bei der Projektion auf zu große Pixel Bildinformationen verloren gehen. Einerseits kann der optimale Skalierungsfaktor  $s$  für jedes Bild dynamisch aus der Flughöhe  $h$ , der Pixelgröße des Sensors  $dx$  und der Brennweite

## 4 Vorverarbeitung

$c$  bestimmt werden.

$$s = \frac{h \cdot dx}{c} \quad (4.1)$$

Unkomplizierter ist es jedoch für die Koregistrierung, also die geografische Registrierung der Bilder relativ zueinander, die Bodenauflösung vorher festzulegen. So kann für alle Aufnahmen die gleiche Skalierung verwendet werden. Im gezeigten Beispiel beträgt diese 20 cm/Pixel.

- Festlegung der Bildgrenzen

Um die Bodengrenzen des Orthobildes bestimmen zu können, müssen die maximalen und minimalen Ost- und Nordwerte des Bildausschnittes ermittelt werden. Dafür werden die vier Ecken des Originalbildes auf das Höhenmodell projiziert. Da nicht für jeden Punkt Höhendaten zur Verfügung stehen und somit die Richtung des projizierten Strahls nicht von Anfang an bestimmt werden kann, werden die Schnitte iterativ ermittelt (Siehe Abb. 4.1).

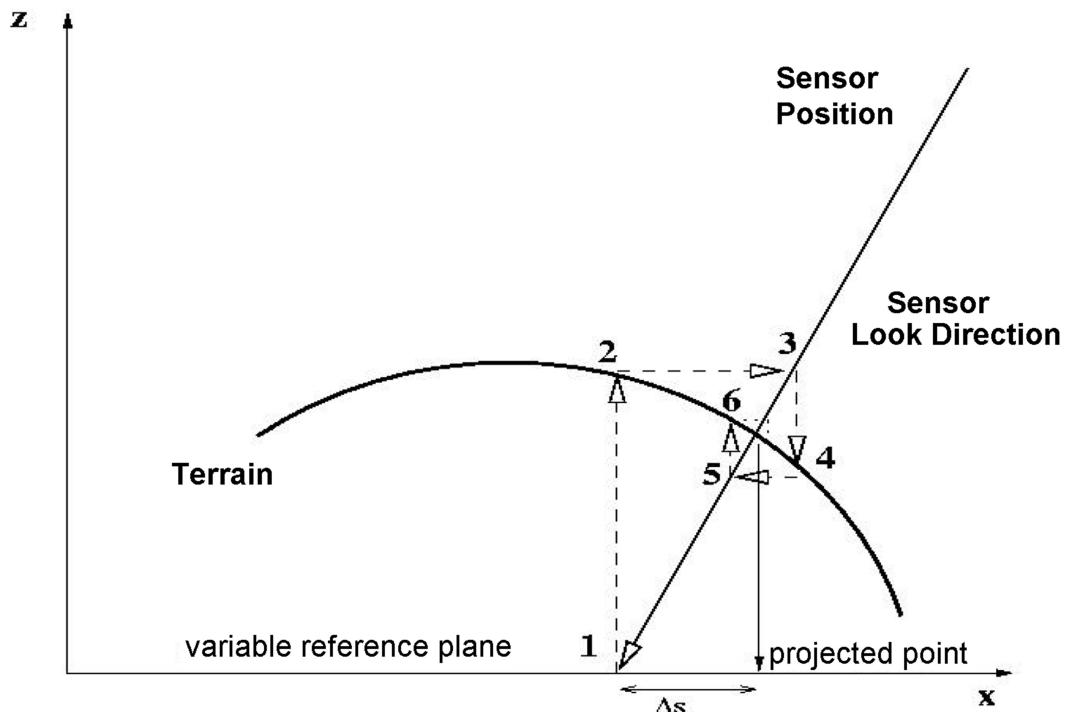


Abbildung 4.1: Iterative Höhenbestimmung (Quelle: [73])

Als Ausgangshöhe wird ein Wert für ein von der Flugzeugposition gefälltes Lot bilinear interpoliert (Abb. 4.1). Der Projektionsstrahl einer Bildecke wird mit dieser Höhenebene geschnitten und für die bestimmte Position eine neue Höhe errechnet.

Der Strahl wird wiederum mit dieser Höhe geschnitten um eine genauere Schnittposition zu ermitteln. Dieser Vorgang wird solange wiederholt, bis sich der Schnittpunkt in horizontaler Richtung nur noch minimal verschiebt. Aus den Koordinatenwerten der vier Schnittpunkte ergeben sich die Bodengrenzen des Orthobildes. Aus diesen kann dann mit Hilfe der vorher bestimmten Bodenauflösung die Größe des Orthobildes errechnet werden.

- Füllen des Orthobildes

Für jedes Pixel im Orthobild wird ein Höhenwert interpoliert und von dieser 3D-Position aus ein Strahl durch das Projektionszentrum mit der Bildebene geschnitten (Abb. 4.2). Der Schnittpunkt wird nach dem Brown-Modell verzerrt und in ein Bildpixel umgerechnet (vgl. Kapitel 3.2). Aus der daraus resultierenden Position kann der Grauwert aus dem Originalbild gelesen werden. Um Aliasing-Effekte zu vermeiden, sollte auch der gelesene Farbwert interpoliert werden.

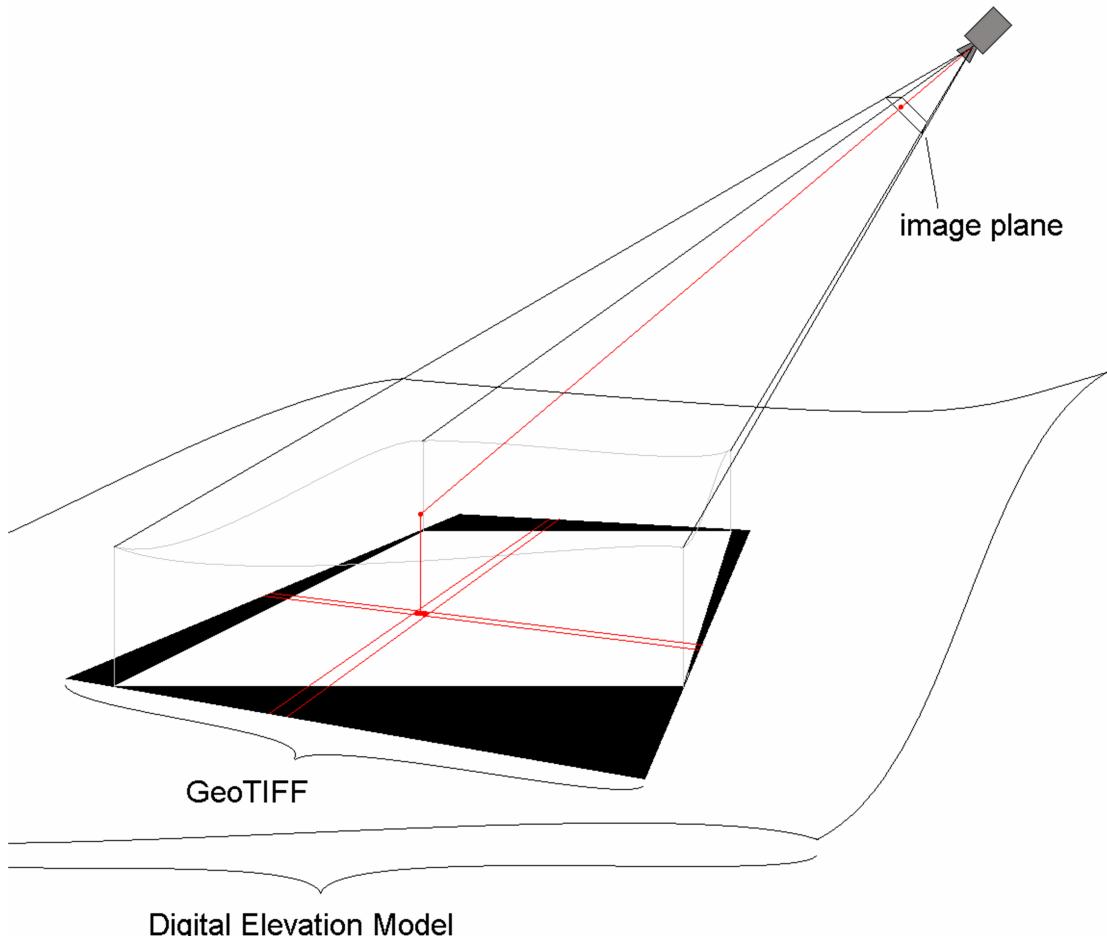


Abbildung 4.2: Projektion bei der GeoTIFF-Erzeugung

## 4 Vorverarbeitung



Abbildung 4.3: GeoTIFF der DLR-Messstrecke „Ernst-Ruska-Ufer“ in Berlin-Adlershof  
(Aufgenommen mit *ANTAR*)

### 4.1.1 GeoTIFF

Um das orthorektifizierte Bild zu georeferenzieren, wird es im TIFF-Format gespeichert und mit zusätzlichen Informationen versehen. „libtiff“ ist eine freie C-Bibliothek und kann zur Arbeit mit TIFF-Dateien verwendet werden. Die Bilder haben vier Kanäle mit je acht Bit Farbtiefe, drei Farbkanäle und einen Alpha-Kanal, der den relevanten Bereich des Bildes maskiert. Die Bibliothek „libgeotiff“ setzt darauf auf und kann das spezielle Format GeoTIFF verwalten. GeoTIFFs sind TIFF-Dateien mit zusätzlichen Metatags, die geographische Informationen zum Bild enthalten. Die wichtigsten sind die

geographische Lage (UTM) mindestens einer Bildecke sowie der Skalierungsfaktor.

Bei *ARGOS* hingegen werden (siehe [96]) alle Pixel des Originalbildes mit dem Höhenmodell geschnitten und von dort ins Orthobild projiziert. Die verbleibenden Lücken auf der Referenzebene müssen dann interpoliert werden. Die erforderlichen Rechenschritte werden per OpenGL auf der Grafikkarte ausgeführt, wobei bei 16,7 Megapixeln bis zu zehn Aufnahmen pro Sekunde verarbeitet werden können. Die Orthobildgenerierung bei *ARGOS* ist somit echtzeitfähig.

#### 4.1.2 Genauigkeit der Orthobilder

Um die Genauigkeit der GeoTIFF-Bilder zu prüfen, wurden aus drei verschiedenen Anflügen je drei Kontrollbilder ausgewählt und in diesen manuell Passpunkte festgelegt. Da die GPS-Positionen dieser Passpunkte bekannt sind, können sie ebenfalls automatisch in das GeoTIFF projiziert werden (siehe Abb. 4.4). Ähnlich wie bei der Kalibrierung der Einbauwinkel (Kap. 3.3) kann die mittlere quadratische Abweichung zwischen der erwarteten und der projizierten Position errechnet werden. Wie schon durch die Genauigkeit der äußeren Orientierung zu erwarten war, lag der RMS-Fehler in den Bildern insgesamt bei 5,6 Pixeln, also ca. einem Meter.



Abbildung 4.4: Passpunkte in GeoTIFF-Bildern, grün - wahre Position, rot - projizierte Position, links - RMS = 1,6 Pixel (30 cm), rechts - RMS = 7,5 Pixel (150 cm)

## 4.2 Straßenprojektion

Um bei der späteren Fahrzeugerkennung (Kapitel 5) Rechenzeit zu sparen sowie Fehlerkennungen zu reduzieren, wird der Suchraum für Verkehrsobjekte zunächst eingegrenzt. Dafür werden die Straßen in den Aufnahmen als *Regions of Interest* maskiert. Digitale Straßenkarten können bei *Navigation Technologies (NAVTEQ* [6]) oder dem *Amtlichen Topographisch-Kartographischen Informationssystem (ATKIS* [11]) erworben werden. Als Datengrundlage für die vorliegende Arbeit wurde die Straßendatenbank von *NAVTEQ* aus dem Jahr 2009 verwendet und direkt in die GeoTIFFs projiziert. Die Daten enthalten die UTM-Koordinaten von Knotenpunkten, die ihnen zugeordneten Straßenabschnitte als Kanten sowie zahlreiche zusätzliche Informationen wie z.B. die Anzahl der

## 4 Vorverarbeitung

Fahrspuren oder die Straßenkategorie. Die Straßenstücke werden in einer Liste mit den zugehörigen Koordinaten ihrer Endpunkte sowie ihres Mittelpunktes abgelegt. Koordinaten werden so zwar redundant gespeichert, doch erlaubt dies schnelleren Zugriff. Da das Lesen der *NAVTEQ*-Dateien im Textformat sehr lange dauert, werden die Listen in eigenen Dateien gespeichert.

Um nicht alle Straßen ins Bild projizieren zu müssen, wird für jede Kante geprüft, ob sich ihr Mittelpunkt in einer bestimmten Umgebung des georeferenzierten Bildes befindet. Da die Länge der Kanten sehr unterschiedlich ist und teilweise mehr als einen Kilometer betragen kann, werden alle Kanten vorher auf eine Maximallänge von 400 Metern zerteilt. Wenn sich der Mittelpunkt einer Kante im Umkreis von 200 Metern um das GeoTIFF-Bild befindet, kann diese im Bild zu sehen sein und wird gezeichnet.

### 4.2.1 Straßenbreite

Prinzipiell gibt es zwei Möglichkeiten, die Straßenbreite zu bestimmen. Zum einen kann diese aus einer Kombination von Spurbreite und Spuranzahl sowie der Breite der zugehörigen Seiten- und Mittelstreifen errechnet werden, andererseits gibt es nach der *Richtlinie für die Anlage von Straßen - Querschnitt* (kurz *RAS-Q*, siehe [1]) eindeutige Vorschriften, was den Gesamtquerschnitt bezüglich der Belastung angeht. Beide Methoden haben jedoch Schwachpunkte.

- Bei die Bestimmung der Gesamtbreite durch Regelquerschnitte (*RAS-Q*) tritt das Problem auf, dass die Straßen bei *NAVTEQ* in fünf verschiedene Hierarchieklassen eingeteilt sind. Diese lassen sich nur bedingt auf die neun Straßenquerschnitte der *RAS-Q* abbilden.
- Die Errechnung durch Addition aller vorhandenen Streifenbreiten erfordert nicht nur die Berücksichtigung verschiedener Streifenbreiten für die Fahrstreifen (zwischen 2,75 und 3,75 Metern), sondern auch für Mittel-, Seiten-, Seitentrenn- und Randstreifen. Da diese Informationen im Einzelnen nicht durch *NAVTEQ* gegeben werden, ist diese Art der Bestimmung fehlerbehaftet und unnötig aufwendig zu rechnen.

Einfacher und nicht weniger genau ist die Verwendung der Regelquerschnitte durch *RAS-Q*. Die Straßenbreiten in Deutschland sind in neun Regelquerschnitte (RQ 7,5 bis RQ 35,5) eingeteilt, wobei der Name dem Querschnitt in Metern entspricht. Fünf dieser Querschnitte (1,3,5,7 und 9) werden den fünf Hierarchieklassen von *NAVTEQ* zugeordnet. Zusätzlich wird ein Kalibrierfehler der äußeren Orientierung durchschnittlich 1,5 m auf beiden Straßenseiten dazugerechnet, um sicherzustellen, dass möglichst wenige Fahrzeuge durch ungenaue Orientierung ausgeschlossen werden.

### 4.2.2 Genauigkeit der Straßenprojektion

Die *NAVTEQ*-Daten sind jedoch fehlerhaft, einerseits weil Straßen ständig neu gebaut werden, zum anderen sind ihre GPS-Positionen ungenau. Beispielsweise ist die DLR-Messstrecke am „Ernst-Ruska-Ufer“ als zweispurige Straße der Hierarchie-Klasse 4 mit

je einem Fahrstreifen pro Richtung angegeben (siehe Abb. 4.5), was dem RQ 7,5 (7,5 m Breite) entspricht. Doch ist diese Straße mittlerweile vierspurig ausgebaut worden und hat mit dem Randstreifen eine Breite von 16,5 Metern, muss also in die Klasse 2 (entspricht RQ 20) eingeordnet werden.

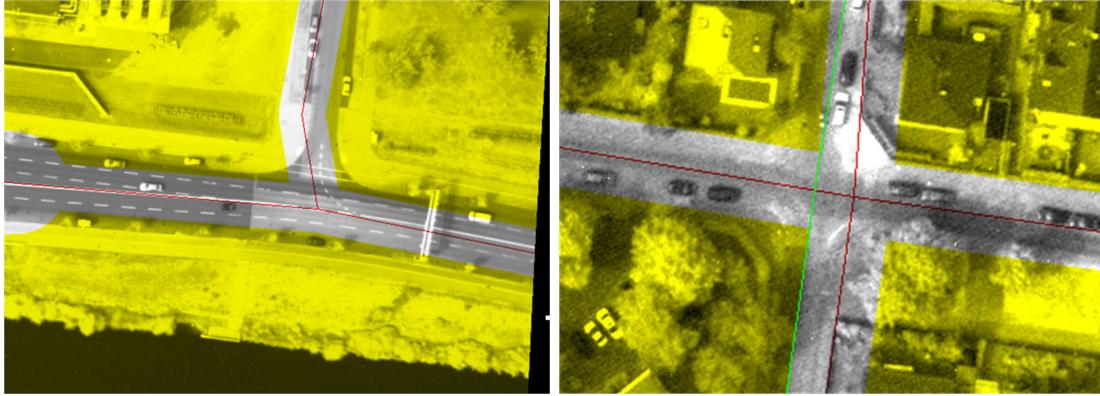


Abbildung 4.5: Fehler in der Straßendatenbank (links falsche Spuranzahl, rechts falsche Position)

In [94] wurden zehn markante *NAVTEQ*-Koordinaten auf ihre Genauigkeit hin überprüft. Die Abweichungen zu den tatsächlichen Koordinaten lagen in den dort verwendeten Testdatensätzen bei durchschnittlich 1,7 Metern. Vereinzelt lagen diese höher, ein Punkt wies sogar einen Positionsfehler von 6,1 Metern auf. Wie in Abbildung 4.5 rechts zu sehen ist, werden die Koordinaten von Straßenknoten bei kleineren Straßen manchmal sehr ungenau angegeben. Im gezeigten Beispiel liegt die vertikal verlaufende Straße (grün) mehr als fünf Meter neben der gegebenen *NAVTEQ*-Kante (rot). Fahrzeuge am linken Rand der Straße würden somit trotz des Puffers von je 1,5 Metern nicht im Erkennungsbereich liegen.

### 4.2.3 Straßenkorrektur

Grundsätzlich gibt es zwei Möglichkeiten, die ungenauen Straßenkoordinaten der *NAVTEQ*-Datenbank zu korrigieren. Zum einen kann man eine genauere Datengrundlage benutzen, da *NAVTEQ* nicht als einziger Anbieter in Frage kommt. Die *Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland (AdV)* verfügt zwar durch das *ATKIS*-Projekt (Amtliches Topographisch-Kartographisches Informationssystem) über weitaus detailliertere Straßenkarten, doch stellte sich deren Genauigkeit im verwendeten Testdatensatz [94] mit 3,3 Metern als noch unzuverlässiger heraus. Die zweite Alternative ist somit, die Straßenlage aus den vorhandenen Bilddaten zu korrigieren. Dafür wurde eine einfache eindimensionale Optimierung implementiert, welche das jeweilige Straßenelement orthogonal zur Fahrtrichtung verschiebt. Als Bewertung werden die Pixel gezählt, welche gleichzeitig als Straßenoberfläche angenommen und vom projizierten Straßenelement abgedeckt werden. Dafür muss die Straßenoberfläche extrahiert werden. Ein einfacher Ansatz geht von einem bestimmten Farbwert der Straße

#### 4 Vorverarbeitung

aus. Auch wenn diese als grau erscheint, deckt deren Farbe im HSV-Farbraum [9] einen bestimmten Winkel ab. Gleichzeitig wird die Straße auf ihre Sättigung geprüft, da diese meist sehr gering ist.



Abbildung 4.6: Straßenprojektion unkorrigiert (Straße - rot, Umgebung - türkis)

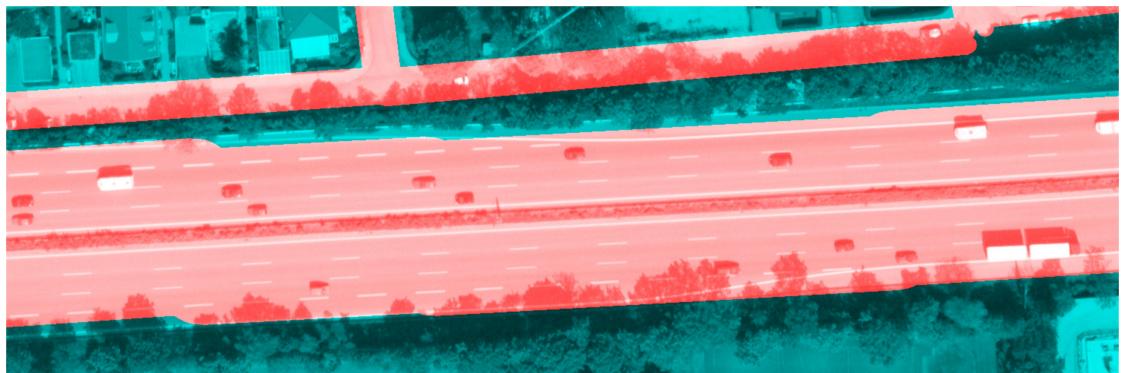


Abbildung 4.7: Straßenprojektion korrigiert

Wie die Abbildungen 4.6 und 4.7 zeigen, werden die Straßenelemente in ihrer Lage optimiert. Dadurch sind mehr Fahrzeuge von der Straßenmaske abgedeckt, welche ansonsten nicht erkennbar wären. Zusätzlich wird die Falscherkennung verringert, da neben den Straßen befindliche Strukturen nicht berücksichtigt werden. Zu beachten ist jedoch, dass der Farbwinkel der Straßenoberfläche im HSV-Farbraum nicht immer gleich ist, sondern sich leicht mit dem Wetter und dem Weißabgleich der Kamera verändert. An der Stelle sollten bessere Straßenerkennungsalgorithmen, wie z.B. in [44] vorgestellt (siehe Kap. 7.1.1), Abhilfe schaffen. Wie sich die Verbesserung der Straßenposition auf die Erkennung auswirkt, wird in Kapitel 5.3 gezeigt.

## 5 Fahrzeugerkennung

Zur Erkennung von Fahrzeugen in Luftbildern gibt es eine Reihe von Veröffentlichungen und verschiedenste Ansätze. Im Allgemeinen können die Ansätze nach [43] in zwei Gruppen unterteilt werden. Zum einen solche mit impliziter, zum anderen mit expliziter Modellierung.

Erstere Ansätze gehen, wie der Name sagt, von einem impliziten Fahrzeugmodell aus. Das heißt, es existiert anfangs keine Modellbeschreibung. In der Trainingsphase leitet ein Klassifikator eine Vielzahl von Merkmalen (features) aus ihm vorliegenden Bildausschnitten ab. Diese Trainingsdaten werden vom Benutzer als *Fahrzeug* oder *Nicht-Fahrzeug* deklariert. Nach dem Training hat der Algorithmus ein implizites Fahrzeugmodell erlernt. Die Arbeiten [35], [64], [66] und [83] befassen sich mit dem Thema der impliziten Fahrzeugerkennung auf Basis von *Support Vector Machines (SVM)*, *Boosting* oder *High Order Statistics (HOS)*.

In [98] werden Histogramme von orientierten Gradienten (*HOG*) verwendet (vgl. [24]), die ebenfalls durch einen Boosting-Algorithmus (siehe [34]) separiert werden. Dafür werden zunächst mehrere positive und negative Trainingsbeispiele erhoben. Dabei wird die Region um ein Objekt in mehrere Fenster unterteilt, für die die Gradienten bestimmt werden. Aus Betrag und Richtung der Gradienten werden Histogramme erstellt und in einen multidimensionalen Merkmalsraum eingetragen. Danach wird eine Reihe von schwachen Klassifikatoren angewandt, um die Daten zu separieren. Der endgültige Klassifikator ergibt sich aus einer Linearkombination aller schwachen Klassifikatoren. Für die Detektion wird das Bild dann zuerst segmentiert, um verkehrsaktive Flächen zu extrahieren und Rechenzeit zu sparen. Über jedes Pixel dieser Region of Interest (*RoI*) werden die jeweiligen *HOG* berechnet und klassifiziert.

Der Algorithmus erreichte für die untersuchten Szenen Vollständigkeiten von bis zu 87 Prozent, hat jedoch den entscheidenden Nachteil, dass die Berechnungszeit bei vielen Straßen trotz Vorsegmentierung bis zu einer Stunde dauern kann. An dieser Stelle wirkt sich die hohe Anzahl der extrahierten Merkmale (269) negativ aus.

Der explizite Ansatz bezieht sich von Anfang an auf ein oder meist mehrere vordefinierte Fahrzeugmodelle. Diese können als zwei- oder dreidimensionale geometrische Konstrukte vorliegen. Die Erkennung erfolgt meist auf der Basis von Kantenfiltern, wobei versucht wird, die extrahierten Fahrzeugkanten mit den Kanten im Modell zu assoziieren. Zwei einfache Ansätze mit zweidimensionalen Modellen werden in [26] und [75] beschrieben. Während in [26] in kantengefilterten Bildern nach rechteckigen Formen mit bestimmter Größe und Ausrichtung gesucht wird, generiert man bei [75] das Kantenfilter selbst in Fahrzeugform.

Ein etwas aufwändigerer Ansatz wird in [39] verfolgt, indem die Autoren auf Basis von 400 dreidimensionalen Modellen versuchen, den Schattenwurf der Fahrzeuge zu

## 5 Fahrzeugerkennung

bestimmen und zu erkennen. In [43] und [103] gehen die Autoren noch weiter und versuchen zusätzlich die Windschutzscheiben der Fahrzeuge zu erkennen sowie auf Basis eines radiometrischen Modells die Stärke der extrahierten Kanten vorauszusagen. Die finale Entscheidung wird bei [103] durch ein Bayes'sches Netz getroffen.

### 5.1 Ansatz

Auf der einen Seite ist der Algorithmus umso zuverlässiger, je mehr Merkmale extrahiert und verglichen werden, doch kostet das Ganze auch wesentlich mehr Rechenzeit. So hat der in [43] vorgestellte Ansatz zwar eine sehr hohe Zuverlässigkeit, was bedeutet, dass in den Tests keine einzige Fehlextraktion auftrat, doch braucht der Algorithmus nach Angaben des Autors in etwa vier Minuten, um eine Aufnahme von einem Megapixel auszuwerten.

Da die Aufnahmen des *ARGOS* eine Originalauflösung von je 16,7 Megapixeln haben und die generierten GeoTiffs bis zu 30 Megapixel, wäre ein solcher Rechenaufwand nicht angemessen. Gleichzeitig hat das System *ANTAR* gezeigt, dass eine Vollständigkeit von etwa 60 bis 75 Prozent für die Erfassung der Verkehrslage akzeptable Ergebnisse liefert [26].

Im gewählten Ansatz liegt der Fokus somit auch vorerst auf der Geschwindigkeit als auf der Genauigkeit (siehe [59]). Die Entscheidung fiel auf einen zweistufigen Algorithmus, um die Vorteile der schnellen und die der zuverlässigen Detektion zu vereinen.

In einer ersten Phase soll ein möglichst einfaches explizites zweidimensionales Fahrzeugmodell verwendet werden, um in akzeptabler Geschwindigkeit Fahrzeughypothesen für die verkehrsaktiven Flächen der Aufnahme zu sammeln. Der Suchraum wird dabei nochmals auf wenige hundert Pixel reduziert. In Phase zwei sollen diese Hypothesen durch einen rechenintensiveren Klassifizierungsalgorithmus auf ihre Richtigkeit geprüft werden. Dadurch sollen in möglichst kurzer Zeit falsch erkannte Objekte überprüft und verworfen werden, was die Zuverlässigkeit bedeutend verbessert.

#### 5.1.1 Phase eins - Erstellung von Objekthypothesen

Wie schon in [75] vorgestellt, arbeitet der in Phase eins verwendete Algorithmus ebenfalls mit speziell geformten Kantenfiltern. Diese Art der Erkennung liefert die besten Ergebnisse, wenn zusätzlich Informationen über die verkehrsaktive Fläche (Straße) sowie deren Ausrichtung berücksichtigt werden. Dies sind genau die Voraussetzungen, welche durch die Straßendatenbank gegeben und in Kapitel 4.2 beschrieben sind.

Am Anfang wird das Originalbild in ein Grauwertbild umgewandelt, da nach einer Studie von PPG Industries [3] aus dem Jahre 2004 über zwei Drittel der weltweit hergestellten Fahrzeuge keine Farbinformation haben, also schwarz, weiß oder grau sind (38% silbern, 15,4% weiß, 13,9% schwarz). Auch neuere Untersuchungen kommen zu ähnlichen Ergebnissen [4]. Darüberhinaus haben Tests ergeben, dass der Algorithmus bei Anwendung auf alle Farbkanäle keine signifikanten Änderungen der Ergebnisse zur Folge hat. Somit sind zwei Drittel der Rechenzeit redundant.



Abbildung 5.1: GeoTIFF mit Straßensegment

Der Algorithmus projiziert die gegebenen Straßensegmente einzeln in das Orthofoto, bevor ein Rechteck um jedes Straßensegment als *Region of Interest* (Abb. 5.1 und 5.2) ausgeschnitten wird. Je nach Pixelskalierung und Ausrichtung des Straßensegments werden vier Kantenfilter (Abb. 5.3) erstellt, welche die vier Kanten des Fahrzeugs repräsentieren. Die Vorgehensweise entspricht somit einem expliziten zweidimensionalen Fahrzeugmodell, ähnlich wie in [75].

Die Filter liefern vier Antwortbilder (Abb. 5.4), welche zunächst schwellwertgefiltert

## 5 Fahrzeugerkennung

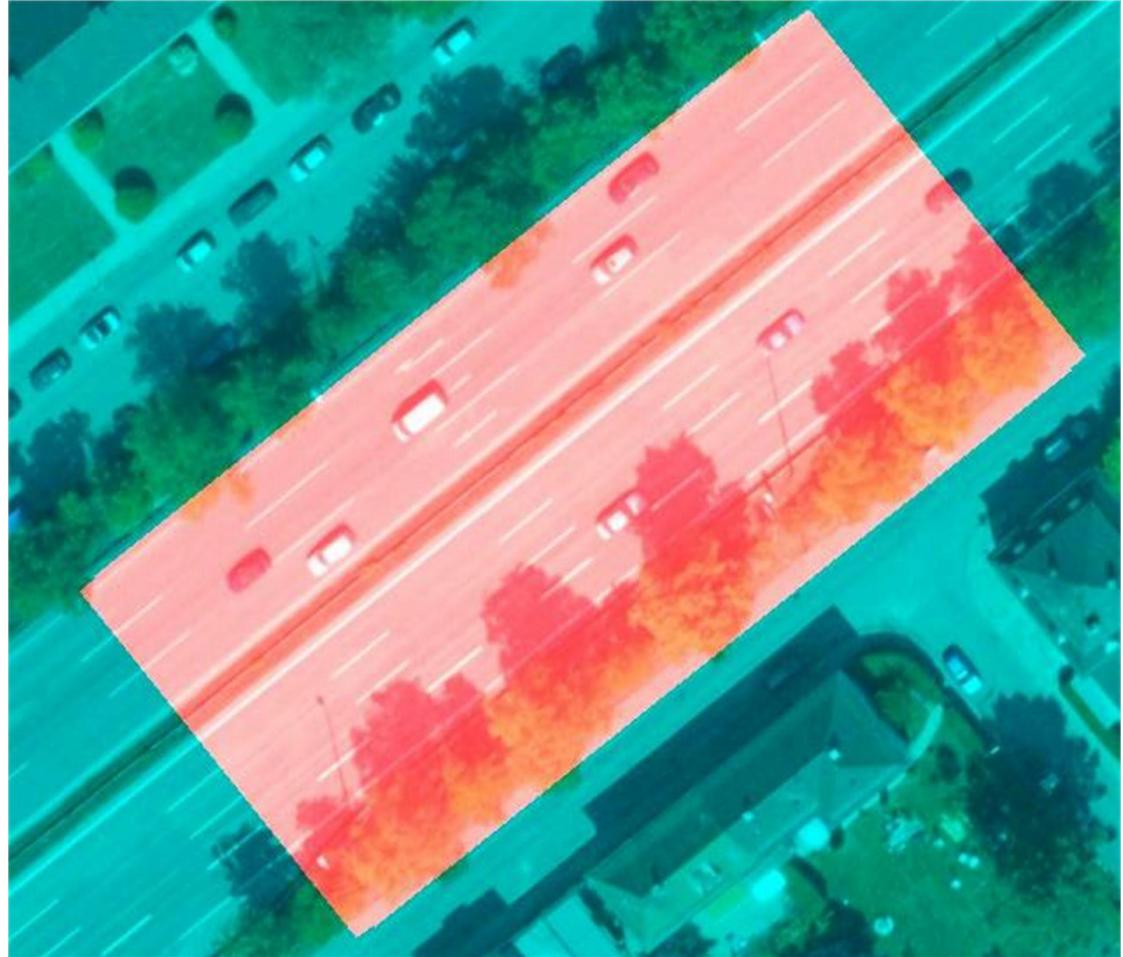


Abbildung 5.2: Straßensegment



Abbildung 5.3: Filter für die vier Fahrzeugkanten

werden (Abb. 5.5). Der Schwellwert kann frei gewählt werden und bestimmt dabei die Sensibilität des Filters. Je höher er ist, desto stärker muss der Kontrast einer Kante sein, um erkannt zu werden. Anstatt die Kanten für verschiedene Fahrzeuggrößen zurecht zu schieben und die Aufnahmen mehrfach zu filtern, ist es eleganter, die Filterantworten selbst (je nach Fahrzeuglänge und -breite) zu verschieben. Die Kanten müssen in beide Richtungen verschoben werden, um sicherzustellen, dass sowohl helle als auch dunkle

Fahrzeuge erkannt werden. Hierbei ist wichtig, nur Kanten zu verknüpfen, die in die gleiche Richtung zeigen, nach innen oder außen. Wird dies nicht berücksichtigt, treten mehr Fehldetektionen auf. Wenn die verschobenen Filterantworten UND-verknüpft werden, bleiben nur die Stellen übrig, an denen alle vier Filter ausreichend stark reagiert haben (Abb. 5.6).

Erst jetzt wird das resultierende Teilbild mit dem Straßensegment maskiert, um sämtliche Objekthypothesen außerhalb der Straße zu entfernen. Dies geschieht jedoch erst nach dem Filtern, damit auch einzelne Fahrzeugkanten, die nicht im maskierten Bereich liegen, berücksichtigt werden. Ein Beispiel dafür ist das Fahrzeug rechts oben in Abb. 5.6.

Um die Hypothesen auf jeweils ein Pixel zu reduzieren, werden die verbleibenden Bereiche mittels *Non-Maxima-Suppression* (vgl. [80]) skelettiert und anschließend, um Doppeldetektionen zu vermeiden, auf ihren Abstand untereinander geprüft. Wenn zwei Hypothesen weniger als die Fahrzeugglänge bzw. -breite voneinander entfernt sind, wird diejenige mit der geringeren Intensität verworfen. Die übrigen Fahrzeughypothesen werden mit Angaben zur Position, Fahrzeugklasse, Winkel, Intensität und Straßensegment in einer Liste gespeichert.

Nachdem alle Straßensegmente einer Aufnahme bearbeitet wurden, werden die Fahrzeuge erneut auf ihren Abstand überprüft, da einige Fahrzeuge in den sich überlappenden Bereichen benachbarter Segmente erkannt und somit doppelt registriert wurden.

Als Ergebnis aus Phase eins liegt nun eine Liste der Positionen aller Fahrzeughypothesen einer Aufnahme mit eben genannten Zusatzinformationen vor.

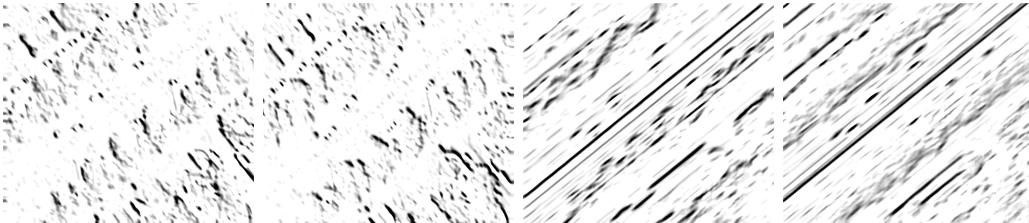


Abbildung 5.4: Die vier Filterantworten

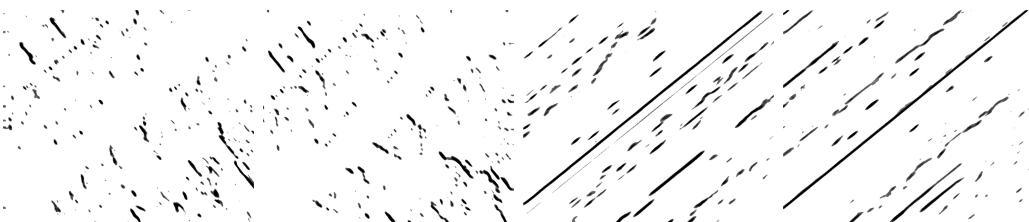


Abbildung 5.5: Die vier Filterantworten (nach Schwellwert)

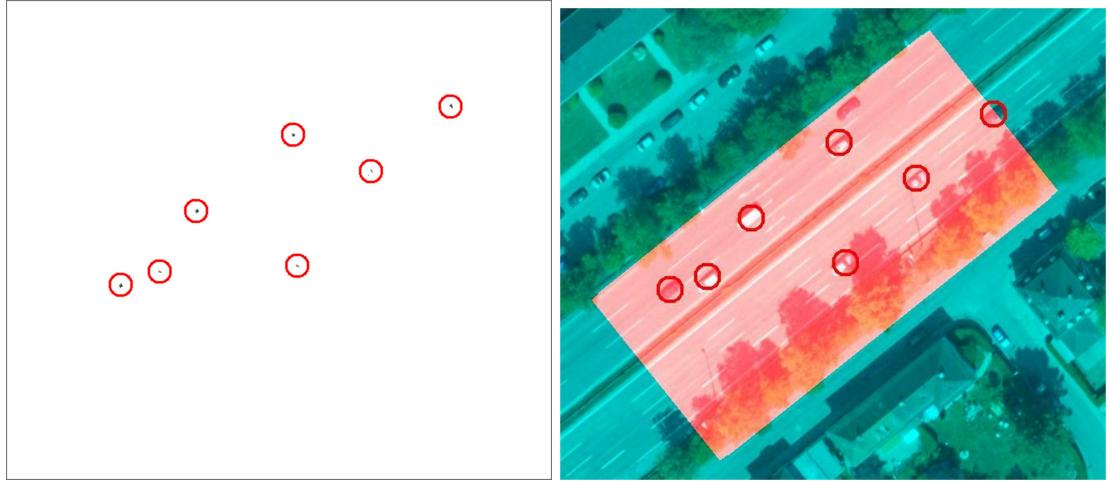


Abbildung 5.6: UND-Verknüpfung der Filterantworten

### 5.1.2 Phase zwei - Validierung der Hypothesen

Nachdem mittels einfacher (wenn auch speziell angepasster) Kantendetektion in möglichst kurzer Zeit möglichst viele Hypothesen für die Existenz von Fahrzeugen erstellt wurden, kommt ein aufwändigerer Klassifizierungsalgorithmus zum Einsatz, um diese Hypothesen zu überprüfen. Phase eins ist notwendig, da das folgende Vorgehen für die gesamte Aufnahme eine Rechenzeit von ungefähr 20 Stunden benötigt.

Der in [64] beschriebene Algorithmus liefert trotz geringer Auflösung der Satellitenfotos von ca. einem Meter (siehe [10]) gute Erkennungsraten. Dafür wurden für vorher definierte Fahrzeuge die *SIFT*-Deskriptoren (siehe [68]) ermittelt, mit welchen dann wiederum eine *Support Vector Machine* (*SVM*) trainiert wurde (siehe [22]). Die Grundidee einer *SVM* besteht darin, eine Menge von Objekten aufgrund ihrer Merkmale binär so zu klassifizieren, dass zwischen ihnen ein möglichst breiter Rand (*margin*) entsteht [19] (Abb. 5.8). Bei mehr als drei Merkmalen bzw. Dimensionen (in diesem Fall 128) entsteht eine Hyperebene, welche die Objekte in zwei Klassen unterteilt. Diese sind im gegebenen Fall die Klassen *Fahrzeug* und *Nicht-Fahrzeug*. Die Merkmale für die Klassifizierung sind die Gradienten des *SIFT*-Deskriptors.

Da der Vergleich zwischen den Deskriptoren *SIFT* und *SURF* in [13] und [99] die beiden als ähnlich zuverlässig einstuft, dem *SURF*-Deskriptor aber aufgrund der geringeren Dimensionalität (64 statt 128) und der Verwendung von Integralbildern eine schnellere Berechnungszeit zuschreibt, wurde für diese Arbeit eine *Support Vector Machine* auf *SURF*-Deskriptoren angewandt. Eine genauere Beschreibung des Vorgehens findet sich in der Diplomarbeit [41].

### Merkalsextraktion

Der *SURF*-Algorithmus dient ähnlich wie *SIFT* der schnellen Erkennung und Wiedererkennung von Bildmerkmalen (siehe [13]). Dafür wird zunächst ein Detektor angewandt,

welcher in verschiedenen Skalierungen den sogenannten *Fast-Hessian Detector* verwendet. Dieser basiert auf der Auswertung der Hesse-Matrix einer bestimmten Bildregion. Anstatt eines Gaußfilters werden Boxfilter angewandt, um die zweiten Ableitungen in  $x$ - und  $y$ -Richtung zu bestimmen. Diese bringen als Näherungen des Gaußfilters ähnliche Ergebnisse, können jedoch mithilfe von Integralbildern Pixelwerte sehr schnell summieren (siehe [100]). Als Indikator für die Einzigartigkeit eines Punktes wird die Determinante der Hesse-Matrix herangezogen. Die Detektion von Punkten ist hier jedoch nicht von Bedeutung, da die Fahrzeugmittelpunkte durch die Hypothesen aus Phase eins vorgegeben sind.

Nachdem die Punkte bestimmt sind, müssen diese durch einen geeigneten Deskriptor beschrieben werden. Hierfür wird das skalierbare Fenster um den Punkt in vier mal vier gleichgroße quadratische Regionen eingeteilt, für welche jeweils vier Kennzahlen bestimmt werden. Diese sind die Summen der Gradienten in  $x$ - und  $y$ -Richtung und die Summen ihrer Beträge. Zusammen ergibt das vier mal 16, also 64 Kenngrößen. Die Summe der Beträge ist notwendig, um Bereiche mit schnell wechselnden Helligkeiten von homogenen Bereichen zu unterscheiden, da die Gradientensumme bei beiden sehr gering ist (siehe Abb. 5.7).

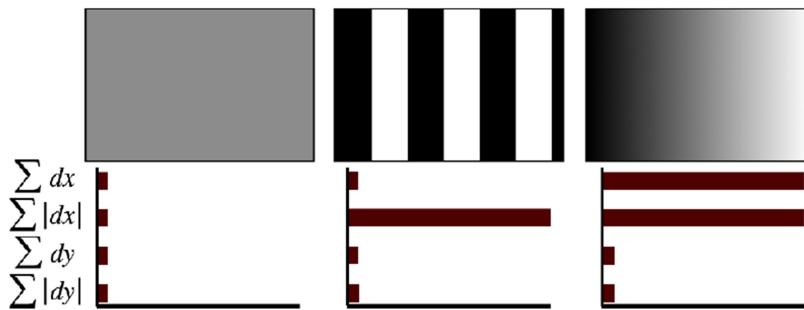


Abbildung 5.7: Kennzahlen des *SURF*-Deskriptors, Quelle [13]

Der *SURF*-Deskriptor hat einen sehr hohen Wiedererkennungswert und ist invariant gegen Skalierung und Rotation. Für Rotationsinvarianz muss vorher zusätzlich die Gradientenhauptrichtung des *Interest Point* bestimmt werden, in welcher das Fenster samt Gradienten ausgerichtet wird.

## Klassifikation

Für die Klassifikation der 64-dimensionalen Vektoren wurde eine *Support Vector Machine* implementiert. Diese muss zunächst mit einer bestimmten Menge

$$\mathcal{X} = \{x_i, y_i\} \mid x_i \in \mathbb{R}^g, y_i \in \{+1, -1\} \quad (5.1)$$

an Trainingsdaten angelernt werden. Die Trainingsdaten  $x_i$  unterscheiden sich in genau zwei Klassen  $y_i \in \{+1, -1\}$ , eine positive (*Fahrzeug*) und eine negative (*Nicht-Fahrzeug*).

Sie werden in den Merkmalsraum  $\mathbb{R}^g$  eingeordnet und versucht, linear zu trennen.

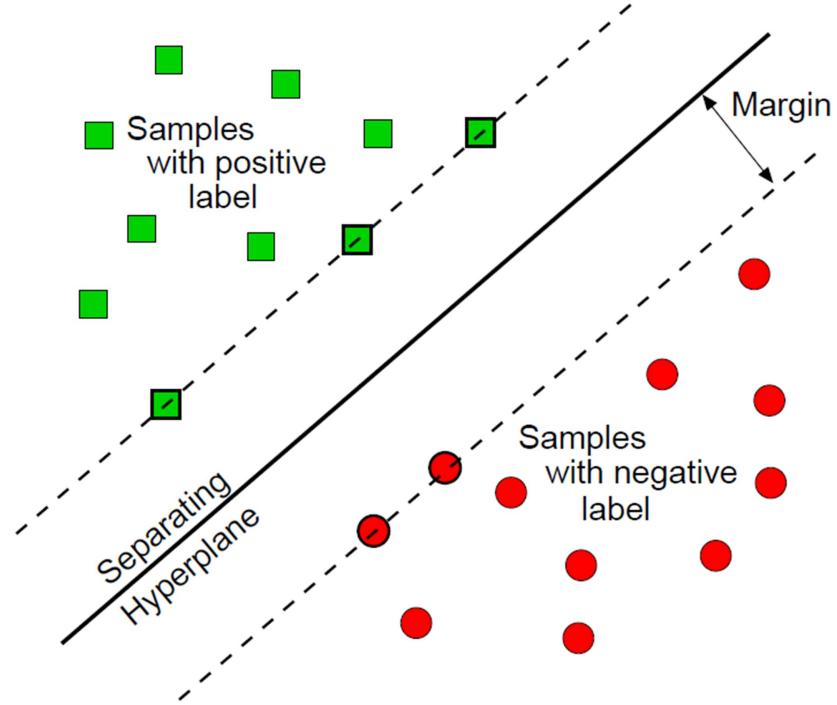


Abbildung 5.8: Merkmalsraum der SVM, Quelle [71]

Dafür muss die Gleichung einer Hyperebene  $\mathcal{H}$  mit dem Normalenvektor  $w$  und der Verschiebung  $b$  (Abb. 5.8) so optimiert werden, dass sie einen maximalen Abstand zu den naheliegendsten Trainingsdaten  $x_i$  (den *Support Vectors*) hat.

$$\mathcal{H} = \{x | \langle w, x \rangle + b = 0\} \quad (5.2)$$

Die genaue Herleitung und Lösung der Optimierungsgleichung für eine Hyperebene mit maximaler Trennspanne bzw. maximal breitem Rand sind in [71] und [19] beschrieben.

Ist die Hyperebene bestimmt, werden neue Vektoren einfach durch ihr Vorzeichen klassifiziert. Dies sagt aus, auf welcher Seite der Hyperebene sich der Vektor befindet.

$$f(x_{neu}) = \text{sign}(\langle w, x_{neu} \rangle + b) \quad (5.3)$$

Jedoch ist es nicht immer möglich, die Vektoren linear zu separieren. Auch wenn der Merkmalsraum schon relativ viele Dimensionen hat, gibt es oft keine trennende Hyperebene. In höheren Dimensionen lassen sich Datenpunkte nach dem *Theorem von Cover* (siehe [91]) einfacher trennen (Abb. 5.9). Daher werden die Vektoren durch nichtlineare Transformation  $\Phi$  in einen höherdimensionalen Merkmalsraum projiziert.

$$\Phi : \mathbb{R}^{d1} \mapsto \mathbb{R}^{d2} \mid d2 > d1 \quad (5.4)$$

$$\Phi : (g_1, g_2) \mapsto (g_1^2, \sqrt{2}g_1g_2, g_2^2) \quad (5.5)$$

Anstatt das Skalarprodukt zweier Vektoren  $p = (g_1, g_2)$  und  $q = (h_1, h_2)$  im niedrigdimensionalen Merkmalsraum  $\mathbb{R}^{d1}$  zu berechnen, wird das Skalarprodukt ihrer Abbildungen berechnet.

$$\langle \Phi(p, q) \rangle = (g_1^2, \sqrt{2}g_1g_2, g_2^2)(h_1^2, \sqrt{2}h_1h_2, h_2^2)^t \quad (5.6)$$

$$= g_1^2h_1^2 + 2g_1h_1g_2h_2 + g_2^2h_2^2 \quad (5.7)$$

$$= (g_1h_1 + g_2h_2)^2 \quad (5.8)$$

$$= \langle p, q \rangle^2 =: \mathcal{K}(p, q) \quad (5.9)$$

Wie das Beispiel zeigt, kann das Skalarprodukt im Merkmalsraum  $\mathbb{R}^{d1}$  errechnet werden, ohne die Vektoren wirklich nach  $\mathbb{R}^{d2}$  projizieren zu müssen. Die daraus resultierende Funktion  $\mathcal{K}(p, q)$  wird Kernel-Funktion genannt. Je nach Projektionsart können so verschiedene Arten von Kernelfunktionen errechnet werden.

$$\mathcal{K}(p, q) = \exp(-\gamma \|p - q\|^2) \quad (5.10)$$

Für die Klassifizierung der *SURF*-Deskriptoren wurde die radiale Basisfunktion (Gl. 5.10) verwendet, da sie am weitesten verbreitet ist und in Tests die besten Ergebnisse erzielte. Weitere Experimente mit anderen Kantenfiltern, Kernelfunktionen und Trainingssets sind in [41] näher erläutert.

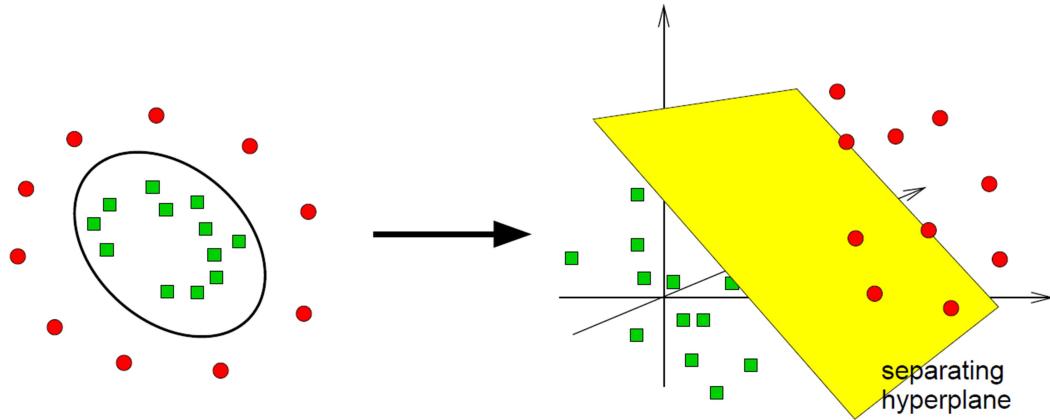


Abbildung 5.9: Merkmalsraum der SVM, Quelle [71]

## 5.2 Fehlerursachen

### 5.2.1 False Negatives

Als *False Negatives* (*FN*) werden all die Fehler bezeichnet, bei denen ein Objekt nicht erkannt wurde, obwohl eines vorhanden ist. Die Rate der *FN* bestimmt also direkt die Vollständigkeit.

$$\text{Vollständigkeit} = 100\% - \frac{\text{False Negatives}}{\text{Alle Fahrzeuge}} \cdot 100\% = \frac{\text{True Positives}}{\text{Alle Fahrzeuge}} \cdot 100\% \quad (5.11)$$

Als Hauptursachen für das Auftreten von *False Negatives* sind die folgenden drei Punkte zu nennen.

- Zum einen werden hellere Fahrzeuge oft übersehen, da ihre Farbe zur oft ebenfalls hellen Straße keinen ausreichenden Kontrast bildet (Abbildungen 5.10 c und 5.12). Abhilfe schafft dabei teilweise die Verringerung des Schwellwertes, wenn nach hellen Fahrzeugen gesucht wird. Da helle und dunkle Fahrzeuge separat gesucht werden, ist dies möglich, doch führt es gleichzeitig dazu, dass mehr *False Positives* auftreten.
- Durch ungenaue Straßenkoordinaten kann es passieren, dass Fahrzeuge gar nicht erfasst werden können, da sie sich nicht in der „Region of Interest“ befinden. Die Straße führt an den Fahrzeugen vorbei (Abbildung 5.11 b).
- Teilweise verdeckte Fahrzeuge werden oft nicht erkannt, wenn sich einige ihrer Kanten unter Bäumen, Schilderbrücken oder ähnlichen Hindernissen befinden (Abbildungen 5.11 a und c).

### 5.2.2 False Positives

Als *False Positives* (*FP*) bezeichnet man die Art von Fehler, bei dem fälschlicherweise ein Objekt erkannt wurde, wo keines vorhanden ist (vgl. Kap 5.3.1). Die Rate der *FP* bestimmt somit die Zuverlässigkeit:

$$\text{Zuverlässigkeit} = 100\% - \frac{\text{False Positives}}{\text{Alle Objekte}} \cdot 100\% \quad (5.12)$$

Die drei Hauptursachen, die dafür beobachtet werden konnten, sind folgende:

- Wenn die Straßenkoordinaten ungenau sind, werden oft rechteckige Strukturen neben der Straße als Fahrzeuge erkannt (Abbildungen 5.10 a und 5.11 a und b). Dies passiert vorwiegend in bewohnten Gebieten, da sich an Häusern neben der Straße viele rechteckige Formen finden.
- Die Schatten zwischen Fahrzeugen werden manchmal als Fahrzeug selbst erkannt (Abbildungen 5.10 b und c), da ihre dunkle Erscheinung einen stärkeren Kontrast

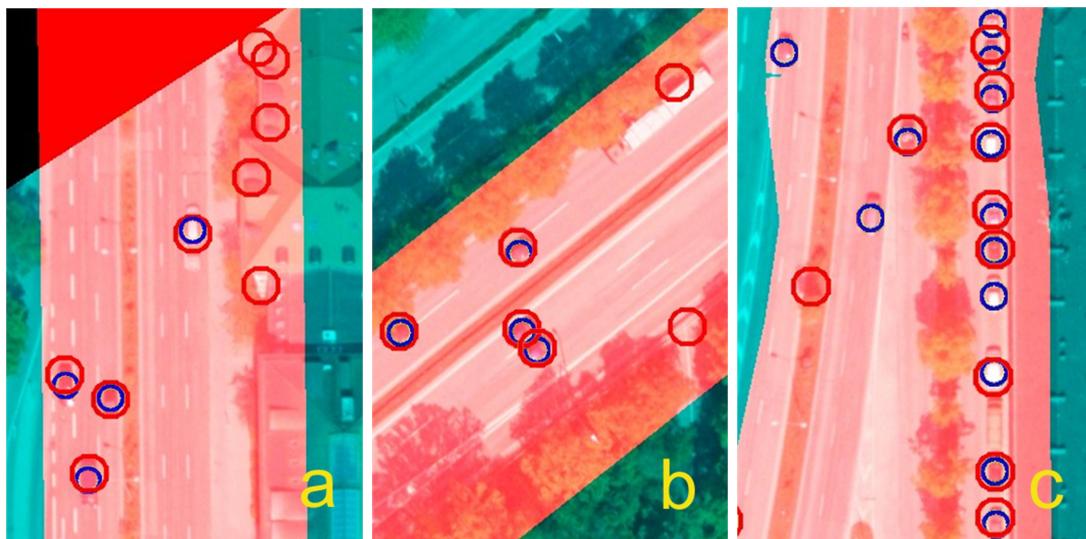


Abbildung 5.10: Ursachen für Erkennungsfehler (blau - wirkliche Fahrzeuge, rot und gelb - erkannte Fahrzeuge)

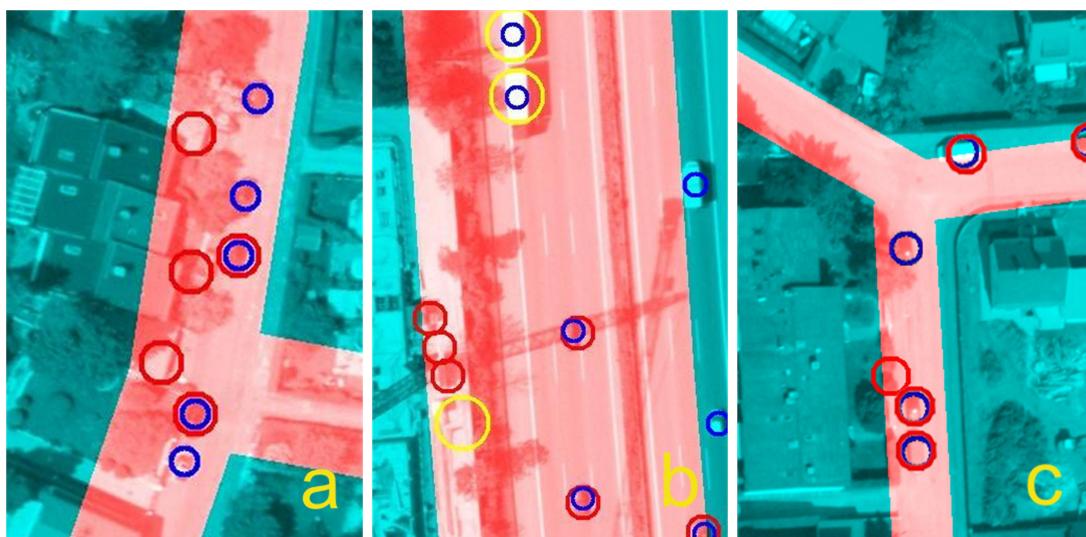


Abbildung 5.11: Ursachen für Erkennungsfehler (blau - wirkliche Fahrzeuge, rot und gelb - erkannte Fahrzeuge)

zur Straße bildet als das eigentliche Fahrzeug. Selbst wenn das Fahrzeug auch erkannt wird, verwirft es der Algorithmus, da es sich zu nah am Schatten befindet.

- Neben Fahrzeugen werfen andere Objekte ihren Schatten in den Straßenraum (Bäume, Krane), was ebenfalls zu Fehlerkennungen führen kann (Abbildungen 5.10 c und 5.11 b).



Abbildung 5.12: Kontrastproblem bei der Fahrzeugerkennung

Diejenigen Fehler, die durch ungenaue Straßeninformationen entstehen, können durch bessere Straßendatenbanken oder, wie gezeigt, eine Korrektur der Koordinaten vermieden werden. Andere Fehlerquellen, wie die fälschliche Detektion von Schatten oder anderen Objekten, werden zwar teilweise durch die *Support Vector Machine* kompensiert, jedoch wurden bislang keine Untersuchungen durchgeführt, in welchem Maße dies passiert. In der Hinsicht besteht noch Forschungsbedarf.

### 5.3 Ergebnisse

Der gesamte Algorithmus wurde auf je vier Aufnahmen von fünf verschiedenen Überflügen in München und Umgebung angewandt. Sämtliche Fahrzeuge mussten per Hand markiert und deren Positionen gespeichert werden, um eine Datengrundlage als *Ground Truth* zu erhalten. Als ein Gütemaß wird das Verhältnis von richtig erkannten Objekten (*True Positives TP*) zu wirklich vorhandenen Fahrzeugen gewertet, welche im Folgenden als Vollständigkeit oder *TP*-Rate bezeichnet wird. Das zweite verwendete Maß stellt das Verhältnis der fälschlich erkannten Objekte (*False Positives FP*) zu den tatsächlich vorhandenen Fahrzeugen dar. Es wird nachfolgend Zuverlässigkeit oder *FP*-Rate genannt. Damit ist jedoch die negierte Zuverlässigkeit gemeint, d.h. je höher die *FP*-Rate ist, desto unzuverlässiger arbeitet die Erkennung.

Die Support Vector Machine wurde mit je 2500 und 6500 positiven und negativen Trainingsdaten angelernt. Beide Versionen wurden getestet, und obwohl die *SVM* mit geringerem Datensatz wesentlich mehr *False Positives* erkannte, entstand gleichzeitig das Problem, dass zu viele *True Positives* verworfen wurden. Die ausführlichere Auswertung dafür findet sich ebenfalls in [41]. Hier soll nur auf den größere Datensatz mit den 7000 Trainingsbeispielen eingegangen werden.

Es wurde versucht, verschiedene Szenarien abzudecken. Der erste Datensatz wurde unter optimalen Bedingungen erhoben. Die Autobahn A8 wurde bei fließendem Verkehr und sonnigem Wetter beobachtet. Datensatz zwei weist ähnliche Bedingungen auf, nur stauen sich die Fahrzeuge. Der dritte Datensatz zeigt auch die A8, nur wurde er bei

Schnee aufgenommen. Die Datensätze vier und fünf zeigen Überflüge über die Innenstadt mit vielen Nebenstraßen und Verdeckungen.

### 5.3.1 Datensatz 1 - Autobahn A8

Für den ersten Datensatz wurden neben der Autobahn keine weiteren Straßen berücksichtigt. Alle vier Aufnahmen wurden mit verschiedenen Erkennungsschwellwerten (siehe Kap. 5.1.1) getestet, wobei sich gut erkennen lässt, dass man bei der Erkennung immer um einen Kompromiss zwischen Vollständigkeit und Zuverlässigkeit eingeht, der mit dem Schwellwert gesteuert wird. Je vollständiger die Erkennung ist, desto mehr Fehldetections treten auf, während eine hohe Zuverlässigkeit auch mehr nicht erkannte Fahrzeuge zur Folge hat. Somit ist es sinnvoll, die Vorselektion mit einem möglichst niedrigen Schwellwert anzusetzen, da die *False Positives* von der *Support Vector Machine* verworfen werden können, nicht erkannte Fahrzeuge jedoch nicht betrachtet werden.

Die Abbildungen 5.13 und 5.14 zeigen die durchschnittlichen Erkennungsraten aller vier Aufnahmen. Die blauen Symbole in Abb. 5.13 zeigen die Vollständigkeit (*True Positives*), während die roten die Prozentzahl der *False Positives* darstellen. An diesem Beispiel lässt sich gut erkennen, dass die Straßenkorrektur (Kreuze  $\times$ ) die Detektionsrate im Vergleich zur ursprünglichen Erkennung (Plus +) leicht verbessert. Die Validierung durch die *Support Vector Machine* verringert zwar die Vollständigkeit (blaue Rauten  $\diamond$ ) leicht, doch wird ebenfalls die Rate der *False Positives* (rote Rauten  $\diamond$ ) besonders im Bereich niedriger Schwellwerte enorm unterdrückt. So ergibt sich bei einer Vollständigkeit von 88 Prozent eine *FP*-Rate um die zehn Prozent. Durch die Ergebnisse der *SVM* wurde ein Polynom höheren Grades approximiert, um den Verlauf besser zu visualisieren. Abbildung 5.14 zeigt die Detektionsraten mit der zugehörigen Standardabweichung. Während diese im vorderen Bereich der *True Positives* nur vier Prozentpunkte beträgt, streut sie sehr stark bei den *False Positives*. Das liegt daran, dass sich die *FP*-Rate auf die Anzahl der wirklich vorhandenen Fahrzeuge bezieht, welche in den vier Aufnahmen zwischen 68 und 100 liegt, also vergleichsweise gering ist. Das hat zur Folge, dass bei hoher Sensibilität die Fehldetections stärker streuen.

### 5.3.2 Datensatz 2 - Autobahn A8, Stau

Der zweite Datensatz (Abb. 5.13 und 5.14) wurde, wie oben erwähnt, unter fast gleichen Bedingungen aufgenommen, nur staut sich der Verkehr auf einer Seite der Fahrbahn. Die Detektionsraten (Vollständigkeit) sind geringer als im ersten Datensatz, doch wird auch hier durch die *SVM* die *FP*-Rate stark verringert. So ergeben sich Vollständigkeiten von etwa 70 Prozent bei Fehlerraten (*FP*) um die zehn Prozent. Die geringere *TP*-Rate hängt damit zusammen, dass viele der Fahrzeuge dicht gedrängt stehen. Bei hoher Sensibilität wird oft zwischen zwei Fahrzeugen eine zusätzliche Hypothese erzeugt. Diese liegt zu nah an einem der tatsächlichen Fahrzeuge, was in einigen Fällen dazu führt, dass die wahre Fahrzeughypothese verworfen wird.

Aufgrund der größeren Anzahl von Fahrzeugen (200-300) fällt auch die Streuung der *TP*- und *FP*-Raten geringer aus (Abb. 5.14).

### 5.3.3 Datensatz 3 - Autobahn A8, Schnee

Datensatz Nummer drei bringt den Detektionsalgorithmus klar an seine Grenzen. Wie in Abbildung 5.17 und 5.18 zu erkennen ist, werden bei der Vollständigkeit ( $TP$ ) im günstigsten Fall zwei Drittel erreicht, während sich die *False Positives* noch weit darüber befinden. Die *False Positives* liegen oft über 100 Prozent, da sie relativ zur Anzahl der wirklich vorhandenen Fahrzeuge berechnet werden, welche in manchen Fällen geringer ist. Bei höheren Schwellwerten wird immer noch die Hälfte richtig erkannt, während die  $FP$ -Rate unter 20 Prozent liegt.

Da die Gebiete neben den Straßen größtenteils mit Schnee bedeckt sind, wird die Blende automatisch reguliert, was die Straßenoberfläche erheblich dunkler erscheinen lässt. Dadurch verringert sich auch der Kontrast zwischen Fahrzeug und Fahrbahnoberfläche (siehe Abbildung 5.12), was wiederum die  $TP$ -Rate verringert, Fahrzeuge werden übersehen. Gleichzeitig werden Sperrflächen als Fahrzeuge erkannt, da sie starke Kanten teilweise in Fahrzeugform aufweisen.

Hier zeigt sich besonders deutlich, wie sich die Straßenkorrektur auf die Erkennungswerte auswirkt (Plus + und Kreuz  $\times$ ). Auch ist bei diesem Datensatz wieder die Auswirkung der geringen Anzahl von Fahrzeugen erkennbar. Die Streuung der *False Positives* ist sehr groß, da sich geringe Abweichungen stark auf die Erkennungsrate auswirken.

### 5.3.4 Datensatz 4 - Stadtgebiet München, Nebenstraßen

Der vierte Datensatz zeigt das Münchener Stadtgebiet entlang der Autobahn A95 in den Stadtteilen Fürstenried und Sendling. Bei diesem Datensatz wurden sämtliche Nebenstraßen berücksichtigt und auf Fahrzeuge untersucht. Die Kurve der Vollständigkeit ( $TP$ ) steigt zunächst durch Straßenkorrektur um fünf bis sechs Prozentpunkte auf maximal 74 Prozent an. Nach der Verifikation durch die *SVM* bleibt bei höheren Schwellwerten eine  $TP$ -Rate von 65 Prozent übrig, während die *False Positives* auf 15 bis 20 Prozent sinken (siehe Abbildung 5.19 und 5.20).

Die starke Streuung und vergleichsweise schlechte Detektionsrate sind mit der Ungenauigkeit der Straßenkoordinaten zu erklären. Auch wenn deren Position durch einen Algorithmus korrigiert wird, konvergiert die Optimierung nicht immer richtig. In manchen Fällen verschlechtert sich die Straßenposition sogar, was an denjenigen Stellen weniger *True Positives* und mehr *False Positives* zur Folge hat. Die dort falsch detektierten Objekte werden zwar größtenteils durch die *SVM* verworfen, doch auch nur bis zu einem bestimmten Grad.

### 5.3.5 Datensatz 5 - Stadtgebiet München, Nebenstraßen

Auch der fünfte Datensatz zeigt das Münchener Stadtgebiet und bezieht sämtliche Nebenstraßen in die Detektion ein (5.21). Auch hier hebt die Straßenkorrektur die Detektionsrate zunächst leicht an. Nach der Evaluierung durch die *SVM* wird diese jedoch wieder auf ca. 60 Prozent gesenkt. Die  $FP$ -Rate jedoch sinkt dabei enorm von weit über 100 Prozent auf weniger als 40, bei höheren Schwellwerten zwischen zehn und 20 Prozent.

Die Tatsache, dass dieser Datensatz mehr Fahrzeuge enthält als Nummer 4 führt wieder dazu, dass die Anzahl der *False Positives* sowie deren Streuung geringer als im vorigen Datensatz ist. Wieder einmal erweist sich die *Support Vector Machine* als signifikante Verbesserung der Fehlerkennungsrate.

## 5.4 Zusammenfassung

Die Kombination eines schnellen und eines aufwendigen Algorithmus erreicht je nach Szenario Erkennungsraten zwischen 60 und 90 Prozent, während die *False Positives* auch in extremen Situationen selten die Marke von 20 Prozent überschreiten, in den meisten Fällen unter zehn Prozent liegen. In den Abbildungen 5.13 bis 5.22 lässt sich erkennen, dass die Auswahl des Schwellwertes (*Threshold*) für die Kantenfilter entscheidend für die Güte der Erkennung ist. Weder darf der Wert zu hoch sein, da sonst kaum Fahrzeuge detektiert werden, noch sollte er zu niedrig gewählt werden, was zu viele Falscherkennungen (*FP*) zur Folge hat. Der optimale Bereich liegt bei den hier verwendeten 8-Bit-Grauwertbildern etwa zwischen 25 und 40. Bei kontrastschwachen Aufnahmen wie in Datensatz 3 übersteigt die Vollständigkeit erst bei sehr niedrigem *Threshold* (15) die 60-Prozent-Marke, was gleichzeitig aber sehr viele *False Positives* verursacht durch Bildrauschen nach sich zieht. Auch in solchen Extremfällen muss ein höherer Wert mit geringerer *TP*-Rate in Kauf genommen werden, da die geringe Zuverlässigkeit das Ergebnis zu stark verfälscht.

Allgemein ist zu sagen, dass der Algorithmus bei Autobahnaufnahmen sehr hohe *TP*-Raten zwischen 80 und 90 Prozent aufweist. Schwieriger wird die Erkennung bei mehr Verkehr oder in Stadtgebieten, wo die heterogene, kantenreiche Umgebung viele *False Positives* hervorbringt. Die bei weitem schlechtesten Ergebnisse liefert der Algorithmus für den Schnee-Datensatz. Die Belichtung wurde aufgrund der hellen Umgebung so stark herunter geregelt, dass sich die Objekte auf den Straßen kaum noch von Hintergrund abheben.

Nach [26] sind Erkennungsraten um die 70 Prozent für die Verkehrslageerfassung ausreichend. Wenn auch von den meisten stationären Erfassungssystemen, insbesondere von Induktionsschleifen, wesentlich genauere Ergebnisse zu erwarten sind, sind doch die Vorteile der luftgestützten Erfassung in Kapitel 1 dargelegt.

Durch die Vorauswahl mittels Kantenfilter kann die Berechnungszeit stark minimiert werden, was bis zu einem gewissen Maße Quasi-Echtzeitbetrieb erlaubt. Das heißt, dass die Auswertung der Aufnahmen während des Überfluges stattfinden kann. Die Berechnung einer Aufnahme dauert je nach Anzahl der Straßenelemente und Fahrzeuge zwischen 1,2 und 7,5 Sekunden (keine Straßenkorrektur).

Da die Aufnahme eines Flugstreifens zwischen 30 und 160 Sekunden dauern kann und die anschließende Wende weitere 200 bis 600 Sekunden, ist es in den meisten Fällen möglich, den aufgenommenen Datensatz während des Überfluges und der anschließenden Wende komplett zu berechnen. In Abbildung 5.23 sind Zeiten aus drei verschiedenen Flugkampagnen dargestellt. Die Balken 1 bis 3 zeigen die Flug- (grün und rot) und Rechenzeiten (blau) der ersten beiden Datensätze aus Kapitel 5.3. Da nur die Segmente der Autobahn berücksichtigt und pro Überflug nur 40 bis 50 Aufnahmen gemacht werden,

## 5 Fahrzeugerkennung

können diese sogar schon in der Überflugzeit ausgewertet werden. Der Computer braucht nur einen Bruchteil der ihm zur Verfügung stehenden Zeit.

Die Balken 4 bis 6 zeigen die Zeiten einer Flugkampagne über Stadtgebiet. Erstens sind die Aufnahme- (grün) und Wendezeiten (rot) im Schnitt kürzer als in den ersten Beispielen. Gleichzeitig werden die Aufnahmen mit höherer Frequenz erfasst, was pro Überflug bis zu 73 Aufnahmen liefert. Drittens erhöht sich die Rechenzeit pro Aufnahme auf durchschnittlich 7,5 Sekunden, da alle Nebenstraßen mit einbezogen werden. Dennoch reicht die Gesamtflugzeit aus, um alle Aufnahmen zu berechnen.

Im letzten Fall (Balken 7 bis 9) handelt es sich um den Datensatz 5 aus dem Kapitel 5.3. Bei diesem Beispiel wurden sehr viele Aufnahmen pro Überflug gemacht (bis zu 90), während gleichzeitig sehr kurze Wenden geflogen wurden (150 bis 300 Sekunden). Da wiederum sämtliche Straßen berücksichtigt werden, dauert die Berechnungszeit sehr lange, im ungünstigsten Fall doppelt so lange wie die Flugzeit. Verwendet man zusätzlich die Straßenkorrektur, benötigt der Algorithmus in etwa die dreifachen Rechenzeiten. In dem Falle wäre Echtzeitbetrieb nur bei Autobahnüberflügen möglich.

Wie oben erläutert, hängt die Echtzeitfähigkeit der Software von verschiedenen Parametern ab. Positiv auf die Möglichkeit der Echtzeitberechnung wirken sich somit eine lange Flugzeit (insbesondere Wendezeit), geringe Aufnahmefrequenz und geringe Anzahl von Straßen bzw. Fahrzeugen aus. Die durchgeführten Tests beziehen sich ausschließlich auf die Vorselektion durch Kantenfilter. Da sämtliche nachfolgenden Teilalgorithmen der Verkehrserfassung (*SVM* und *Tracking*) erheblich weniger Zeit beanspruchen und parallel ablaufen können, sind deren Berechnungszeiten nicht von Belang.

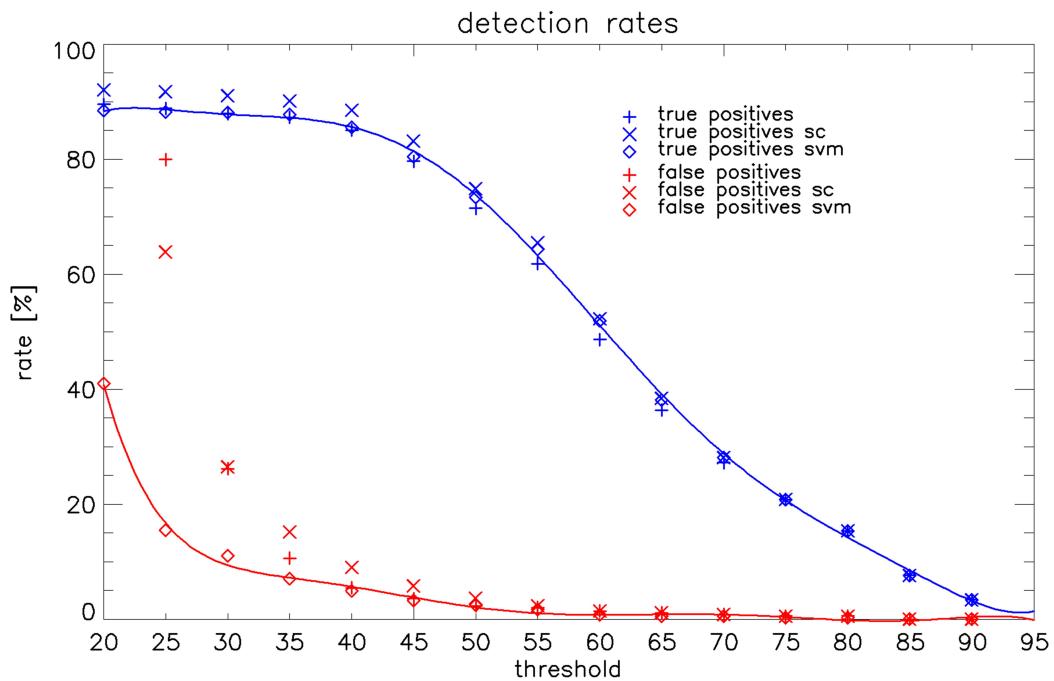


Abbildung 5.13: Detektionsraten Datensatz 1

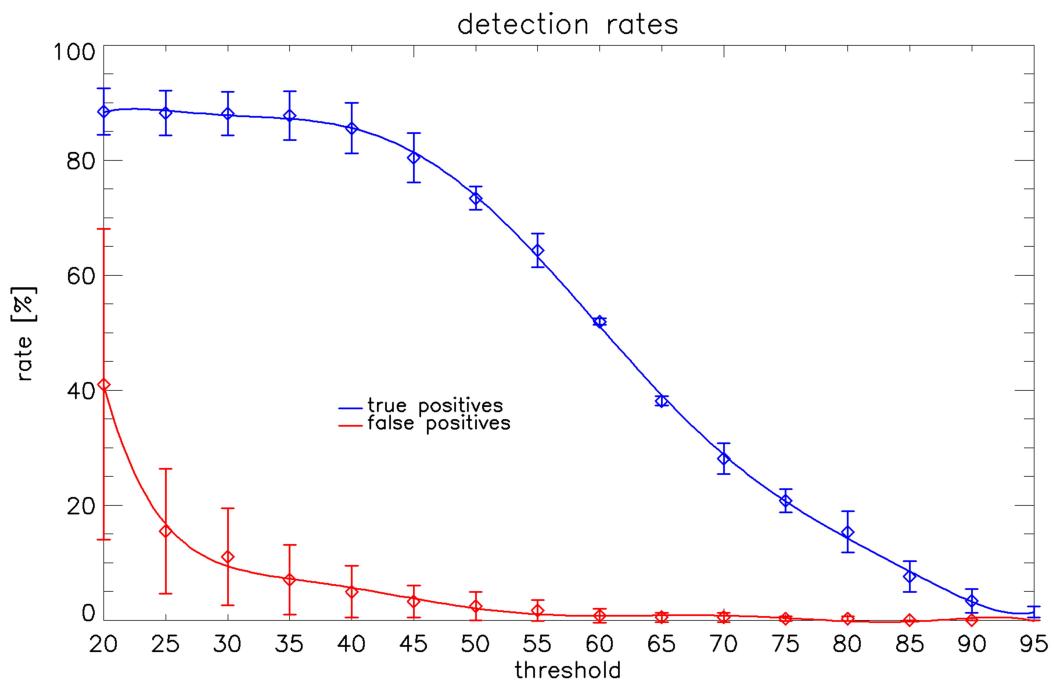


Abbildung 5.14: Detektionsraten Datensatz 1

## 5 Fahrzeugarkennung

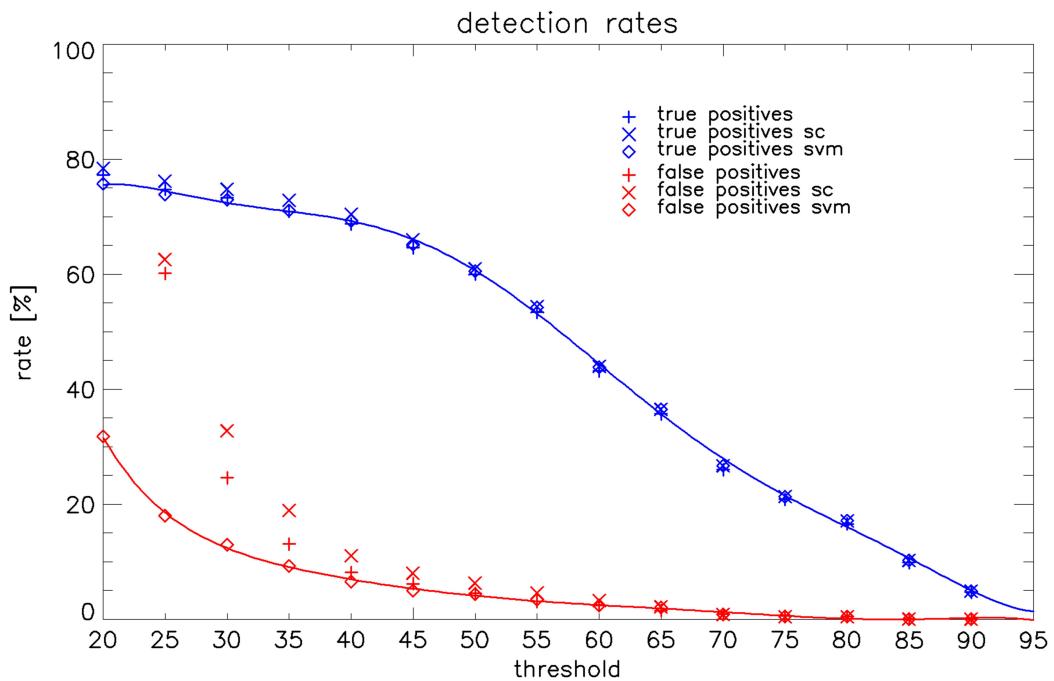


Abbildung 5.15: Detektionsraten Datensatz 2

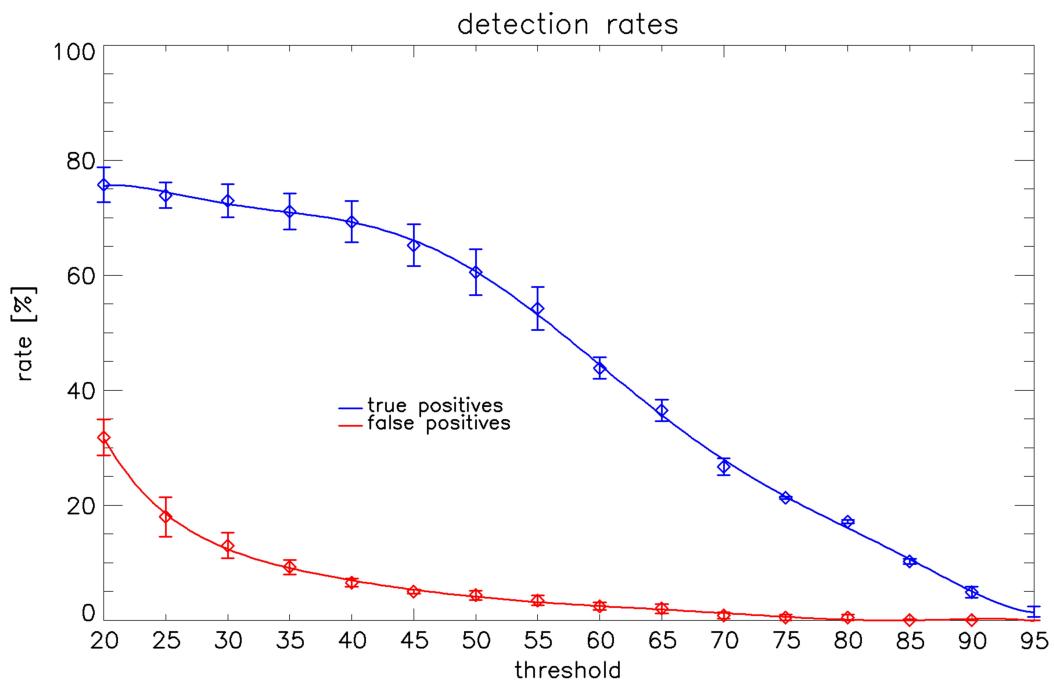


Abbildung 5.16: Detektionsraten Datensatz 2

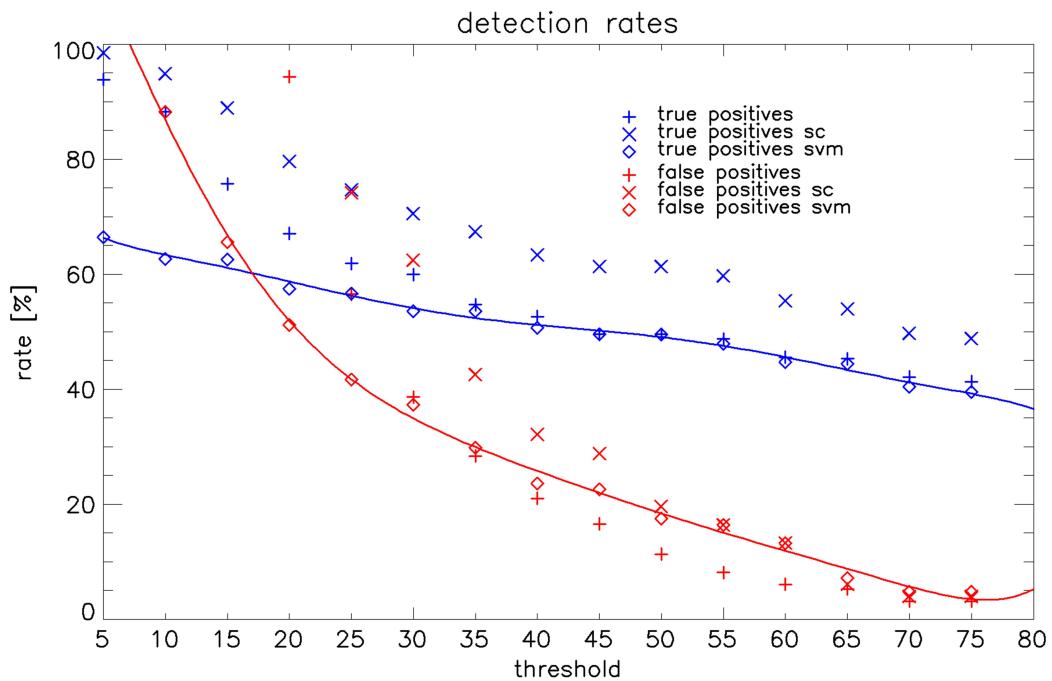


Abbildung 5.17: Detektionsraten Datensatz 3

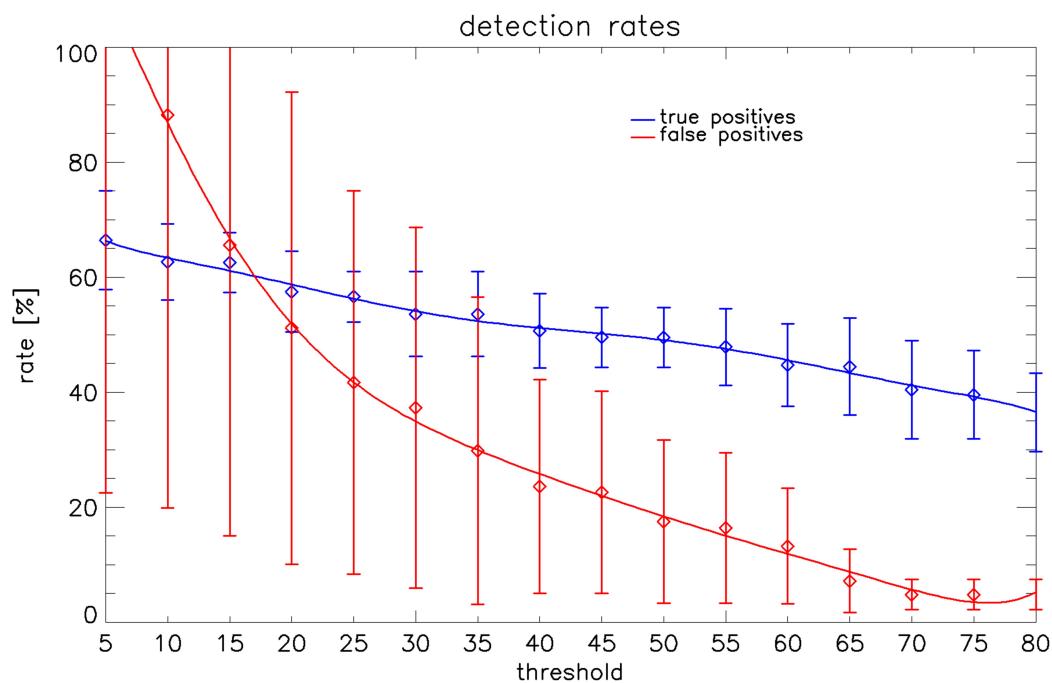


Abbildung 5.18: Detektionsraten Datensatz 3

## 5 Fahrzeugarkennung

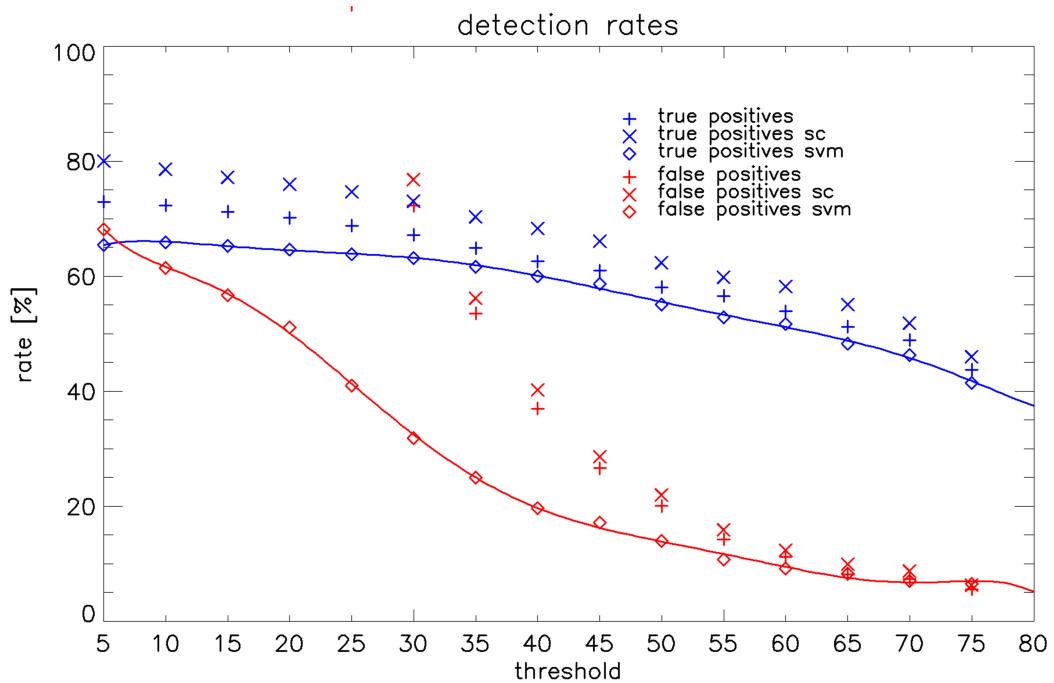


Abbildung 5.19: Detektionsraten Datensatz 4

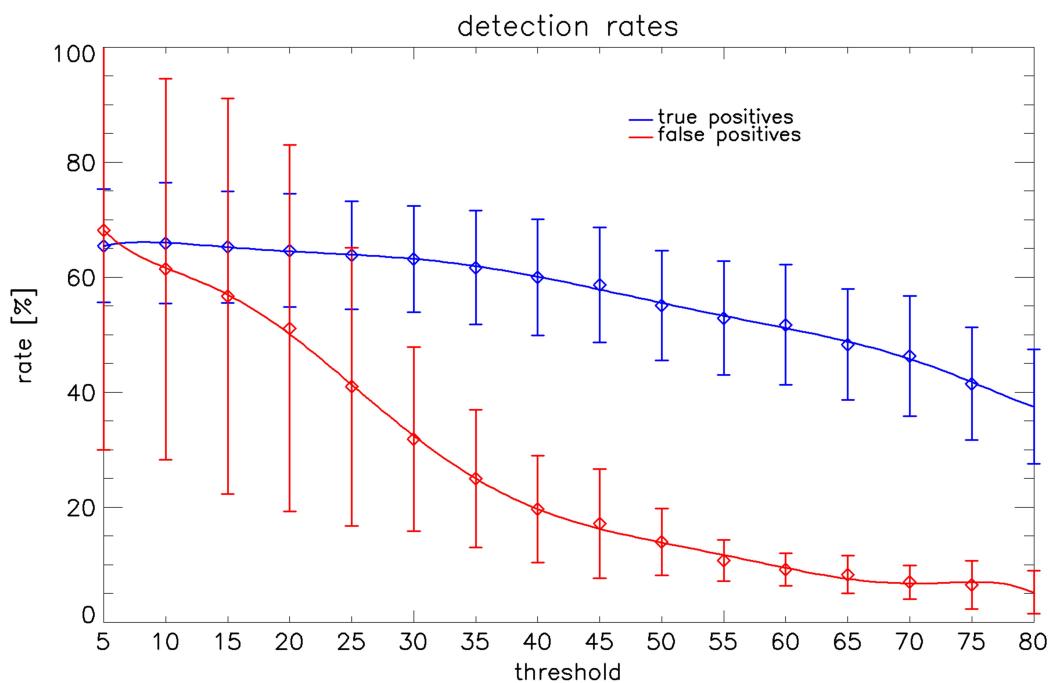


Abbildung 5.20: Detektionsraten Datensatz 4

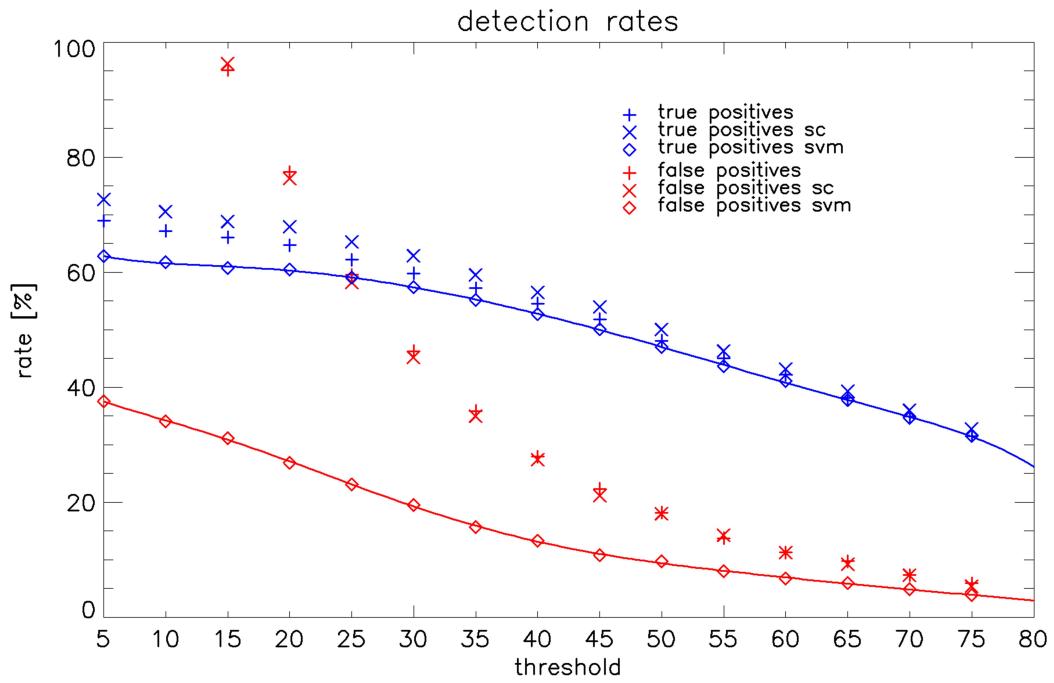


Abbildung 5.21: Detektionsraten Datensatz 5

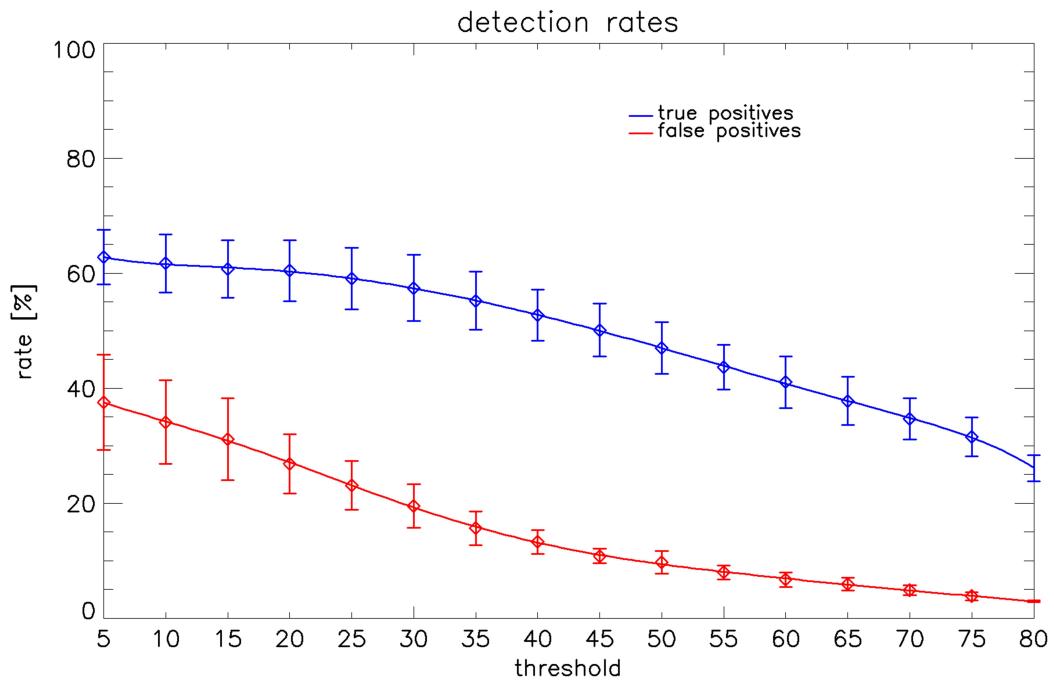


Abbildung 5.22: Detektionsraten Datensatz 5

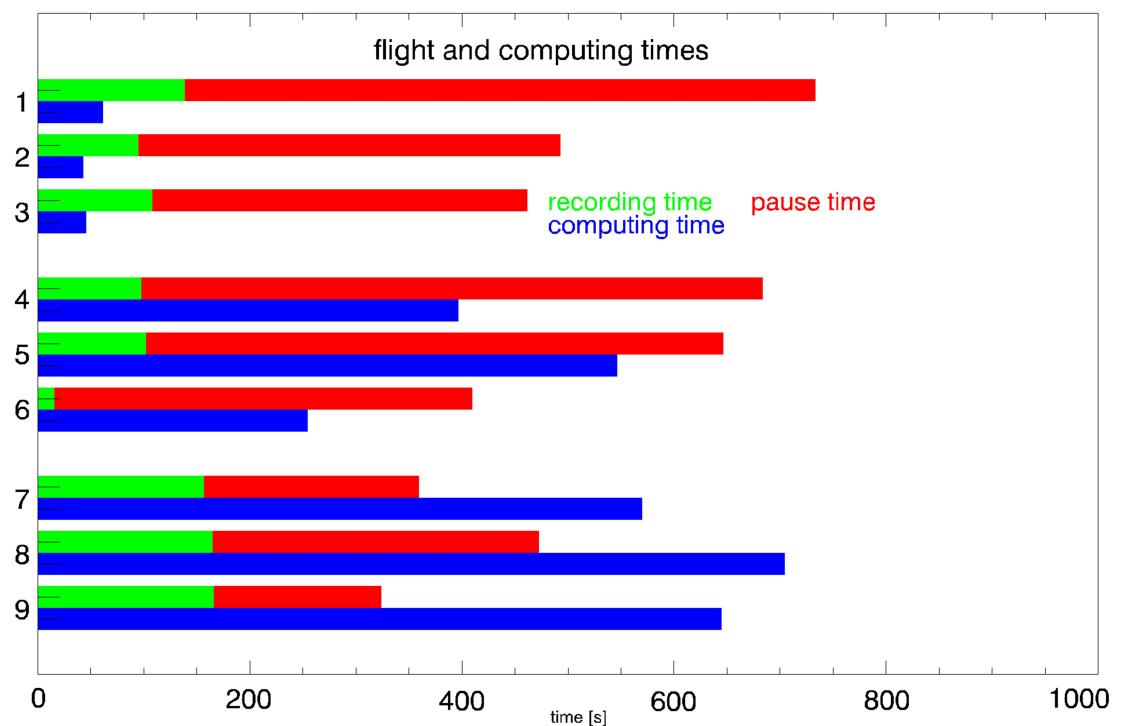


Abbildung 5.23: Flug- und Rechenzeiten der Fahrzeugerkennung

# 6 Tracking

Die Art von Verkehrsdaten, welche im *ARGOS*-Projekt in erster Linie erhoben werden sollen, sind die Anzahl der Fahrzeuge pro Straßensegment und deren Durchschnittsgeschwindigkeit. Das Zählen bzw. die Detektion der Fahrzeuge wurde in Kapitel 5 erläutert. Die Geschwindigkeiten sollen durch die Wiedererkennung in Folgebildern ermittelt werden. Dafür existieren in der Literatur viele Ansätze. Am naheliegendsten ist dabei die Wiedererkennung durch verschiedene Matchingverfahren wie Kreuzkorrelation, *Sum of Squared Differences* oder den Vergleich aufwändigerer Deskriptoren wie *SIFT* [68] oder *SURF* [13]. In [85] wird zum Beispiel in Fahrtrichtung per Korrelationskoeffizient nach ähnlichen Objekten gesucht. In [67] wird ebenfalls ein Ansatz vorgestellt, der auf Wiedererkennung durch Matching setzt. Zusätzlich wird jedoch ein einfaches Bewegungsmodell erstellt, welches aus zwei Bildern die Positionsverhersage für die dritte Aufnahme trifft. Die wahrscheinlichste Kombination aus drei Aufnahmen wird dann als „Trajektorie“ akzeptiert. Diese Methode eignet sich sehr gut für Bursts von drei Aufnahmen, ist jedoch bei nur zwei Bildern allein auf das verwendete *Shape Matching* angewiesen.

In [12], [15] und [93] werden Bewegungen und Positionen der zu trackenden Objekte durch Kalman- oder Partikelfilter geschätzt. Der optimal wahrscheinliche Zustand eines Objektes wird aus einer Kombination aus Vorhersage und Beobachtung errechnet. Diese Methoden liefern zuverlässige Trajektorien, benötigen jedoch eine gewisse Anzahl an Schritten, um die Bewegungsparameter optimal zu schätzen. Aufgrund der Unterbrechungen bei der *ARGOS*-Bildaufnahme sind diese Filter im gegebenen Fall nicht als Tracker einsetzbar.

Die Quellen [39], [78] und [84] befassen sich damit, bewegte Objekte zu erkennen. In [84] sollen mittels Differenzbildverfahren die beweglichen Fahrzeuge detektiert werden, während in den Quellen [39] und [78] der Optische Fluss verwendet wird. Für sich bewegende Fahrzeuge sind das zweifellos sehr zuverlässige Algorithmen, doch stoßen sie an ihre Grenzen, sobald stehende Fahrzeuge auftreten, was an Ampeln oder in Staus der Fall ist.

## 6.1 Ansatz

Aufgrund kurzer Bildfolgen und langer Pausen sollten Fahrzeuge nur in benachbarten Aufnahmen miteinander assoziiert werden. Ein vielversprechender Ansatz, korrespondierende Objekte oder Merkmale aus zwei Bildern miteinander zu matchen, wird in [90] erläutert. Darin werden die Abstände aller Objekte der ersten Aufnahme zu sämtlichen Objekten der zweiten Aufnahme berücksichtigt. Eine Erweiterung des Algorithmus unter Berücksichtigung der Ähnlichkeit der Merkmale wurde später in [81] vorgestellt.

## 6 Tracking

Der dort beschriebene Algorithmus basiert auf der Singulärwertzerlegung einer Abstands- und Ähnlichkeitsmatrix. Gegeben seien zwei Aufnahmen  $I$  und  $J$  mit  $m$  Objekten  $I_i$  ( $i = 1..m$ ) und  $n$  Objekten  $J_j$  ( $j = 1..n$ ). Diese sollen in eine Eins-zu-Eins-Beziehung gesetzt werden. Um dies zu erreichen, wird eine Nachbarschaftsmatrix  $\mathbf{G}$  erstellt, deren Elemente  $\mathbf{G}_{ij}$  die Gauß-gewichteten Abstände der Objekte zueinander enthalten.

$$\mathbf{G}_{ij} = e^{-r_{ij}^2/2\sigma^2} \quad (6.1)$$

$$r_{ij}^2 = ||I_i - J_j|| \quad (6.2)$$

Der Parameter  $r_{ij}$  enthält die euklidischen Abstände der Objekte  $I_i$  und  $J_j$ . Die Werte für  $\mathbf{G}_{ij}$  liegen zwischen 1 und 0 und werden mit zunehmendem Abstand der Objekte kleiner. Der Parameter  $\sigma$  kontrolliert nach [81] den Grad der Interaktion zwischen den beiden Objektmengen, wobei ein kleines  $\sigma$  mehr lokale und ein großes  $\sigma$  mehr globale Interaktionen berücksichtigt. Allgemein gesagt, je größer  $\sigma$  ist, desto flacher fällt die Verteilungsdichte-funktion ab und desto größere Verschiebungen zwischen zwei assoziierten Objekten  $I_i$  und  $J_j$  werden akzeptiert.

Der nächste Schritt ist, die Nachbarschaftsmatrix  $\mathbf{G}$  einer Singulärwertzerlegung zu unterziehen.

$$\mathbf{G} = \mathbf{T} \mathbf{D} \mathbf{U}^t \quad (6.3)$$

Die Matrizen  $\mathbf{T}$  und  $\mathbf{U}$  sind Orthogonalmatrizen, und die Diagonalmatrix  $\mathbf{D}$  enthält die positiven Singulärwerte der Matrix  $\mathbf{G}$  in absteigender Reihenfolge.

Als nächstes wird die Matrix  $\mathbf{D}$  nach  $\mathbf{E}$  umgewandelt, indem die Singulärwerte auf der Hauptdiagonalen durch Einsen ersetzt werden.

Dann wird erneut das Produkt aus den drei Matrizen errechnet.

$$\mathbf{P} = \mathbf{T} \mathbf{E} \mathbf{U}^T \quad (6.4)$$

Die Matrix  $\mathbf{P}$  ist jetzt eine Art Abbildungsmatrix der minimalen quadrierten Distanzen in  $\mathbf{G}$ . Für gute Paarungen zwischen allen Objekten in den aufeinanderfolgenden Bildern stehen hohe Werte in  $\mathbf{P}$  und für schlechte Paarungen niedrige. Wenn ein Element  $\mathbf{P}_{ij}$  gleichzeitig das größte in Zeile und Spalte ist, werden die Elemente  $I_i$  und  $J_j$  als zusammengehörig angenommen.

Um nicht allein den Abstand zweier Objekte als Indikator für deren Korrespondenz zu nehmen, werden die Matrixeinträge  $\mathbf{G}_{ij}$  zusätzlich mit einem Korrelationskoeffizienten gewichtet. Ein Fenster von beispielsweise 30 mal 30 Pixeln um jedes Objekt  $I_i$  wird mit den Umgebungen aller Objekte  $J_j$  gematcht. Der Korrelationskoeffizient der Farbwerte aller Objektpaare  $I_i$  und  $J_j$  wird als  $\mathbf{C}_{ij}$  bezeichnet und ebenfalls Gauß-gewichtet in die Matrix übernommen.

$$\mathbf{G}_{ij} = e^{-r_{ij}^2/2\sigma^2} \cdot e^{-(C_{ij}-1)^2/2\gamma^2} \quad (6.5)$$

$$C_{ij} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x}^2)} \cdot \sqrt{\sum_{i=1}^n (y_i - \bar{y}^2)}} \quad (6.6)$$

Der linke Faktor in 6.5 ist noch derselbe wie in Gleichung 6.1. Der Parameter  $\gamma$  im rechten Faktor hat hier die gleiche Funktion wie  $\sigma$  im vorher beschriebenen Fall. Er legt fest, wie flach die Gauß-Kurve mit abnehmender Ähnlichkeit abfällt.

## 6.2 Ergebnisse

### 6.2.1 Erste Tests

Die Implementierung wurde zunächst auf eine Straße in zwei aufeinanderfolgenden Aufnahmen angewandt. In den Aufnahmen wurden 49 und 53 Objekte detektiert. Aus diesen Ergebnissen lassen sich 38 korrekte Paarungen bilden. Die Abbildungen 6.1 und 6.2 zeigen die Vollständigkeiten für die Testaufnahme in Abhängigkeit der Parameter  $\sigma$  und  $\gamma$ . Gleichzeitig zeigen die schwarzen Balken in Abbildung 6.1 den Anteil der False Positives an.

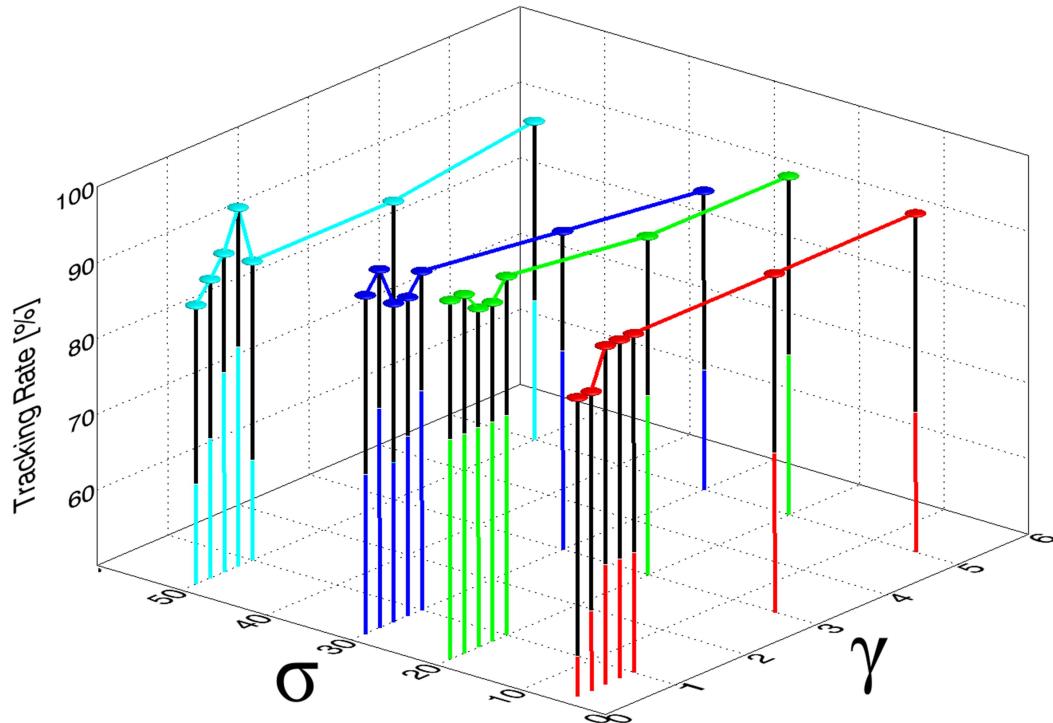


Abbildung 6.1: Vollständigkeit des Trackings in Abhängigkeit von  $\sigma$  und  $\gamma$

In Abbildung 6.2 sind die gleichen Graphen auf die xy-Ebene projiziert. Wie man sieht, ergeben sich die besten Vollständigkeiten etwa im Bereich  $0.4 \leq \gamma \leq 1.0$  und

## 6 Tracking

$20 \leq \sigma \leq 30$  (grüne und blaue Kurve). Zwar ist die Trackingrate auch bei  $\sigma = 5$  vergleichbar hoch, doch treten in diesem Bereich auch sehr viele Fehlzuordnungen auf, was durch den oberen schwarzen Teil der Balken dargestellt ist. Maximal wurden 37 von den 38 möglichen richtigen Paarungen erkannt, was einer Vollständigkeit von 97,4 Prozent bei einer Zuverlässigkeit von 84,1 Prozent (sieben *False Positives*) entspricht.

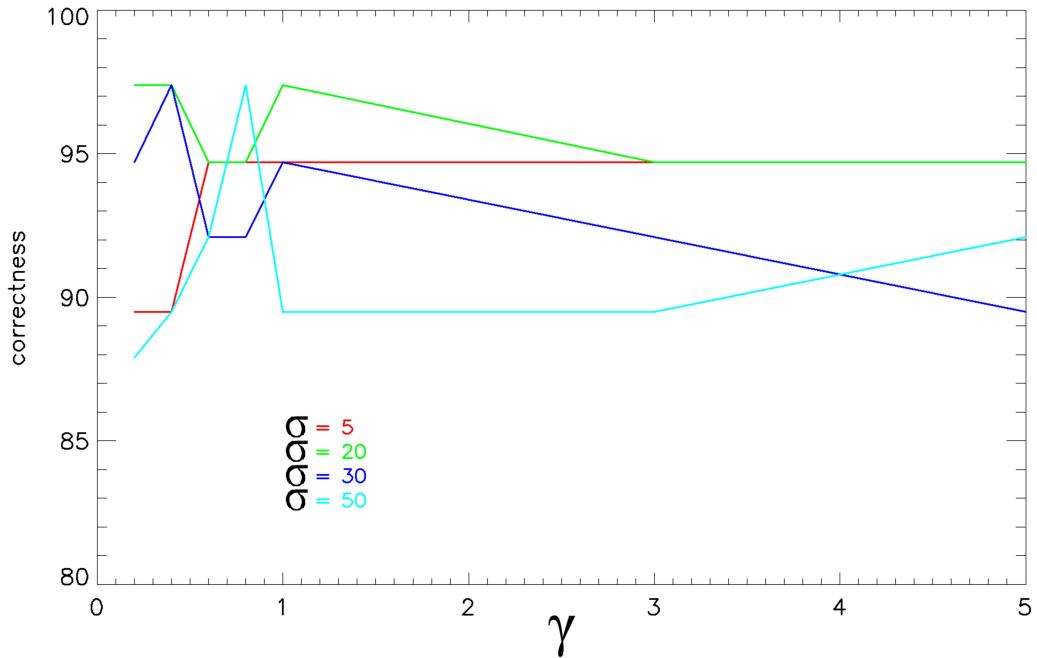


Abbildung 6.2: Vollständigkeit des Trackings in Abhängigkeit von  $\sigma$  und  $\gamma$

Um die Zahl der Fehlzuordnungen weiter zu reduzieren, werden die Objektpaare im Nachhinein nochmals auf ihre Korrelation geprüft. Liegt diese unter einem gegebenen Schwellwert, wird die Zuordnung verworfen. Im gezeigten Beispiel (Abbildung 6.3) wurden 36 der 38 möglichen Objektpaare richtig zugeordnet, was einer Vollständigkeit von 94,7 Prozent entspricht. Doch gleichzeitig wurden sechs Paarungen falsch zugeordnet (15,8%). Nach der Zuordnung durch Singulärwertzerlegung wurde auf die Paare der Korrelationsschwellwert von 0.9 angewandt, wobei zwar nur noch 27 von 36 richtigen Zuordnungen (75%) übrig blieben, aber auch alle sechs Fehlzuordnungen verworfen wurden. Die Zuverlässigkeit liegt in dem Fall bei 100 Prozent. Drei Viertel Vollständigkeit sind ausreichend, um einen repräsentativen Wert für die Durchschnittsgeschwindigkeit zu ermitteln, wie in Kapitel 6.2.2 zu sehen ist. Neben einem Korrelationsschwellwert ist es sinnvoll, einen Schwellwert für ermittelte Geschwindigkeiten bzw. Objektabstände festzulegen. Teilweise lag der Geschwindigkeitsfehler im Stadtgebiet bei 250 km/h, da eine einzige Fehlzuordnung über eine zu große Entfernung in sehr hohen Abweichungen resultieren kann. Bei einer Zeitdifferenz von 0,7 Sekunden und einem Öffnungswinkel von 800 Metern am Boden kann eine Fehlzuordnung theoretisch einen Fehler von mehr

## 6.2 Ergebnisse

als 4000 km/h ergeben. Selbst wenn sämtliche anderen Paare richtig zugeordnet würden, wäre die errechnete Durchschnittsgeschwindigkeit stark verfälscht. Daher wurden die Paarungen zusätzlich auf deren Distanz untereinander geprüft.

Bei weiteren Tests stellte sich heraus, dass das Tracking bessere Ergebnisse bringt, wenn die Fahrzeuge hinsichtlich ihrer Fahrtrichtung unterschieden, also auf beiden Straßenseiten separat getrackt werden. Zusätzlich ist der Parameter  $\sigma$  auf die zu erwartende Disposition anzupassen. In der Stadt sind beispielsweise geringere Geschwindigkeiten und somit kleinere Verschiebungen zu erwarten als auf Autobahnen. Dem Parameter  $\sigma$  werden somit je nach Straßenklasse verschiedene Werte zugewiesen, die zusätzlich durch die Zeit zwischen den Aufnahmen normiert werden. Eine größere Zeitspanne zwischen zwei Aufnahmen steht ebenfalls für größere Verschiebungen der zu trackenden Objekte.

### 6.2.2 Ergebnisse

Das Tracking wurde wie beschrieben auf mehrere Aufnahmen mit korrigierten Straßen angewandt. In den Aufnahmefolgen ON0015-ON0016, ON0024-ON0025 und ON100-ON0101 (91, 94 und 30 Fahrzeugpaare) wurden zunächst nur die Autobahnen berücksichtigt. Das Tracking wurde darauf mit verschiedenen Korrelationsschwellwerten getestet. In Abbildung 6.4 ist zu erkennen, dass in den Bildfolgen ON0024-ON0025 und ON0100-ON0101 sogar 100 Prozent Zuverlässigkeit (Correctness) erreicht werden können, was jedoch nicht bei ON0015-ON0016 der Fall ist. Bei ON0015-ON0025 nimmt die *FP*-Rate bei sinkender *TP*-Rate sogar zu. Der Effekt kommt daher, dass sich die *False Positives* in Relation zur Gesamtanzahl der gefundenen Tracks stehen. Wenn diese bei gleicher Anzahl von Fehlalarmen sinkt, steigt die relative *FP*-Rate. Alle drei Bildfolgen erreichen zwar Vollständigkeiten zwischen 70 und 80 Prozent, doch liegt an der Stelle die Rate der *False Positives* bei durchschnittlich einem Fünftel (12-28%). Für eine genaue Bestimmung der Durchschnittsgeschwindigkeit ist es jedoch wichtiger, möglichst wenige Fehlerkennungen mit einzubeziehen.

In der Stadt (Abb. 6.5) wurden die Aufnahmefolgen ON0100-ON0101 und ON0110-ON0111 (213 und 425 Fahrzeugpaare) getestet. Hier sind die Trackingraten wesentlich besser, da die meisten Objekte parkende Autos sind und sich deren Relativposition zueinander kaum ändert. Dadurch ist es für die Singulärwertzerlegung einfacher, die Objekte einander zuzuordnen. Die Werte erreichen selbst bei einer Vollständigkeit um die 70 Prozent die volle Zuverlässigkeit von 100 Prozent. Bei niedrigerem Korrelationsschwellwert erreicht das Tracking bis zu 90 Prozent Vollständigkeit bei durchschnittlichen *FP*-Raten von 14 Prozent. Die schlechteren Ergebnisse auf den Autobahnaufnahmen kommen daher, dass sich Fahrzeuge mit höhere Geschwindigkeit in gegensätzliche Richtungen bewegen, was ihre Relativpositionen teilweise stark verändert und falsche Zuordnungen zur Folge hat. Auch wenn beide Straßenseiten separat getrackt werden, werden nicht sämtliche Fahrzeuge einer Richtung zugleich getrackt. Das Problem dabei ist, dass positive sowie negative Abschnitte einer Straße nicht immer gleichgerichtet sind, d.h. nicht immer die gleiche Straßenseite repräsentieren. Einige Fahrzeuge bewegen sich also dennoch in entgegengesetzte Richtungen.

Noch wichtiger aber als die Anzahl der richtigen und falschen Zuordnungen ist die Ermittlung der Geschwindigkeit und deren Abweichung vom wirklichen Wert. Aus den manuell ermittelten Fahrzeugpositionen und -zuordnungen wurden die mittleren Geschwindigkeiten für jedes Set von Aufnahmen errechnet.

Allgemein ergeben sich die geringsten Geschwindigkeitsabweichungen (siehe Abb. 6.6 - 6.7) bei einem Korrelationsschwellwert von 0,85. Bei allen Bildfolgen liegt die Abweichung zwischen 0,4 und etwa 10 km/h. Bei den Stadtaufnahmen liegt die Geschwindigkeitsabweichung sehr niedrig, stellt sich aber erst ab 0,85 und darüber ein, während die Werte für die Autobahn oberhalb von 0,8 wieder stärker divergieren. Die zuverlässigsten Durchschnittsgeschwindigkeiten werden somit im Bereich von 0,8 bis 0,85 ermittelt.

## 6.3 Zusammenfassung

Es wurde in diesem Kapitel ein Fahrzeug-Tracking entwickelt, implementiert und für verschiedene Luftaufnahmen getestet. Dabei wurde gezeigt, dass die Assoziation von Fahrzeugen in Folgebildern durch Singulärwertzerlegung ihrer Abstände und Ähnlichkeit möglich ist. Dabei wurden Vollständigkeiten von bis zu 90 Prozent erzielt. Die ermittelten Geschwindigkeiten wichen in allen Datensätzen bei optimalen Parametereinstellungen um höchstens zehn Stundenkilometer vom manuell ermittelten Wert ab, in mehreren Fällen war die Abweichung sogar weniger als ein km/h.

Um die Zuverlässigkeit noch weiter zu verbessern, ist es wichtig, Fehlzuordnungen zu erkennen und zu eliminieren. Einen guten Ansatz dafür liefert [23], indem die ermittelten Geschwindigkeitswerte mit der Glaubwürdigkeit ihres Auftretens gewichtet werden (siehe Abbildung 6.8 links). Für bestimmte Verkehrsdichten oder Straßenbereiche, beispielsweise vor Kreuzungen, sind hohe Geschwindigkeiten unwahrscheinlicher und gehen mit geringerem Gewicht in die Durchschnittsgeschwindigkeit ein. Zusätzlich wird die Standardabweichung errechnet und Fahrzeuge, welche außerhalb eines gewissen Bereiches der Verteilungsdichtefunktion (Abb. 6.8 rechts) liegen, als Ausreißer deklariert und verworfen.

Mit dieser Bewertungsmethode ist es den Autoren gelungen, die Zuverlässigkeit des Trackings von 65 auf 95 Prozent zu erhöhen.

## 6 Tracking

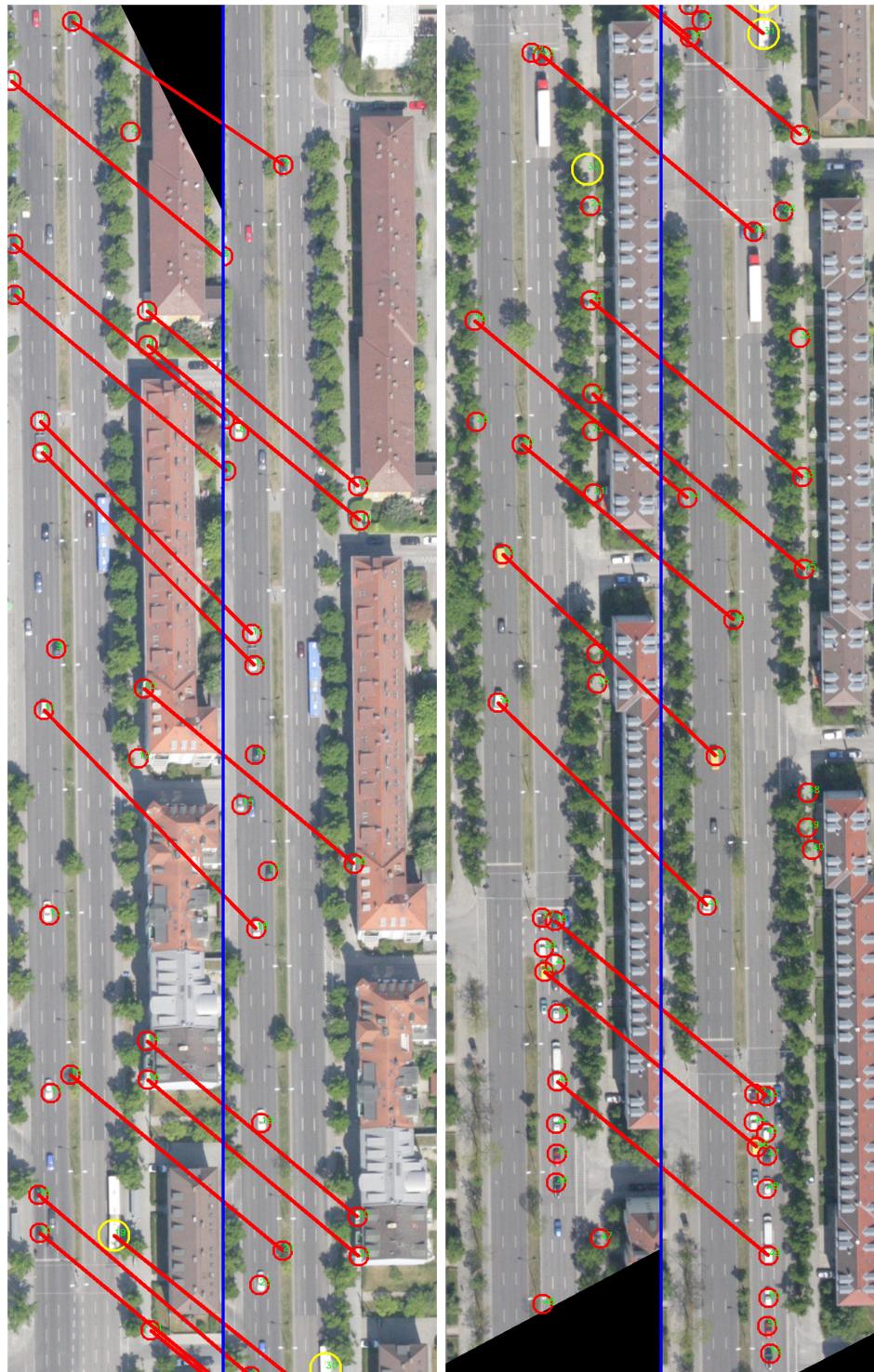


Abbildung 6.3: Zuordnungen nach Singulärwertzerlegung ( $\sigma=30$ ,  $\gamma=1,0$ ) und Korrelationsschwellwert (CC=0,9), rot - PKW, gelb - LKW

### 6.3 Zusammenfassung

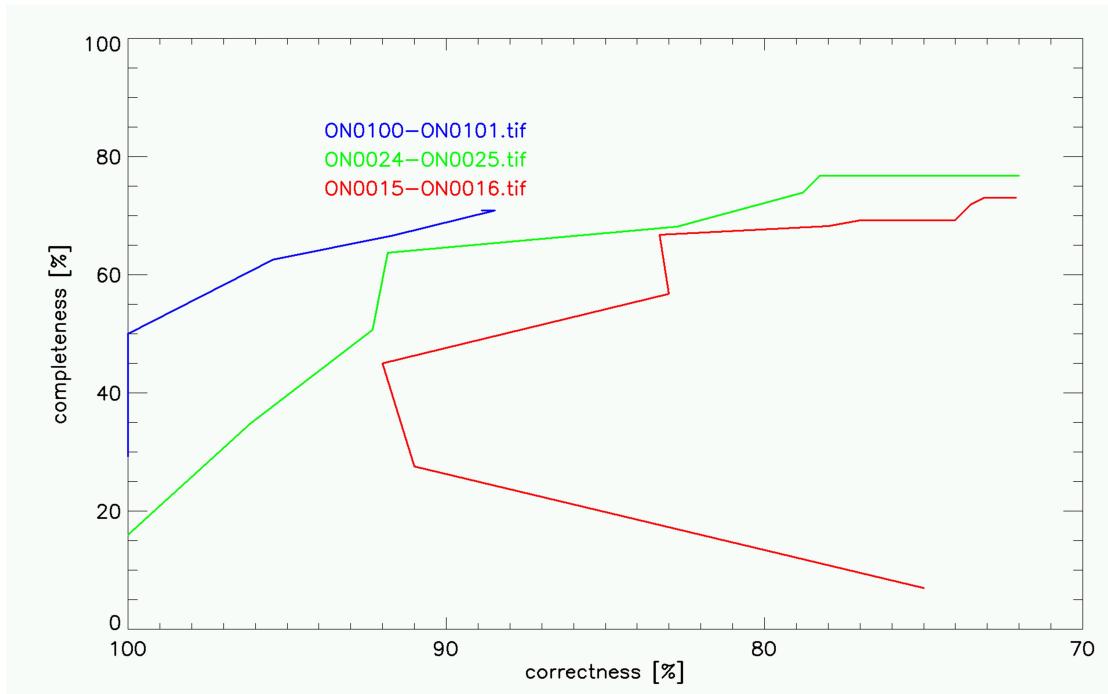


Abbildung 6.4: Trackingraten bei verschiedenen Korrelationsschwellwerten (Autobahn)

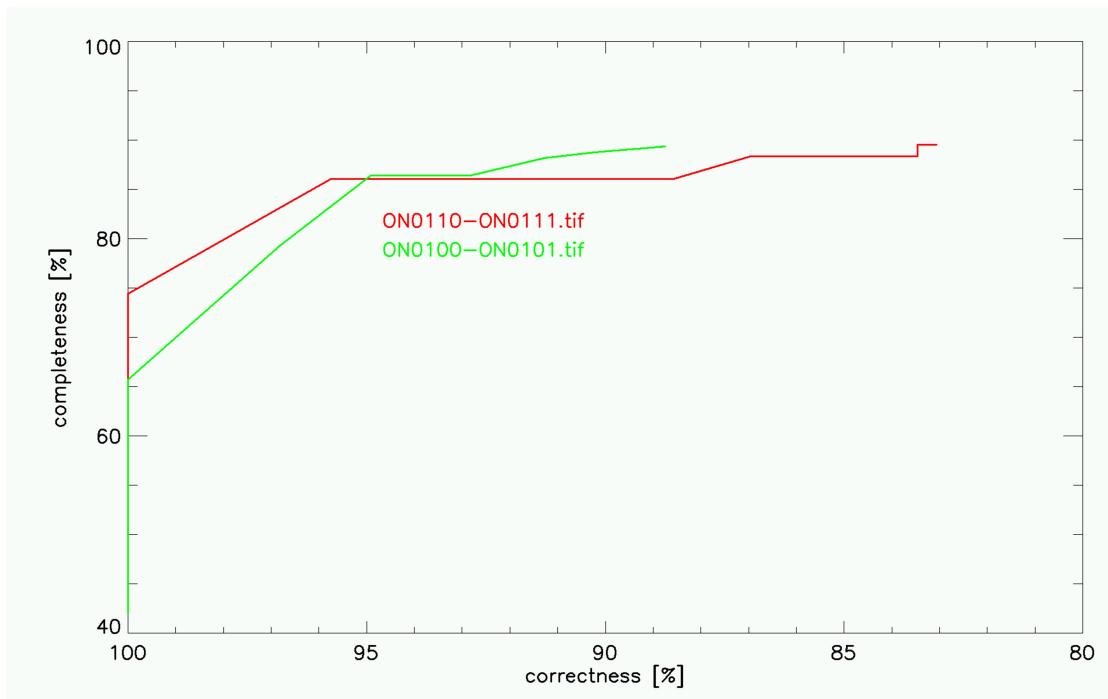


Abbildung 6.5: Trackingraten bei verschiedenen Korrelationsschwellwerten (Stadt)

## 6 Tracking

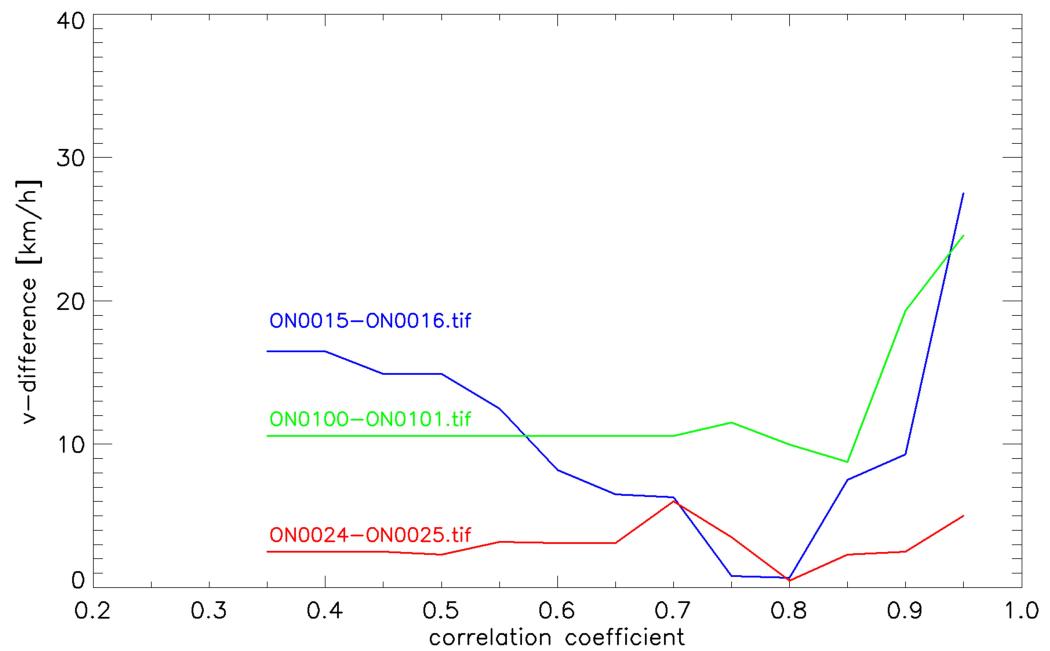


Abbildung 6.6: Geschwindigkeitsabweichung in Abhangigkeit des Korrelationsschwellwertes (Autobahn)

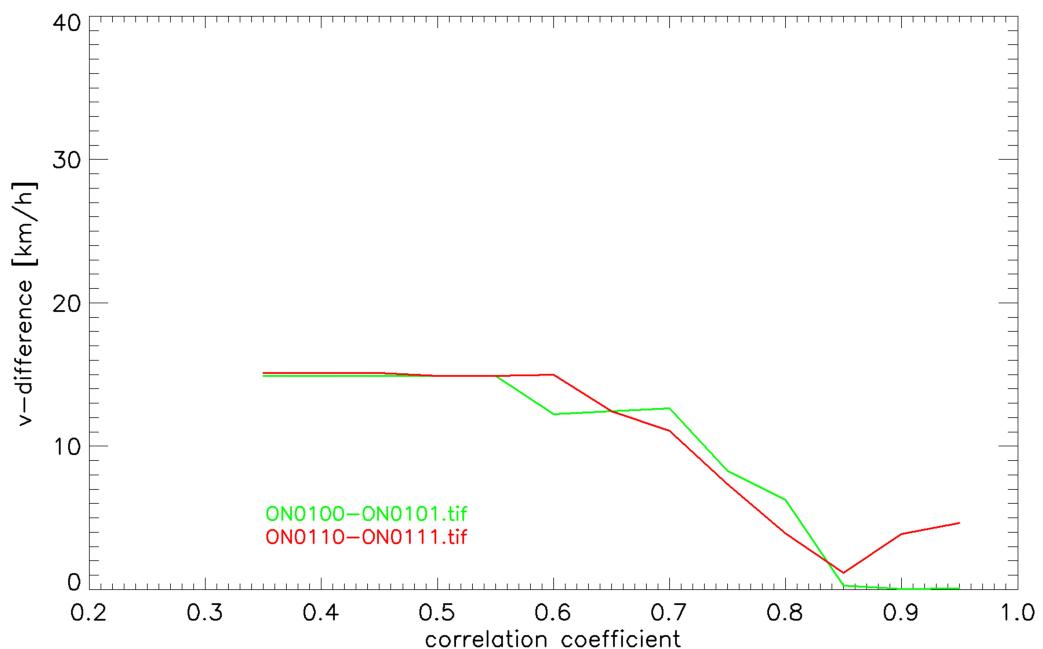


Abbildung 6.7: Geschwindigkeitsabweichung in Abhangigkeit des Korrelationsschwellwertes (Stadt)

## 6 Tracking

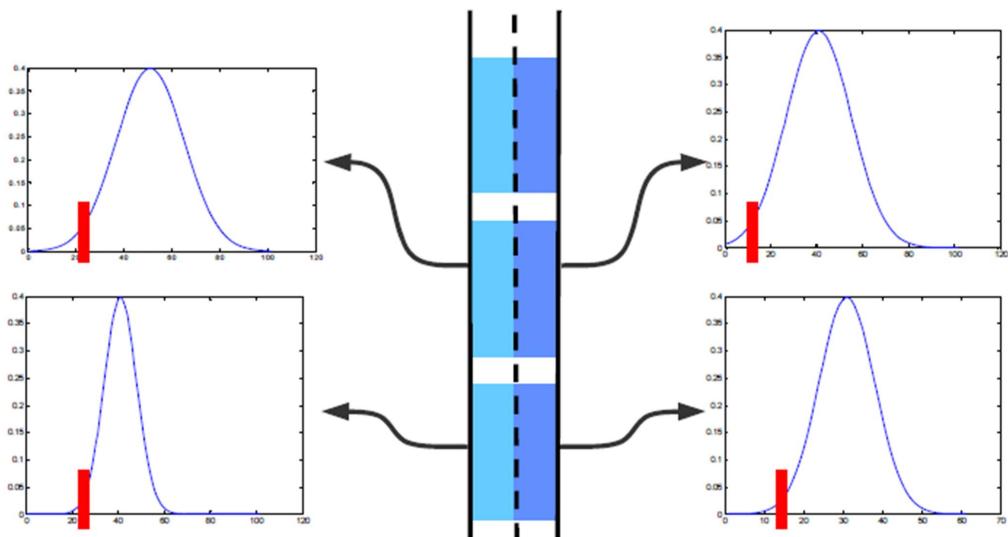
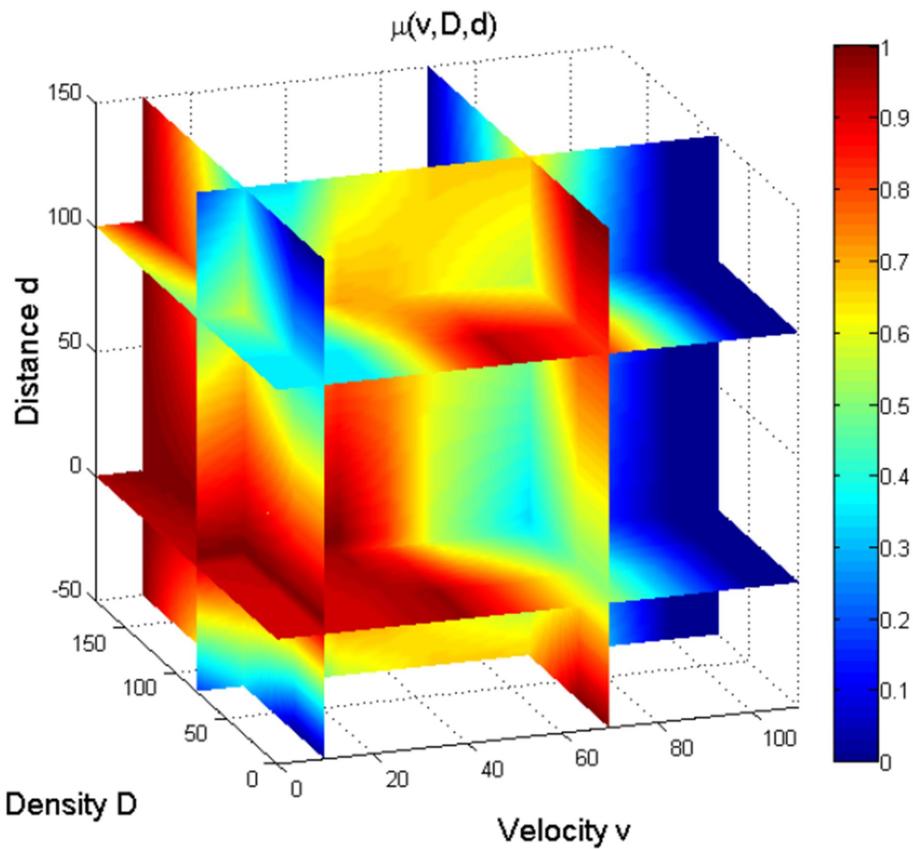


Abbildung 6.8: Oben - Wahrscheinlichkeiten der Geschwindigkeitswerte, Unten - Verteilungsdichtefunktion mit Grenzwert (rot), Quelle: [23]

## 7 Optische Orientierung

In diesem Kapitel wird die Entwicklung einer optischen Orientierungsbestimmung erläutert, welche das in Kapitel 2 erwähnte Inertialsystem (IMU) ersetzen soll. Da die GPS/IMU-Einheit eine der kostspieligsten Komponenten des *ARGOS* darstellt, ist die Entwicklung einer alternativen Fluglagebestimmung erstrebenswert. Praktisch ist es, dabei die Sensoren zu verwenden, die im *ARGOS* ohnehin integriert sind, zum Beispiel die Kameras. Allgemein ist die Methode, die Eigenorientierung durch das Auswerten optischer Merkmale zu bestimmen, schon lange bekannt und auf verschiedenste Arten realisiert.

Ein klassischer Ansatz ist, die äußere Orientierung einer Kamera durch räumlichen Rückwärtsschnitt zu bestimmen (siehe dazu [70] S. 239 ff.). Dabei müssen die Weltkoordinaten  $X$ ,  $Y$  und  $Z$  von mindestens drei Objektpunkten und ihre zugehörigen Abbildungskoordinaten  $x$  und  $y$  im Bildraum bekannt sein sowie die innere Orientierung der Kamera. Die Koordinaten werden dabei in die Kollinearitätsgleichungen 7.1 und 7.2 eingesetzt, welche die Abbildung in den Bildraum beschreiben.

$$x' = x'_0 + z' \frac{r_{11}(X - X_0) + r_{12}(Y - Y_0) + r_{13}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)} + \Delta x' \quad (7.1)$$

$$y' = y'_0 + z' \frac{r_{21}(X - X_0) + r_{22}(Y - Y_0) + r_{23}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)} + \Delta y' \quad (7.2)$$

Die Parameter der inneren Orientierung sind dabei der Bildhauptpunkt  $(x'_0, y'_0)$ , die Brennweite ( $c = -z'$ ) und die Verzeichnungsoffsets ( $\Delta x'$ ,  $\Delta y'$ ) und werden als bekannt vorausgesetzt. Durch Ausgleichung nach vermittelnden Beobachtungen werden die unbekannten Parameter  $X_0$ ,  $Y_0$ ,  $Z_0$  (Weltkoordinaten des Projektionszentrums) und  $r_{11}$  bis  $r_{33}$  (Elemente der Rotationsmatrix) bestimmt. Das Problem bei dieser Methode liegt jedoch darin, dass die Abbildungen der Punkte im Bild ermittelt werden müssen. Auch wenn eine ganze Reihe an Software für die Orientierungsbestimmung existiert, müssen die Passpunkte im Bildraum nach wie vor manuell bestimmt werden.

Die NASA benutzt optische Navigation in der Raumfahrt [14], um ähnlich wie beim GPS die eigene Position aus der Triangulation von Sensor und bekannten Himmelskörpern zu schätzen.

In [29] und [31] kompensieren die Autoren ungenaue Fluglagedaten dadurch, dass sie die Kanten eines dreidimensionalen Umgebungsmodells mit den sichtbaren Gebäudekanten im Bild matchen. Damit lässt sich die Fluglage zwar automatisiert und auch genauer bestimmen, doch ist dies nur möglich, wenn ein virtuelles Modell der Umgebung existiert, was nur an sehr wenigen ausgewählten Orten der Fall ist.

In [60] nutzen die Autoren die *Monte Carlo localization* für die Selbstlokalisierung eines Roboters. Dabei werden Wahrscheinlichkeiten für eine Reihe möglicher Positionen und Orientierungen des Roboters bestimmt. Mittels eines Bewegungsmodells und der Extraktion bekannter optischer Umgebungsmerkmale werden diese Wahrscheinlichkeiten permanent aktualisiert. Aus einer Reihe von Stichproben kann zu jeder Zeit die glaubwürdigste Orientierung ermittelt werden. Die optischen Merkmale werden jedoch ähnlich wie in [29] mit bekannten Umgebungsmodellen verglichen.

In den Quellen [56], [95] und [102] errechnen die Autoren die relativen Positionen aufeinanderfolgender Bilder mithilfe des Optischen Flusses. Dabei werden die Passpunkte vom Computer selbst festgelegt und durch *Lucas-Kanade Feature Tracking* in Folgebildern wieder gefunden. Die gewonnenen Bewegungsinformationen fließen in ein Kalman-Filter ein, um die Navigationsdaten des GPS/IMU-Systems zu stützen.

Durch die Assoziation der Punktpaare in Folgebildern kann eine relative Orientierung (siehe [70] S. 255 ff.) der Aufnahmen zueinander errechnet werden. Diese relative Orientierung enthält drei Rotationswinkel und drei Modellkoordinaten der Translation, welche nach Belieben skaliert werden können.

In [38] verwenden die Autoren eine Stereokamera zur Stützung der IMU und können zu jedem Zeitpunkt die dreidimensionalen Positionen von detektierten *Interest Points* bestimmen. Dadurch lässt sich zwischen zwei Aufnahmen sogar die absolute Orientierung errechnen. Um Echtzeitfähigkeit zu erreichen, wird der Suchraum in Folgebildern minimiert. Dafür stützen sich IMU und Bildverarbeitung gegenseitig. Aus den Drehraten des Inertialsystems wird zunächst der Bewegungsvektor im Bild eingegrenzt. Die optischen Informationen wiederum verringern ein Abdriften der IMU.

Das Problem bei diesen Ansätzen ist jedoch einerseits, dass zu Anfang eine absolute Orientierung bekannt sein muss. Durch das Verfolgen von Passpunkten können allenfalls nachfolgende Orientierungen ermittelt werden, doch bedarf es nach wie vor bekannter Landmarken, um die Fluglage initial zu orientieren. Andererseits werden die optischen Informationen nur verwendet, um die inertialen Daten zu stützen. Der hier entwickelte neuartige Ansatz kann die IMU bis zu einer gewissen Genauigkeit sogar ersetzen.

In [82] wird ein Algorithmus präsentiert, welcher in Satellitenbildern Straßen extrahiert und die Stützpunkte von Straßennetzen in diese matcht. Durch ein Gradientenverfahren werden die Parameter der affinen Abbildung ermittelt, durch welche die Straßennetze optimal übereinander passen. Der für diese Arbeit entwickelte Ansatz verfolgt grundsätzlich die gleiche Idee, unterscheidet sich jedoch in einigen Details wie etwa der Straßenextraktion oder dem gewählten Optimierungsverfahren. Darüberhinaus wird der hier vorgestellte Ansatz um die zusätzliche Komponente der relativen Folgeorientierung und deren Fusion mit den zuvor erhobenen Daten erweitert.

### 7.1 Absolute optische Orientierung

Die Idee, ein Umgebungsmodell zu benutzen, dessen Abbildung im Bildraum erkannt und gematcht wird, soll auch hier aufgegriffen werden. Statt jedoch wie in [29] sporadisch vorhandene Gebäudemodelle zu verwenden, soll das Straßennetz verwendet werden, da

die *ARGOS*-Software, wie schon in Kapitel 5 erwähnt, eine *NAVTEQ*-Straßendatenbank enthält. Straßen sind an allen Stellen vorhanden, an denen Verkehrsinformationen erhoben werden sollen. Somit kann man davon ausgehen, dass in den meisten Aufnahmen genügend Straßen enthalten sind, um diese mit den Straßenkanten der Datenbank zu matchen.

### 7.1.1 Extraktion von Straßenbereichen

Zur Extraktion von Straßen wurden in den letzten Jahren viele Ansätze veröffentlicht. Sehr ausführlich befasst sich die Dissertation [44] mit diesem Thema. In mehreren Phasen werden die Aufnahmen zunächst segmentiert und in reduzierter Auflösung homogene Bänder als Straßenhypthesen extrahiert. Am Ende werden die glaubwürdigsten Hypothesen im globalen Kontext zu Straßennetzen verknüpft.

In [20] wird ein *region growing*-Algorithmus auf LIDAR-Aufnahmen angewandt, um von bestimmten Startpunkten aus Gebiete mit weitgehend homogener Höhe und bestimmten reflektiven Eigenschaften zu extrahieren. Letztendlich werden diese vektorisiert und zu einem Straßennetz zusammengefasst.

Andere Ansätze, wie in [32], verwenden speziell adaptierte Kantenfilter, um in Luftbildern geringer Auflösung sämtliche Pixel zu bewerten. Anschließend werden diese zu einem Pfad verbunden, dessen Kostenfunktion minimiert wird.

In [46] wird die Straßenoberfläche im HSV-Farbraum [9] auf Basis ihres Farbwertes oder ihrer Sättigung und Intensität extrahiert. Hierbei wird unterschieden, ob die Intensitätsunterschiede im Bild groß genug sind. Ist das der Fall, ist der Farbwert (hue) der Hauptindikator für die Extraktion. Bei geringen Intensitätsunterschieden werden Sättigung und Intensität zur Extraktion herangezogen. Auch wenn die Aufnahmen in diesem speziellen Fall aus einem Fahrzeug heraus gemacht wurden, lässt sich der HSV-Ansatz in gleicher Weise auf Luftbilder anwenden.

Die *ARGOS*-Aufnahmen werden ebenfalls in den HSV-Farbraum umgerechnet. Der Farbwinkel der Straßenoberfläche liegt meist in einem Bereich zwischen 150 und 190 Grad, kann jedoch abhängig von Wetterlage und Weißabgleich variieren und zusätzliche Maxima in anderen Bereichen aufweisen. Daher ist es sinnvoll, zusätzlich die Sättigung mit einem Schwellwert zu begrenzen, um die Pixelmenge weiter einzuschränken. Dadurch lässt sich zwar die Straßenoberfläche nicht eindeutig bestimmen, doch werden insgesamt signifikant mehr Pixel auf Straßen extrahiert als in der Umgebung (siehe Abbildungen 7.1 - 7.3).

### 7.1.2 Straßenmatching

Da die Zusammenhänge und Umrechnungen zwischen Flugzeug, Kamera und Weltkoordinatensystemen bereits ausführlich in Kapitel 3 erläutert wurden, soll hier nicht weiter darauf eingegangen werden.

Die zweidimensionalen Straßenknoten sind als UTM-Koordinaten [2] gespeichert und werden mit dem zugehörigen Höhenwert versehen (Abb. 7.4), welcher aus dem SRTM-Geländemodell [7] interpoliert wird. Die entstehenden 3D-Koordinaten werden in die

## 7 Optische Orientierung



Abbildung 7.1: Autobahn A96 in München

Aufnahme projiziert [58]. Die Position des Flugzeuges bzw. der Kamera liefert der GPS-Sensor. Die Startwerte für den Nick- ( $\omega$ ) und Rollwinkel ( $\phi$ ) werden zunächst als 90 und 0 Grad angenommen, d.h. die Kamera blickt senkrecht nach unten. Der Gierwinkel ( $\kappa$ ) wird aus zwei aufeinanderfolgenden Flugpositionen errechnet wird. Mit der gegebenen Anfangsorientierung der Kamera können die Straßen ins Bild projiziert und je nach innerer Orientierung verzerrt werden. Als Gütemaß für die Projektion werden sämtliche Pixel zusammengerechnet, die sowohl zur extrahierten als auch zur projizierten Straßenfläche gehören.

Da die Kostenfunktion nur ein Aufsummieren von Pixeln ist, können aus ihr keine Gradienten abgeleitet werden. Daher ist eine klassische Ausgleichung an der Stelle nicht angebracht. Stattdessen wird ein von Nelder und Mead entwickelter Simplex-Algorithmus angewandt [79], [87]. Da nur die drei Rotationswinkel ausgeglichen werden müssen, wird im dreidimensionalen Zustandsraum ein Simplex aus vier Eckpunkten um die Startwerte aufgebaut. Die Eckpunkte werden zu jedem Zeitpunkt bewertet, wodurch die zwischen ihnen entstehenden Gefälle dem Algorithmus anstelle von Gradienten als Anstieg dienen. Der Algorithmus verschiebt, spiegelt, expandiert und kontrahiert das Simplex solange, bis sich dessen Ecken bei erfolgreicher Konvergenz um ein Maximum zusammenziehen.

In Abbildung 7.6 ist die Güte der Projektion in Abhängigkeit zweier Rotationswinkel dargestellt. Die Grauwertsprünge im Schwarz-Weiß-Bild kommen durch den mehrfachen Überlauf des 8-Bit-Grauwertes zustande, stellen also eigentlich einen steigen Verlauf dar.

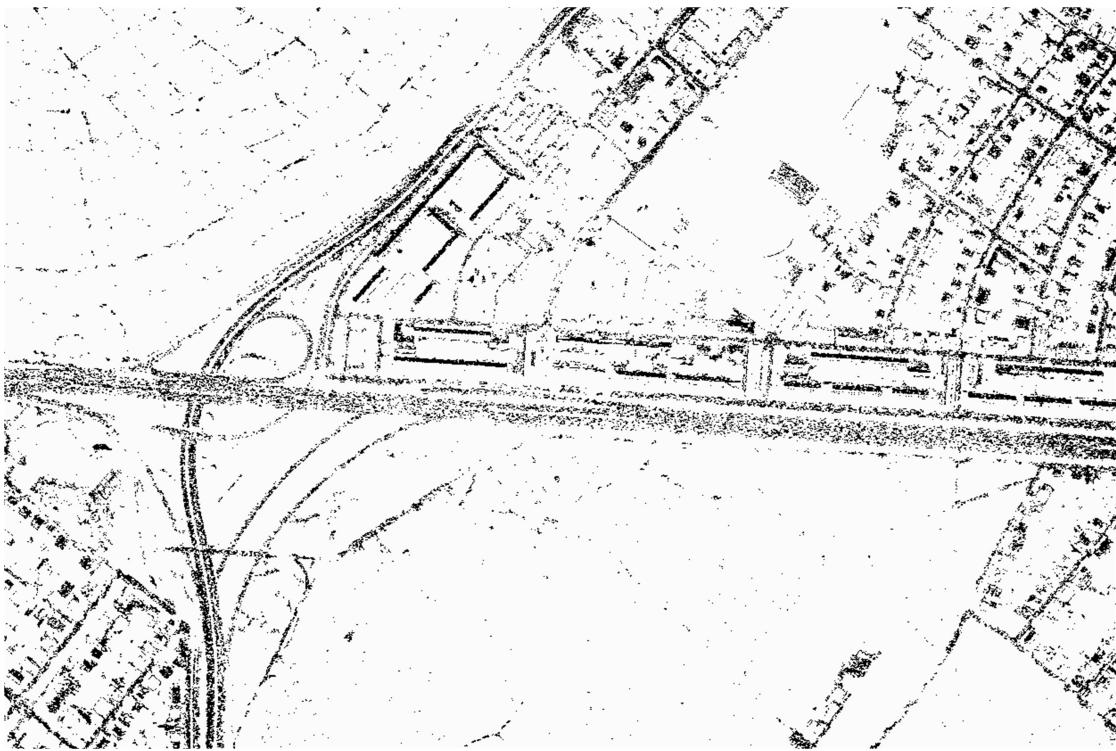


Abbildung 7.2: Mittels Farbwinkel extrahierte Straßen

Auch wenn sich das globale Maximum eindeutig abzeichnet, welches die optimalen Rotationswinkel repräsentiert, ist die Oberfläche der Kostenfunktion stark zerfurcht und weist viele lokale Extrema auf, in welchen die Ausgleichung oft falsch konvergiert. Daher muss die Bewertung geglättet werden. Ein passender Ansatz wird in einem Kantenmatching-verfahren (*Chamfer Matching*) angewandt [16], wobei sämtliche Kanten verbreitert werden, jedoch an ihren Rändern geringere Werte erhalten. Dadurch wird der Einzugsraum der Straßen vergrößert. Die breiteren Straßen überlappen leichter mit den Extrahierten Pixeln ihrer Umgebung, was eine bessere Konvergenz gewährleistet. Die Abbildungen 7.7 und 7.8 zeigen die verbreiterten Straßen und ihre Auswirkung auf den Bewertungsraum. Die Oberfläche ist sichtlich glatter.

Problematisch ist jedoch, dass die Projektion und Verbreiterung der Straßen bei jeder Iteration erneut ausgeführt werden müssen, was je nach Anzahl der Straßenelemente sehr viel Rechenzeit in Anspruch nehmen kann. Daher arbeitet der Algorithmus hierarchisch. In der geringsten Auflösung werden zunächst empirisch die Startwerte bestimmt, indem in einem bestimmten Bereich um das letzte Optimum der Parameterraum in Ein-Grad-Schritten durchsucht wird.

Um die Startwerte mit der besten Bewertung wird durch geringes Verändern einzelner Parameter das Simplex definiert und in der nächsthöheren Auflösung der Algorithmus gestartet. Wenn das Konvergenzkriterium erreicht ist, werden die Eckpunkte in die nächsthöhere Auflösung übernommen, und die Optimierung wird erneut gestartet.

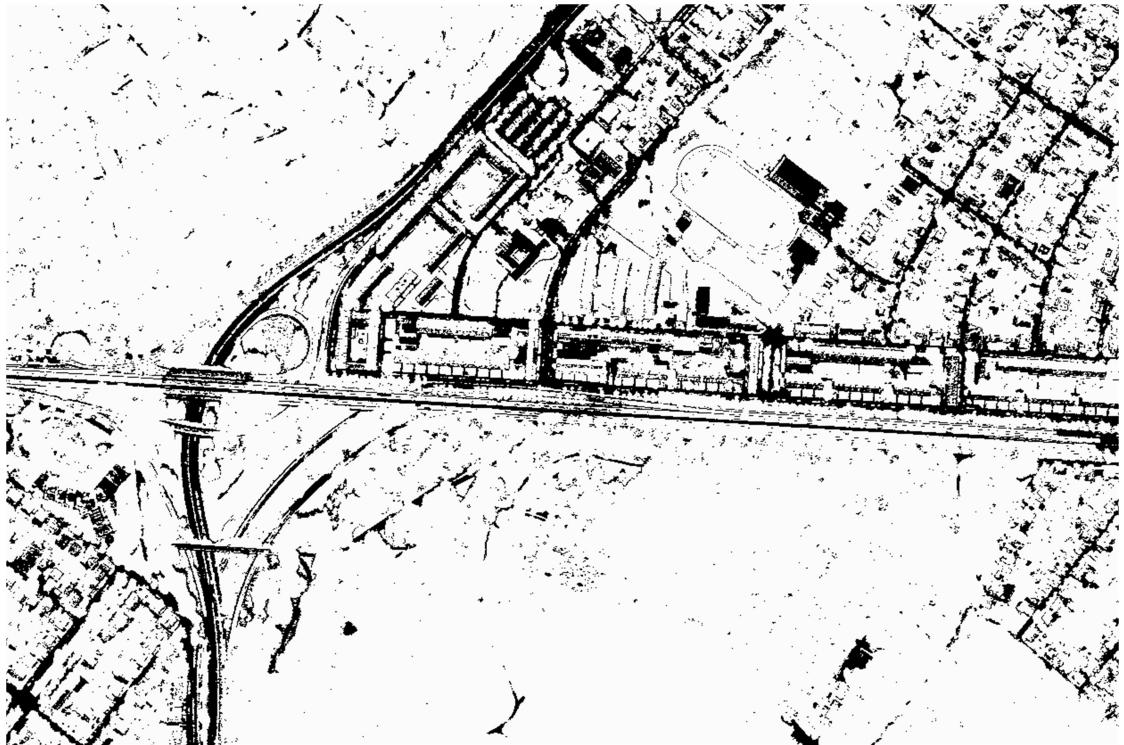


Abbildung 7.3: Mittels Sättigung extrahierte Straßen

### 7.1.3 Ergebnisse

Der Algorithmus wurde auf über 400 Aufnahmen in elf Überflügen im Münchener Stadtgebiet getestet. Die Aufnahmen stammen aus zwei verschiedenen Messkampagnen der Jahre 2007 und 2008. Abbildung 7.9 zeigt zwei dieser Aufnahmen nach der Konvergenz des *Nelder-Mead*-Algorithmus mit optimierten Rotationswinkeln. Verglichen wurden die Ergebnisse mit den Rotationswinkeln, die das Inertialsystem lieferte und welche in photogrammetrische Winkel überführt wurden. Die optisch ermittelten Winkel konvergierten in 83,5 Prozent der Fälle richtig (Abb. 7.10) und erreichten im Falle korrekter Konvergenz Genauigkeiten zwischen 0,1 und 0,2 Grad.

Zwei Datensätze (Überflug A und B) sollen genauer untersucht und auf diese auch spätere Algorithmen angewandt werden. Ergebnisse weiterer Datensätze befinden sich im Anhang ab Seite 99. Die Abbildungen 7.11 und 7.12 zeigen die ermittelten  $\omega$ -Winkel beider Überflüge im Vergleich zu den Werten, die das Inertialsystem lieferte. Wie man erkennen kann, treten bedingt durch falsche Konvergenz gelegentlich Ausreißer auf, wobei die sonstige Genauigkeit bei 0,05 bis 0,2 Grad liegt. Die Ausreißer verzerrten die RMS-Werte jedoch auf durchschnittlich 1,2 Grad, wie in Abb. 7.13 und 7.14 zu sehen ist.

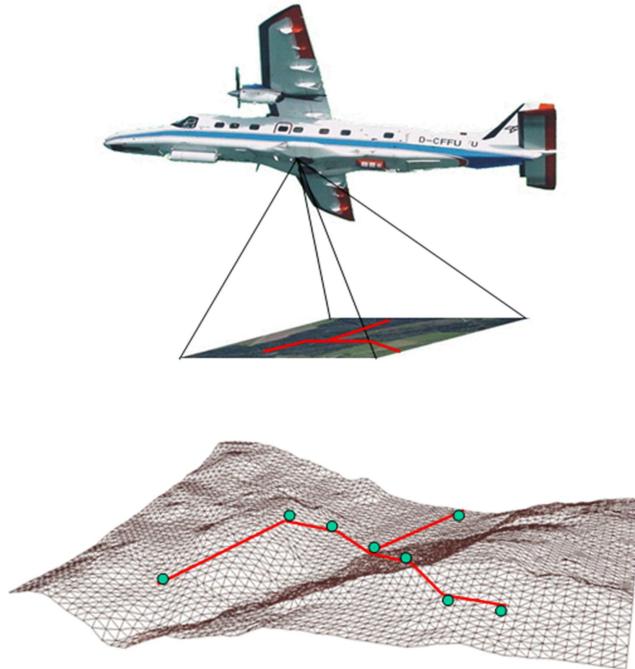


Abbildung 7.4: Projektion der dreidimensionalen Straßenelemente in die Aufnahme

#### 7.1.4 Fehlerquellen

Manche Datensätze zeigen jedoch im Gegensatz zum gezeigten Beispiel wesentlich seltener Konvergenz. Dies kann verschiedene Ursachen haben.

- Zu wenig Straßeninformation

Zu wenig Straßeninformation kann zu falscher Konvergenz führen, wenn das Bild sehr wenig Straßenoberfläche zeigt. Einige Pixel werden jedoch fast immer extrahiert. Navteq-Straßenelemente, welche außerhalb der Aufnahme liegen (Abbildung 7.15), verursachen jedoch eine höhere Bewertung, wenn sie im Bild auf zufällig extrahierte Pixel treffen.

- Zu viel Straßeninformation

Ein weiterer Grund für falsche Konvergenz kann auch zu viel Straßeninformation sein. In Abbildung 7.16 sind sehr viele kleine Straßen sichtbar, was zu vielen lokalen Extrema im Bewertungsraum führt. Das Simplex konvergiert an diesen.

- Schlechte Straßenextraktion

Ein weiterer oft auftretender Fehler wird durch schlechte Straßenextraktion hervorgerufen. Wenn sich die Wettersituation oder der Weißabgleich der Kamera ändern,

## 7 Optische Orientierung



Abbildung 7.5: Extrahierte (schwarz) und projizierte (rot) Straßen überlagert

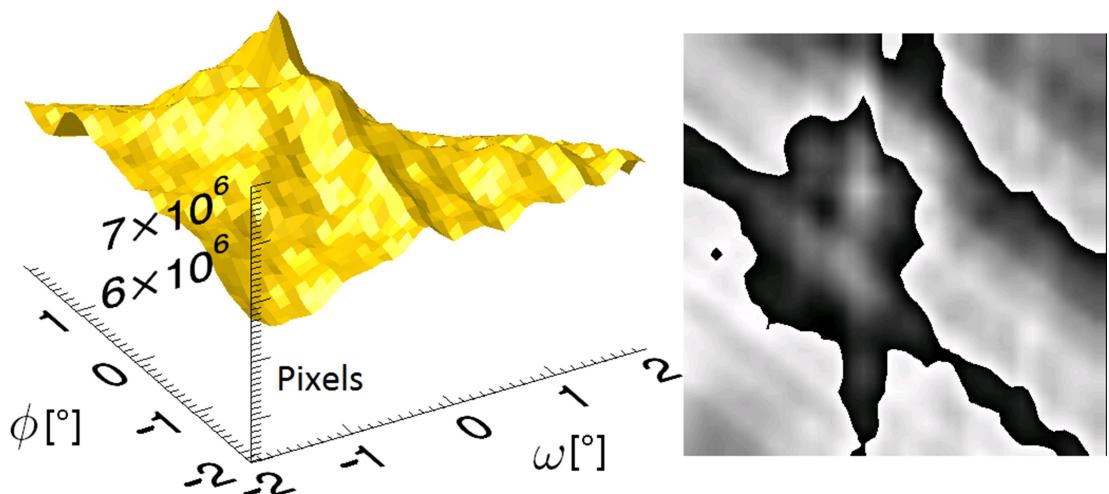


Abbildung 7.6: Projektionsgüte in Abhängigkeit der Winkel  $\omega$  und  $\phi$

verschiebt sich der Farbwinkel der Straßenoberfläche, was zum einen zur Folge haben kann, dass sie nicht mehr als solche erkannt wird, zum anderen, dass die

## 7.1 Absolute optische Orientierung



Abbildung 7.7: Extrahierte (schwarz) und verbreiterte projizierte (rot) Straßen überlagert

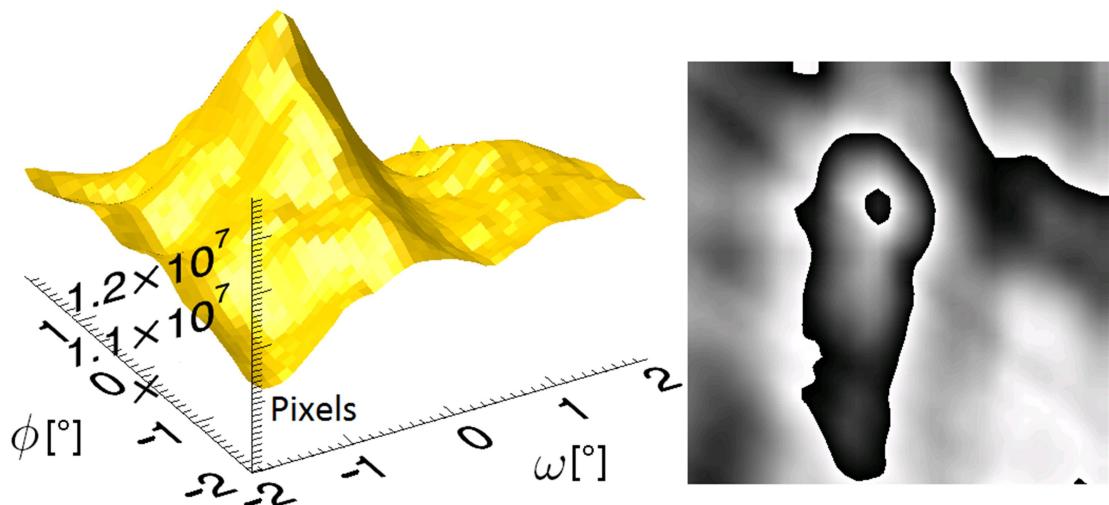


Abbildung 7.8: Projektionsgüte in Abhängigkeit der Winkel  $\omega$  und  $\phi$  (verbreiterte Straßen)



Abbildung 7.9: Richtig konvergierte Aufnahmen

Extraktion stärker auf andere Flächen wie Baustellen, Brachflächen oder Bahnanreal (Abbildung 7.17) reagiert.

Um diese Fehlerquellen zu umgehen, bieten sich verschiedene Ansätze an. Beispielsweise könnte eine exaktere Straßenextraktion die erste und dritte Fehlerquelle abdecken. Wenn die Straßen nicht mehr nur unstrukturiert extrahiert werden, könnte das der falschen Konvergenz bei wenig Straßeninformation vorbeugen, da weniger Straßenelemente auf zufällig auftretende Pixel gematcht werden würden. Die falsch extrahierten Oberflächen in Fehlerbeispiel drei würden das Matching auch weniger beeinflussen. Die zweite Fehlerquelle könnte durch ein Multistart-Simplex-Verfahren vermieden werden. Mehrere Startwerte werden zufällig in einem gegebenen Bereich gesucht und das Verfahren auf diese angewandt. Zwar dauert die Konvergenz mehrerer Simplizia entsprechend länger, doch gewährt das Verfahren, wie in [47] beschrieben, weitaus bessere Konvergenz zum globalen Optimum. Die Berechnung kann auch parallelisiert werden.

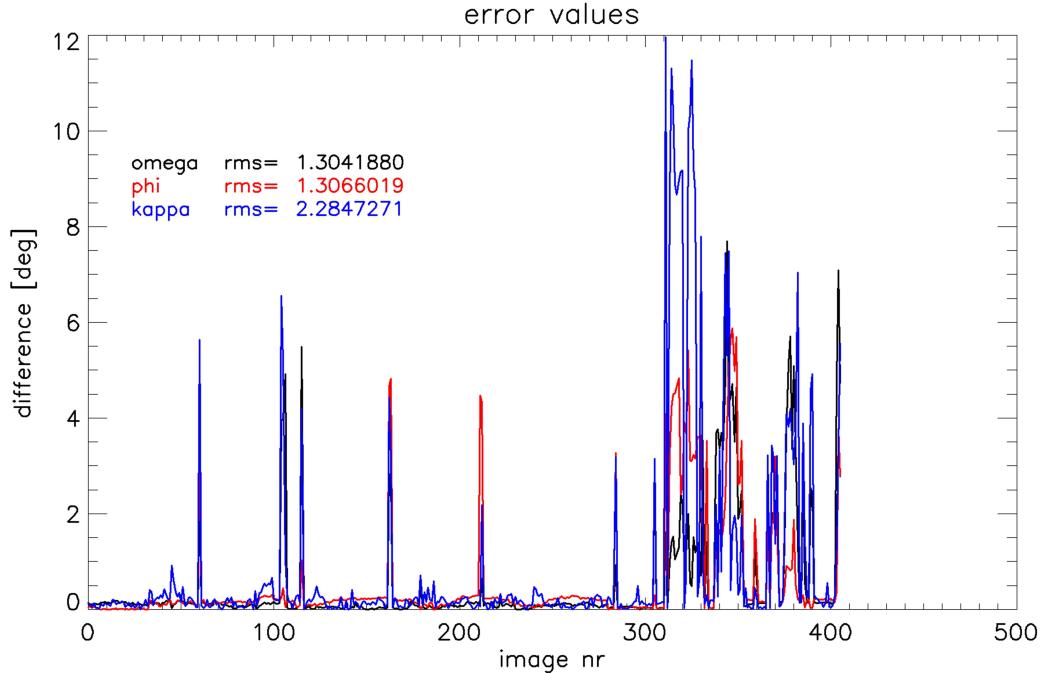


Abbildung 7.10: Winkelfehler aller 400 Aufnahmen

## 7.2 Relative optische Orientierung

Ein weiterer Ansatz für die Fehlervermeidung in 7.1.4 wäre, die Ausreißer einfach zu verwerfen. Dazu müssen diese aber erst als Ausreißer erkannt werden, was sich durch den Vergleich aufeinanderfolgender Orientierungen realisieren lässt. Die oben erwähnten Ansätze von [31], [56] und [95] bedienen sich dabei der relativen Orientierung zwischen benachbarten Bildern.

### 7.2.1 Interest Points

Da die Berechnung des Optischen Flusses für die Assoziation von Punkten in den großen Aufnahmen sehr zeitaufwändig ist, wird hier auf die Suche und das Matching bestimmter *Interest Points* oder Ground Control Points zurückgegriffen. Diese müssen, um sie in Folgebildern richtig zuzuordnen, bestimmte Bedingungen erfüllen (siehe[57]), im Allgemeinen sich von ihrer Umgebung abheben. Einfache Ansätze für deren Bestimmung, wie etwa *Harris Corner Detector*, Förstner- oder SUSAN-Operator, sind in [57] beschrieben. In Verbindung mit einem Punkt-Matching-Algorithmus wie etwa *sum of squared differences (SSD)* oder einfache Kreuzkorrelation *cross correlation (CC)* wären diese Passpunkte wieder auffindbar, doch stellten sich in den letzten Jahren, wie in Abschnitt 5.1.2 erwähnt, Ansätze wie *SIFT* [68], [72] oder *SURF* [13] als robuster heraus. Wie in [99] untersucht wurde, sind Varianten des *SURF*-Algorithmus nicht nur schneller, sondern auch zuverlässiger als der *SIFT*-Algorithmus.

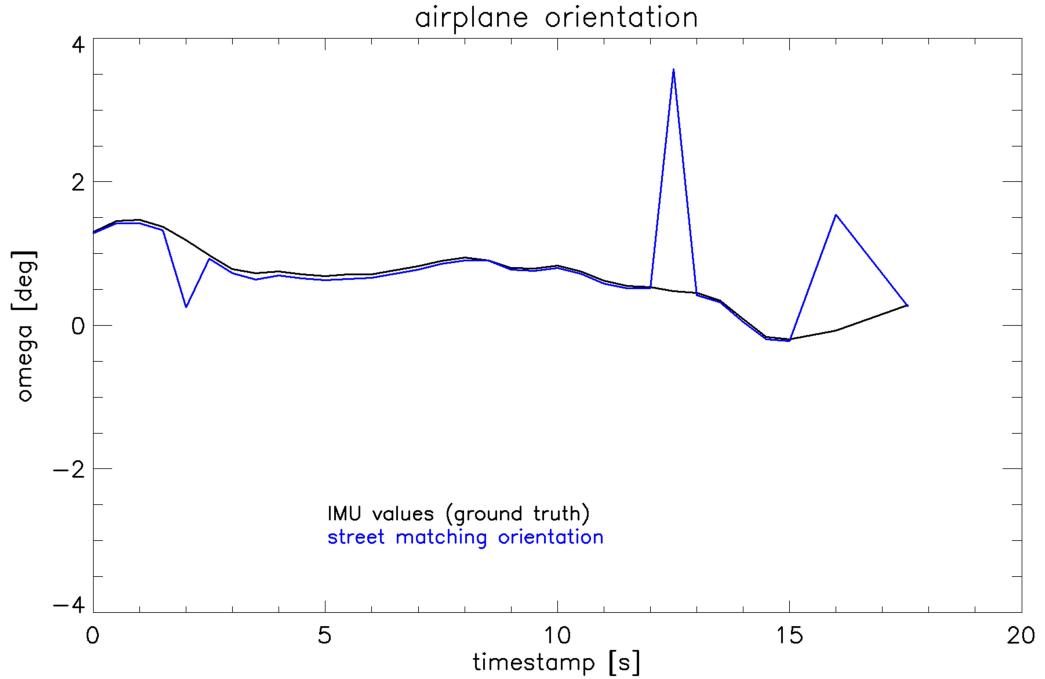


Abbildung 7.11:  $\omega$ -Winkel der absoluten optischen Orientierung (Überflug A)

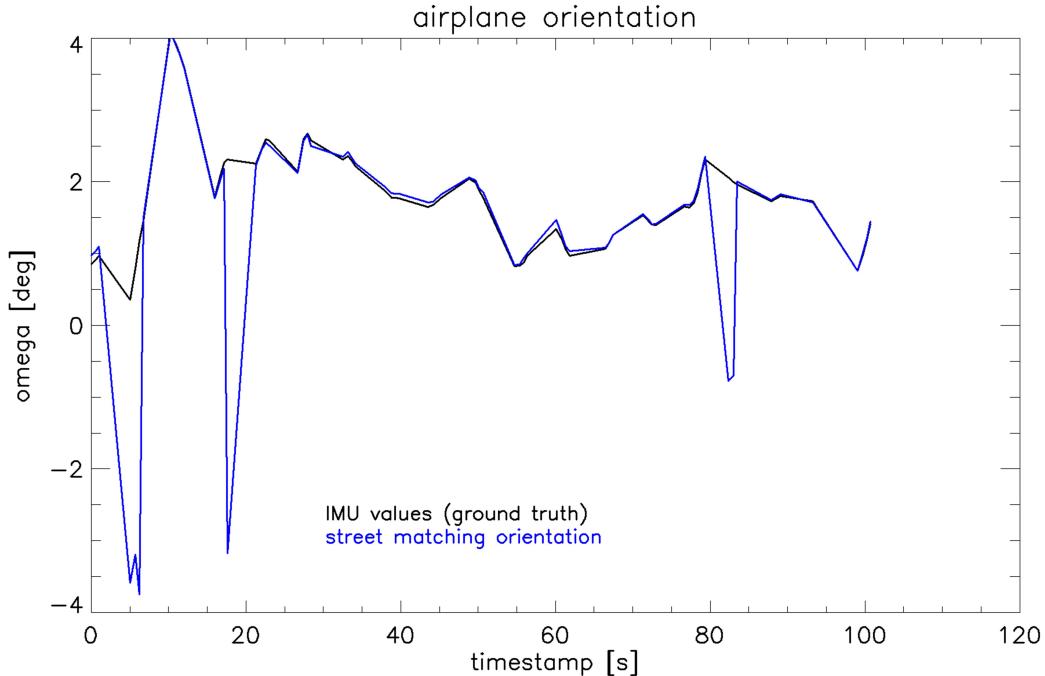
Für diese Arbeit wurde der *SURF*-Algorithmus aus der freien C-Bibliothek OpenCV verwendet, welcher in je zwei aufeinanderfolgenden Aufnahmen je nach Parametereinstellung hunderte von *Interest Points* findet. Wichtig ist dabei, dass diese Punkte nur in den überlappenden Bereichen der Aufnahmen gesucht werden, was die Chance auf Ausreißer verringert und größere Stabilität in der Ausgleichsrechnung gewährleistet.

### 7.2.2 Ausgleichung

Die Berechnung der relativen Orientierung mit Folgebildanschluss (siehe [70] S. 250 ff.) wird aus der Koplanaritätsbedingung der Epipolareometrie (Abb. 7.18) hergeleitet. Da die Basis beider Aufnahmen  $b$  sowie die Abbildungsstrahlen  $r'$  und  $r''$  eines Punktes  $P$  durch beide Projektionszentren  $O'$  und  $O''$  auf einer Ebene liegen müssen, kann diese Bedingung (Gleichung 7.3) mit mindestens drei Punktpaaren für eine Ausgleichung verwendet werden. Die Orientierung zwischen beiden Aufnahmen wird also solange optimiert, bis das Spatprodukt der drei Vektoren für alle Punktpaare minimal wird.

$$\langle b, r', r'' \rangle = (b \times r') \cdot r'' = 0 \quad (7.3)$$

Die Verbesserungsgleichungen und Differentialquotienten sind ebenfalls in [70] nachzulesen und sollen hier nicht weiter erörtert werden. Als Ergebnis der Ausgleichung erhält man die drei Rotationswinkel  $\omega$ ,  $\phi$  und  $\kappa$  sowie drei Modellkoordinaten für die Translation, welche beliebig skaliert werden können. Im gegebenen Fall ist der Translationsvektor


 Abbildung 7.12:  $\omega$ -Winkel der absoluten optischen Orientierung (Überflug B)

jedoch keine Unbekannte, da die Position, wie erwähnt, durch das GPS ermittelt wird. Daher geht die Translation als konstant in die Ausgleichung ein, und die Terme für  $y$  und  $z$  kürzen sich in der Verbesserungsgleichung weg, was die Ausgleichung stabilisiert.

Da der Algorithmus sensibel auf Ausreißer reagiert und deren Auftreten bei hunderten von Passpunktpaaren sehr wahrscheinlich ist, wird er iterativ im *RANSAC*-Verfahren [33] angewandt. Um Ausreißer zu eliminieren, wird aus der Passpunktmenge  $P$  eine kleine Menge  $S$  von  $n$  Passpunkten ausgewählt und die Optimierung darauf angewandt. Alle anderen Punkte aus  $P$ , die innerhalb einer bestimmten Fehlertoleranz liegen, werden zur Menge  $S^*$  zusammengezählt. Die Menge  $S^*$  nennt sich *consensus set* von  $S$ .

Wenn die Anzahl der Elemente in  $S^*$  einen bestimmten Schwellwert übersteigt, wird die Optimierung erneut mit der neuen Menge  $S^*$  durchgeführt. Liegt die Anzahl der Elemente in  $S^*$  unter dem Schwellwert, wird zufällig eine neue Startmenge  $S$  gewählt. Es wird jeweils die Untermenge  $S^*$  mit den meisten Elementen als bestes *consensus set* beibehalten.

Der Algorithmus kann entweder nach einer gegebenen Anzahl von Iteration abbrechen oder, nachdem eine ausreichende Anzahl von Elementen für  $S^*$  gefunden wurde. Als letzter Schritt wird die Optimierung nochmals mit der optimalen Untermenge  $S^*$  durchgeführt.

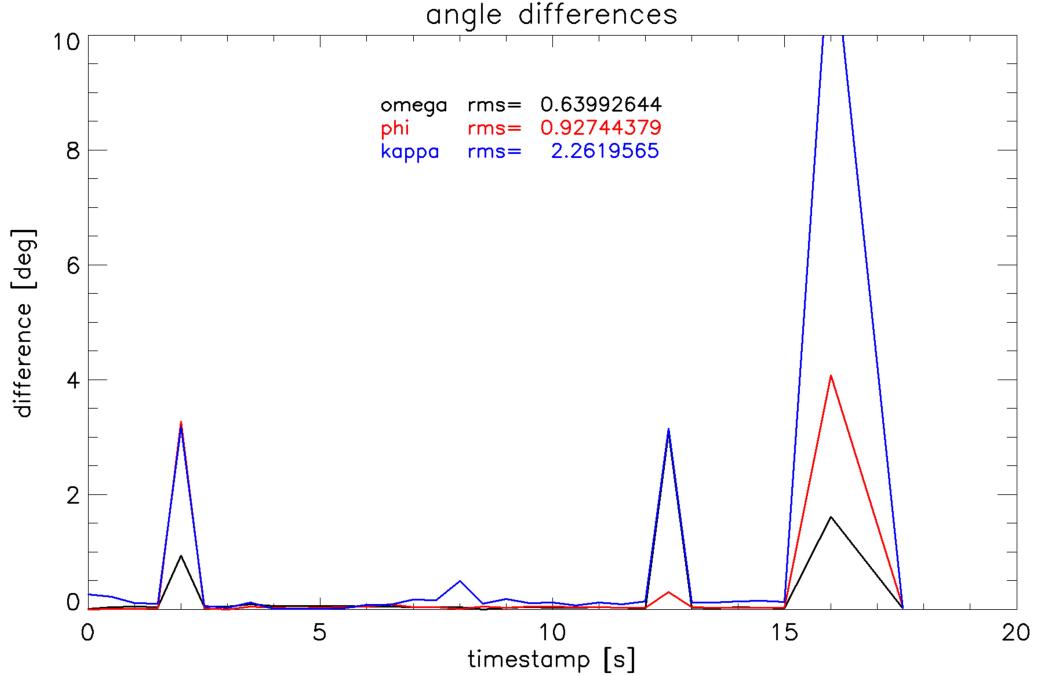


Abbildung 7.13: Winkeldifferenzen der absoluten optischen Orientierung (Überflug A)

### 7.2.3 Ergebnisse

Als Datengrundlage wurden teilweise dieselben Aufnahmen verwendet wie für die absolute Orientierung. Nur wurden diese in Überflüge von ca. 30-70 Aufnahmen aufgeteilt, da die relative Orientierung nur zwischen sich überlappenden Bildern errechnet werden kann. Zusätzlich wurden Aufnahmen aus einer anderen Flugkampagne verwendet, bei der die Kamera um 90 Grad versetzt eingebaut war. Insgesamt wurden elf Überflüge für den Test verwendet, von denen nur zwei Beispiele hier gezeigt werden. Die gesamten Ergebnisse sind ebenfalls im Anhang zu finden.

Die Abbildungen 7.20 und 7.21 zeigen den  $\omega$ -Winkel der Kamera (Rollwinkel des Flugzeuges) während der Überflüge A und B. Die relative Orientierung (rot) bezieht sich jeweils auf den ersten IMU-Wert (schwarz). Die absolute optische Orientierung wurde zum Vergleich als blau gestrichelte Linie eingezeichnet. Bei Überflug A hält sich die relative Orientierung sehr genau an die IMU-Orientierung und ist selbst nach 32 Aufnahmen nicht sichtbar abgedriftet. Bei Überflug B hingegen ist zu erkennen, dass sich die relative Orientierung zunehmend vom IMU-Wert weg bewegt, jedoch im Gegensatz zur blauen Linie keine Ausreißer aufweist.

Die Abbildungen 7.22 und 7.23 zeigen für alle drei Winkel die Differenzen der relativen Orientierung zur IMU, wobei bei Überflug B auch wesentlich stärkeres Abdriften erkennbar ist als bei Überflug A, dessen RMS-Werte um die 0,02 Grad liegen. Grund dafür ist zum einen, dass Überflug B aus mehr als doppelt so vielen Aufnahmen (71)

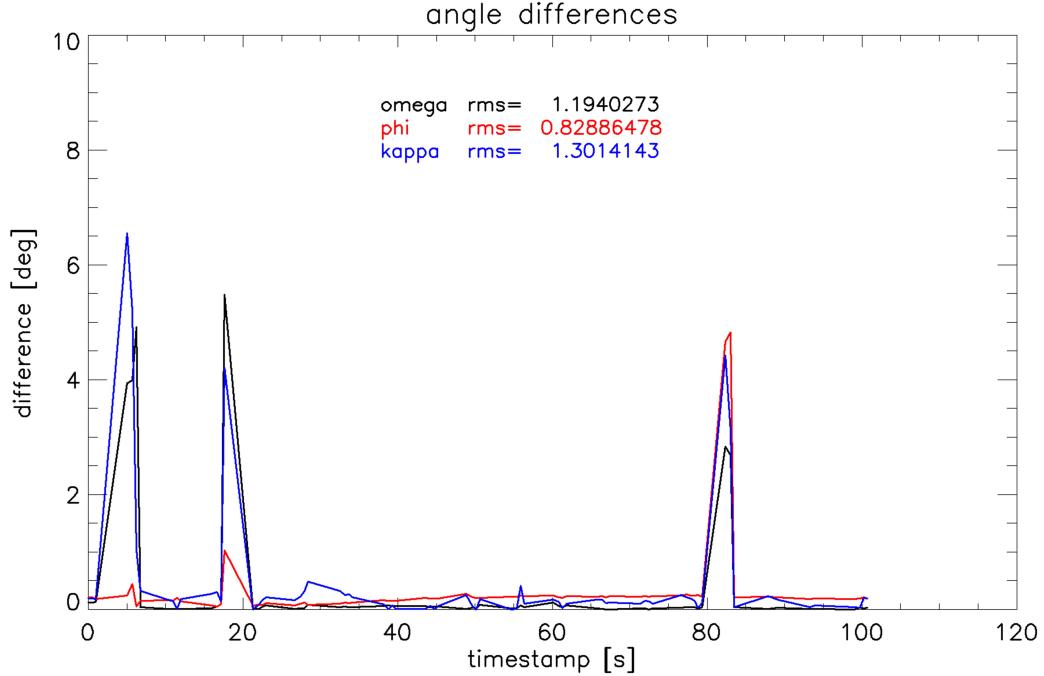


Abbildung 7.14: Winkeldifferenzen der absoluten optischen Orientierung (Überflug B)

besteht wie Überflug A. Zum anderen wurde Überflug B im Burst-Modus aufgenommen. Das bedeutet, dass nach zwei bis drei Aufnahmen eine etwa viersekündige Pause benötigt wird, um die Daten auf die Speicherplatte zu schreiben. Das bedeutet nicht nur, dass der Überflug B mit 100 Sekunden ungefähr sechsmal so lange dauert wie Überflug A, sondern auch, dass der überlappende Bereich wesentlich kleiner wird, was zu weniger *Interest Points* führt. Diese beiden Gründe erklären das stärkere Abdriften der Winkel  $\omega$  und  $\phi$  auf bis zu 0,85 Grad bei Überflug B.

## 7.3 Navigationsfilter

Wie in den vorigen Abschnitten dargelegt wurde, haben sowohl die absolute als auch die relative optische Orientierung ihre Vor- und Nachteile. Bei der absoluten treten Ausreißer auf, jedoch bleiben die Abweichungen über die gesamte Zeit konstant. Die relative Orientierung hingegen ist wesentlich genauer zwischen zwei Aufnahmen und gegen Ausreißer immun. Dafür vergrößert sich der Messfehler, je länger der Flug dauert. Die Orientierung driftet ab.

Die beste Lösung an der Stelle ist, die Vorteile beider Verfahren zu nutzen, um einen optimalen Wert zu berechnen. Ein guter Ansatz dafür ist das Kalman-Filter ([12], [15], [50], [51], [54], [56] und [101]).

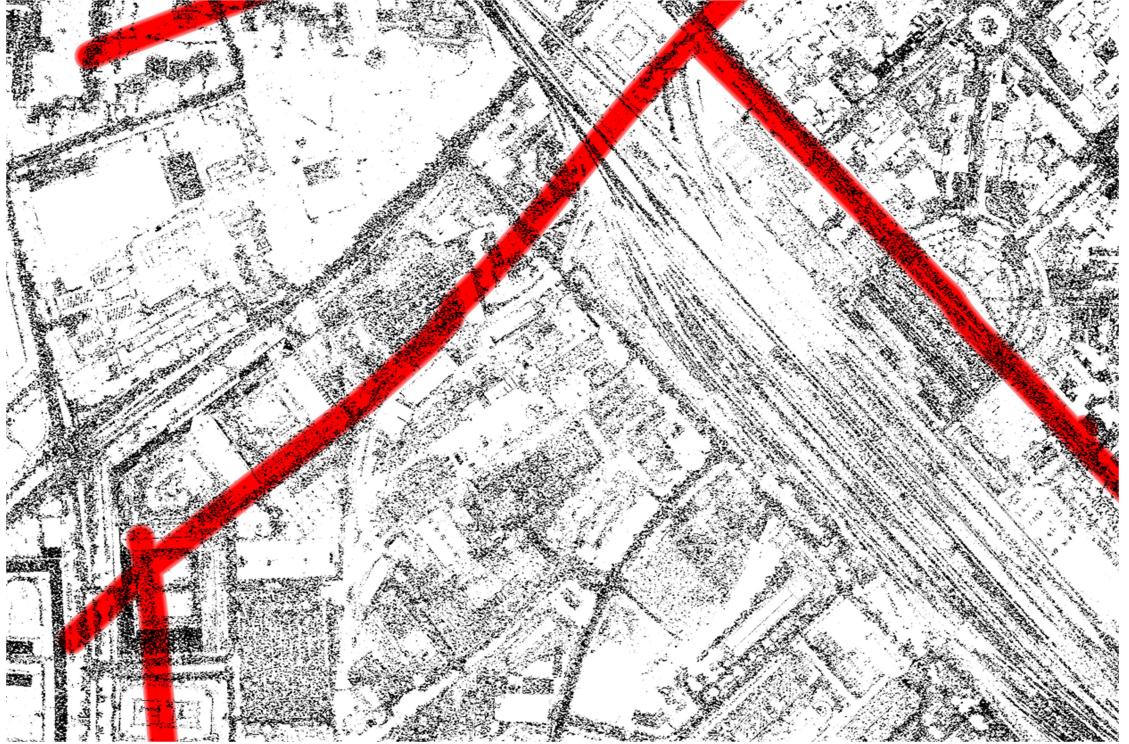


Abbildung 7.15: Falsch konvergierte Aufnahme aufgrund von zu wenig Straßeninformation

### 7.3.1 Kalman-Filter

Bereits 1960 entwickelte Rudolf Emil Kalman einen rekursiven Schätzalgorithmus für zeitdiskrete dynamische Systeme [51]. Das Kalman-Filter dient dazu, die Systemzustände auf Basis fehlerhafter Vorhersagen und Messwerte optimal zu schätzen [15].

Ausgegangen wird von einem zeitdiskreten, stochastischen Prozess in einem dynamischen System. Es wird versucht, den optimalen linearen Schätzwert des unbekannten Zustandsvektors  $\mathbf{x}_k$  anhand von gestörten Messungen  $\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_k$  zu bestimmen [50]. Das lineare System ist durch folgende Systemgleichungen definiert:

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{G}_k \mathbf{v}_k \quad (7.4)$$

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{w}_k \quad (7.5)$$

Die Systemmatrix  $\mathbf{F}$  beschreibt den Übergang des Zustandsvektors  $\mathbf{x}$  vom Zeitpunkt  $k$  nach  $k + 1$ . Die Messmatrix  $\mathbf{H}$  beschreibt die Abbildung des eigentlichen Zustandsvektors  $\mathbf{x}$  auf die beobachtbaren Größen des Messvektors  $\mathbf{y}$ .  $\mathbf{G}$  ist die Einflussmatrix des Systemrauschens, und die Vektoren  $\mathbf{v}$  und  $\mathbf{w}$  enthalten die Störgrößen des Systemrauschens sowie des Messrauschens. Sowohl System- als auch Messrauschen werden als

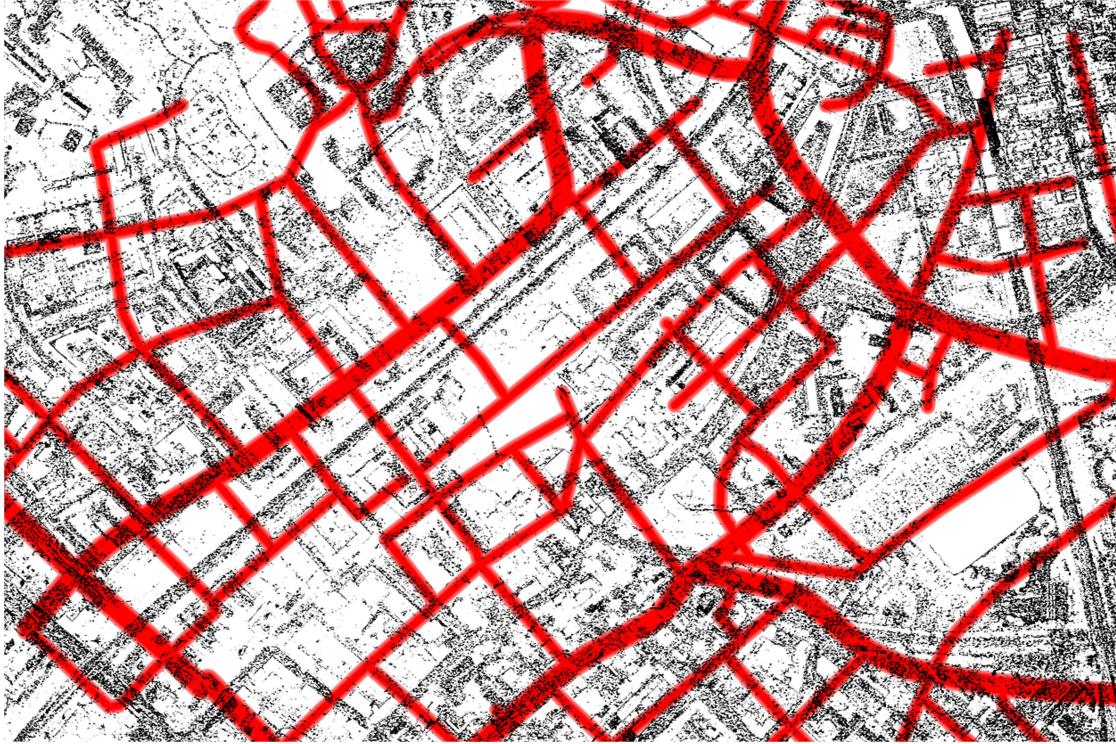


Abbildung 7.16: Falsch konvergierte Aufnahme aufgrund von zu viel Straßeninformation

additives, weißes, gaußsches Rauschen mit keinerlei Korrelation untereinander und den Kovarianzmatrizen  $\mathbf{Q}$  und  $\mathbf{R}$  angenommen.

$$E[\mathbf{v}_i, \mathbf{v}_k^T] = \begin{cases} \mathbf{Q}_k & i = k, \\ 0 & i \neq k \end{cases} \quad (7.6)$$

$$E[\mathbf{w}_i, \mathbf{w}_k^T] = \begin{cases} \mathbf{R}_k & i = k, \\ 0 & i \neq k \end{cases} \quad (7.7)$$

$$E[\mathbf{v}_i, \mathbf{w}_k^T] = E[\mathbf{x}_i, \mathbf{w}_k^T] = E[\mathbf{y}_i, \mathbf{v}_k^T] = 0 \quad (7.8)$$

Der eigentliche Filter-Algorithmus besteht aus zwei Schritten, der Prädiktion und der Innovation bzw. Korrektur.

### Prädiktion

Im Prädiktionsschritt wird zunächst mithilfe der Gleichung 7.4 ein A-priori-Schätzwert für den Systemzustand  $\hat{\mathbf{x}}_{k+1}^-$  aus dem bekannten A-posteriori-Zustand  $\hat{\mathbf{x}}_k$  ermittelt. Da das Systemrauschen als mittelwertfrei angenommen wird, ist der Erwartungswert gleich

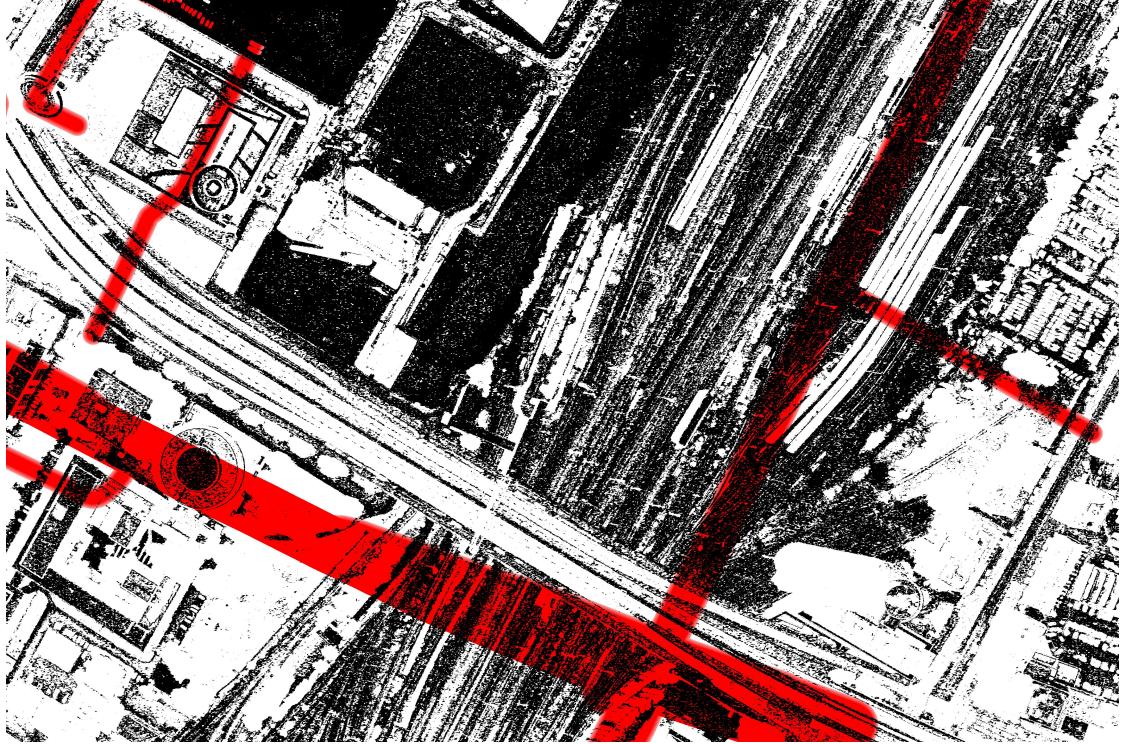


Abbildung 7.17: Falsch konvergierte Aufnahme aufgrund von falscher Straßenextraktion

Null, und die Zustandsübergangsgleichung vereinfacht sich zu

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{F}_k \mathbf{x}_k. \quad (7.9)$$

Dabei wächst die Unsicherheit an, und die Fehlerkovarianzmatrix  $\mathbf{P}$  muss um das Systemrauschen erweitert werden.

$$\mathbf{P}_{k+1}^- = \mathbf{F}_k E[(\hat{\mathbf{x}}_k - \mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T] \mathbf{F}_k^T + \mathbf{G}_k E[\mathbf{v}_k \mathbf{v}_k^T] \mathbf{G}_k^T \quad (7.10)$$

$$= \mathbf{F}_k \mathbf{P}_k \mathbf{F}_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T \quad (7.11)$$

Nach der Prädiktion wird der Index erhöht.

$$k := k + 1 \quad (7.12)$$

### Innovation / Korrektur

Im Innovationsschritt des Kalman-Filters wird der A-priori-Schätzwert mit dem Residuum korrigiert. Das Residuum ist die Differenz aus dem Messwertvektor  $\tilde{\mathbf{y}}_k$  und dem erwarteten Messwertvektor  $\hat{\mathbf{y}}_k = \mathbf{H}_k \hat{\mathbf{x}}_k$ , welche durch die zunächst unbekannte Matrix

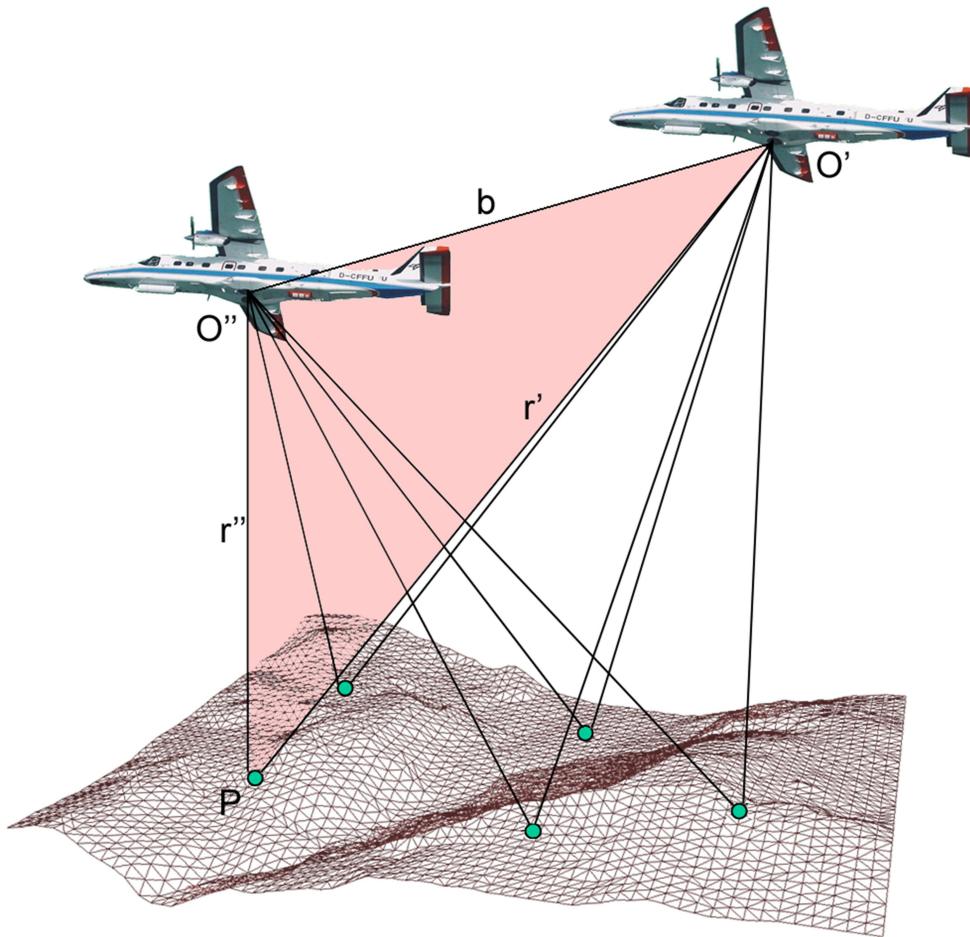


Abbildung 7.18: Epipolarebene zwischen zwei Aufnahmen

$\mathbf{K}_k$  gewichtet wird. Das ergibt folgende A-posteriori-Schätzung:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\tilde{\mathbf{y}}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) \quad (7.13)$$

Um die Gewichtsmatrix  $\mathbf{K}_k$  so zu bestimmen, dass der Schätzfehler minimal wird, muss die Spur der Kovarianzmatrix  $\mathbf{P}_k$  des Schätzfehlers minimiert werden, d.h. der quadratische Fehler wird minimiert. Dafür wird die Spur nach  $\mathbf{K}_k$  abgeleitet, um nach Nullstellen zu suchen.

$$\frac{d\text{Spur}(\mathbf{P}_k)}{d\mathbf{K}_k} = -2(\mathbf{H}_k \mathbf{P}_k^-)^T + 2\mathbf{K}_k(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k) = 0 \quad (7.14)$$

Nach  $\mathbf{K}_k$  umgestellt ergibt das folgende Gewichts- oder Kalman-Verstärkungs-Matrix:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (7.15)$$



Abbildung 7.19: Vektoren der *Interest Points* zwischen zwei Aufnahmen

Damit kann der Innovationsschritt in Gleichung 7.13 mit minimalem Fehler berechnet werden. Als letzter Innovationsschritt wird die Fehlerkovarianz  $\mathbf{P}_k^-$  korrigiert.

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \quad (7.16)$$

Damit ist ein Zyklus des Filters abgeschlossen, und der Algorithmus beginnt wieder bei Gleichung 7.9 mit der Prädiktion des nächsten Zustandsvektors  $\hat{\mathbf{x}}_{k+1}^-$ .

In Abbildung 7.24 sind die beiden Schritte des Filters und die wichtigsten Gleichungen nochmals zusammengefasst.

### 7.3.2 Anpassung des Filters

Wenn man die absolute optische Orientierung als Messvektor  $\mathbf{y}_k$  verwendet und die relative Folgeorientierung als Prädiktion  $\hat{\mathbf{x}}_{k+1}^-$ , kann ein lineares Kalman-Filter die Gesamtorientierung mit optimaler Wahrscheinlichkeit schätzen. In den Zustandsvektor  $\mathbf{x}_k$  gehen dabei nur die drei Drehwinkel ein. Die Winkelgeschwindigkeiten, mit welchen üblicherweise der Folgezustand vorausgesagt wird, werden nicht benötigt, wie später noch erläutert wird.

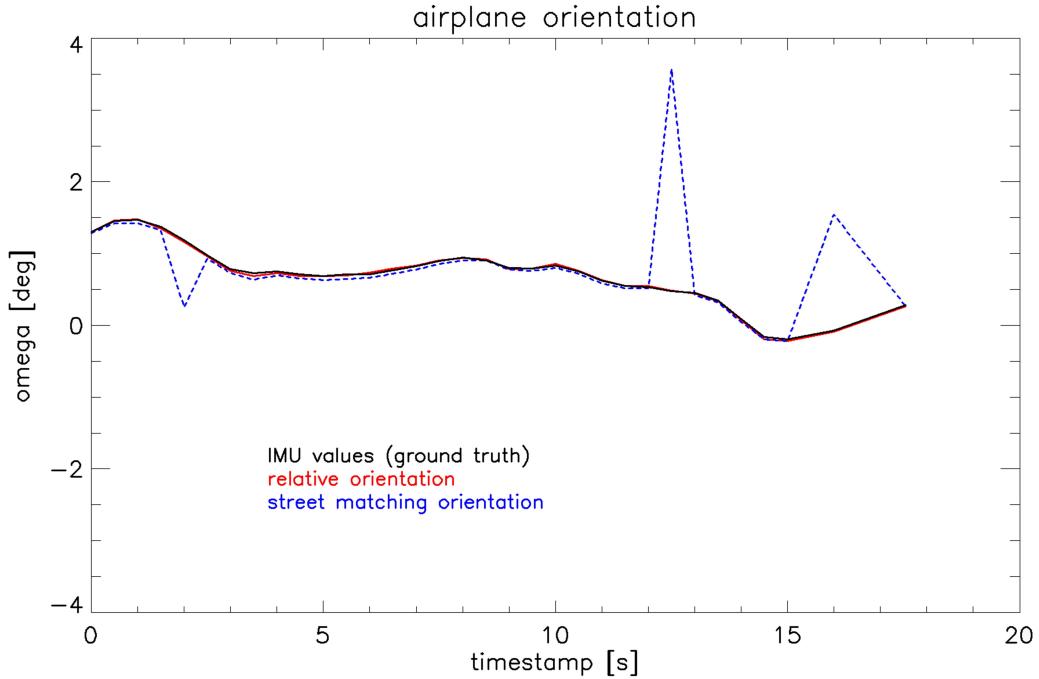


Abbildung 7.20:  $\omega$ -Winkel der relativen optischen Orientierung (Überflug A)

### Startzustand

Als initialer Zustand der Schätzung wird die absolute optische Orientierung angenommen.

$$\mathbf{x}_0 = \begin{bmatrix} \omega_{aoo} \\ \phi_{aoo} \\ \kappa_{aoo} \end{bmatrix} \quad (7.17)$$

Die Fehlerkovarianzmatrix  $\mathbf{P}_0$  wird mit den anfänglichen Varianzen der Schätzfehler gefüllt, welche zuerst noch recht hoch angenommen werden können, etwa Eins.

$$\mathbf{P}_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7.18)$$

### Prädiktion

Der Folgezustand wird nach der Überführung der Winkel in Rotationsmatrizen (Siehe Kap. 3.3) wie folgt prädiziert und nachfolgend wieder in Eulerwinkel zurückgerechnet:

$$\mathbf{M}_{k+1}^- = \mathbf{M}_k^{rel} \cdot \mathbf{M}_k \quad (7.19)$$

## 7 Optische Orientierung

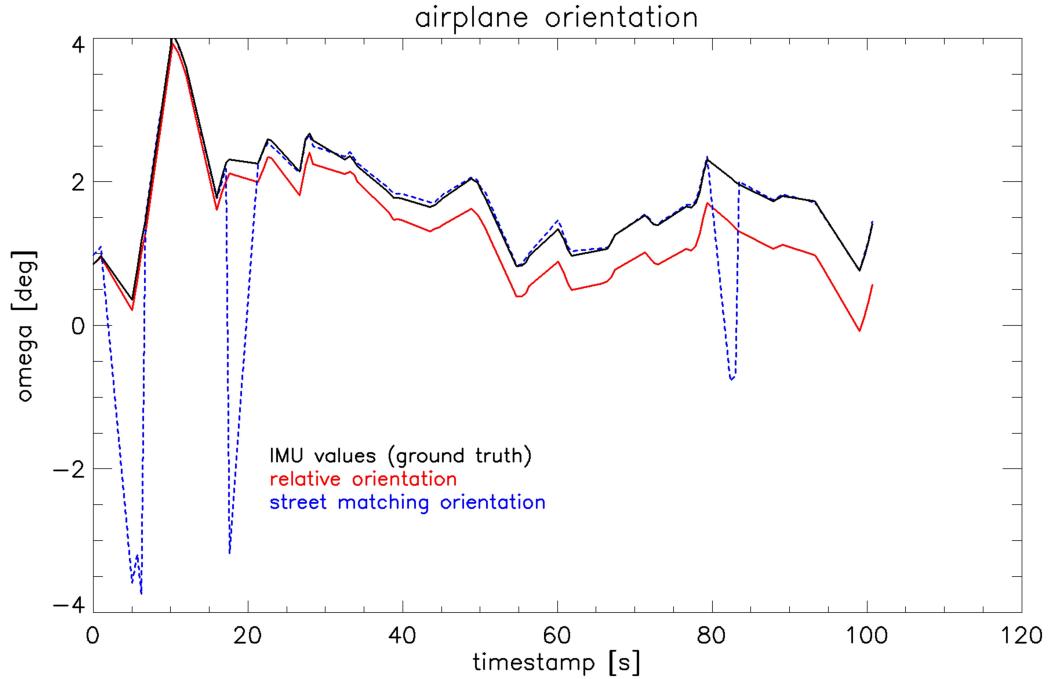


Abbildung 7.21:  $\omega$ -Winkel der relativen optischen Orientierung (Überflug B)

Als Zustandsübergangsmatrix wird jedoch eine lineare Vorhersage angenommen, welche nur den vorherigen Zustand berücksichtigt, was eine  $3 \times 3$ -Einheitsmatrix ergibt, die sich somit wegfürzt. Beim klassischen linearen Ansatz wäre der Zustandsvektor sechs-dimensional und würde auch die Winkelgeschwindigkeiten enthalten, um den nächsten Zustand vorherzusagen. Da die Vorhersage jedoch durch die relative Orientierung getroffen wird und Tests ergaben, dass das Filter in höheren Dimensionen instabiler wird, werden die Winkelgeschwindigkeiten vernachlässigt und durch das Systemrauschen abgedeckt.

$$\mathbf{F}_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7.20)$$

Die Einflussmatrix  $\mathbf{M}$  des Systemrauschen enthält lediglich die jeweilige Zeitdifferenz zwischen den Schritten.

$$\mathbf{G}_k = \begin{bmatrix} \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \end{bmatrix} \quad (7.21)$$

Die Kovarianzen des Systemrauschen  $\sigma_\omega^2$ ,  $\sigma_\phi^2$  und  $\sigma_\kappa^2$  können aus der Ausgleichsrechnung abgeleitet werden. Als Gütemaß für die Ausgleichung werden die Determinanten

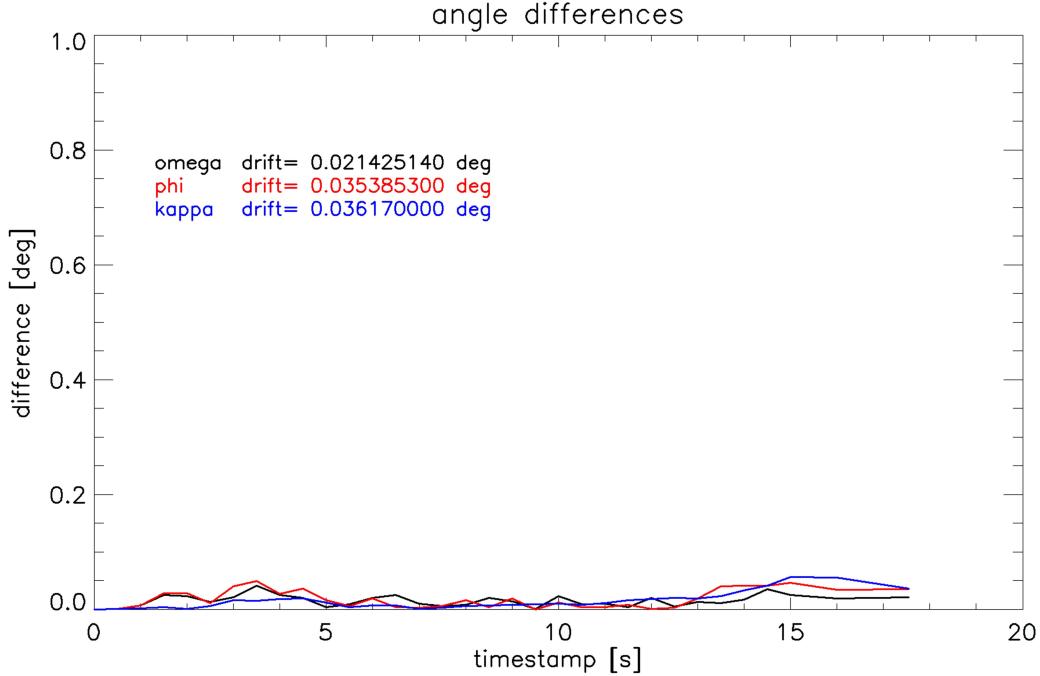


Abbildung 7.22: Winkeldifferenzen der relativen optischen Orientierung (Überflug A)

(oder Spatvolumen) für sämtliche Vektortupel errechnet. Da dieses Volumen nicht aus Kubik-Modelleinheiten besteht, sondern aufgrund des gegebenen GPS-Vektors eindeutig skaliert ist, kann damit auch die Güte zweier Ausgleichungen untereinander verglichen werden. Je kleiner das Volumen ist, desto genauer ist die relative Orientierung und desto kleiner sind auch die Werte für die Kovarianzen.

$$\mathbf{Q}_k = \begin{bmatrix} \sigma_\omega^2 & 0 & 0 \\ 0 & \sigma_\phi^2 & 0 \\ 0 & 0 & \sigma_\kappa^2 \end{bmatrix} \quad (7.22)$$

### Innovation / Korrektur

Als Messwert  $\tilde{\mathbf{y}}_k$  im Innovationsschritt wird wiederum die aktuelle absolute Orientierung angenommen, deren Varianzen auf etwa 0,5 gesetzt werden. Da genau die Größen auch gemessen werden, welche den Zustandsvektor ausmachen, also Zustands- und Beobachtungsvektor identisch sind, ist die Abbildungsmatrix  $\mathbf{H}$  ebenfalls eine Einheitsmatrix und kann vernachlässigt werden. Das Kalman-Gain kürzt sich somit auf:

$$\mathbf{K}_k = \mathbf{P}_k^- (\mathbf{P}_k^- + \mathbf{R}_k)^{-1} \quad (7.23)$$

## 7 Optische Orientierung

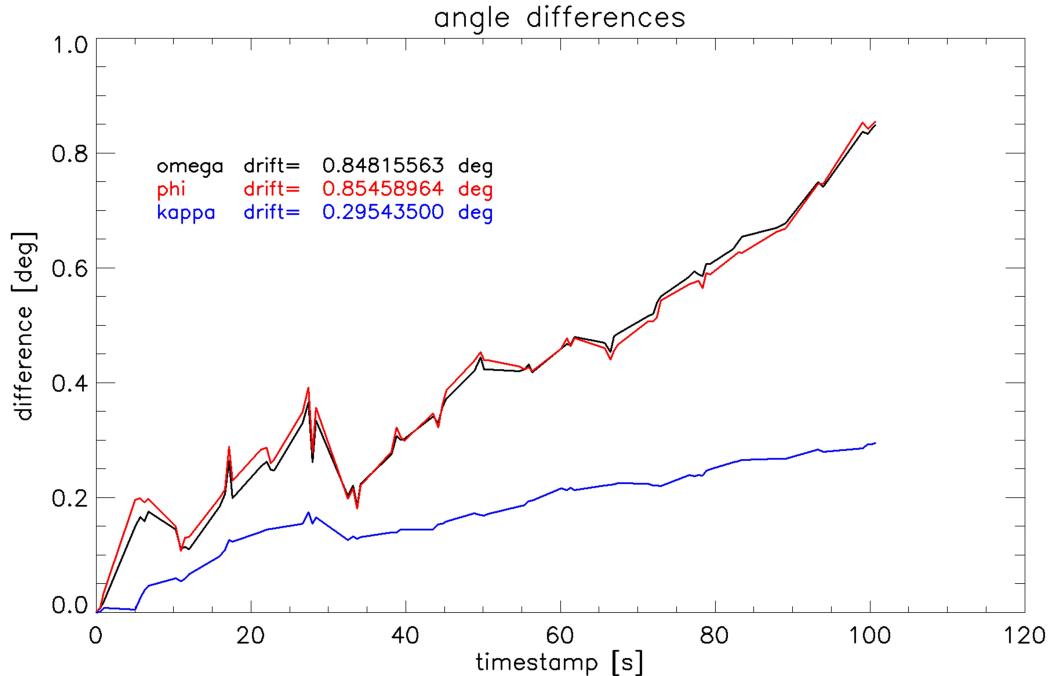


Abbildung 7.23: Winkeldifferenzen der relativen optischen Orientierung (Überflug B)

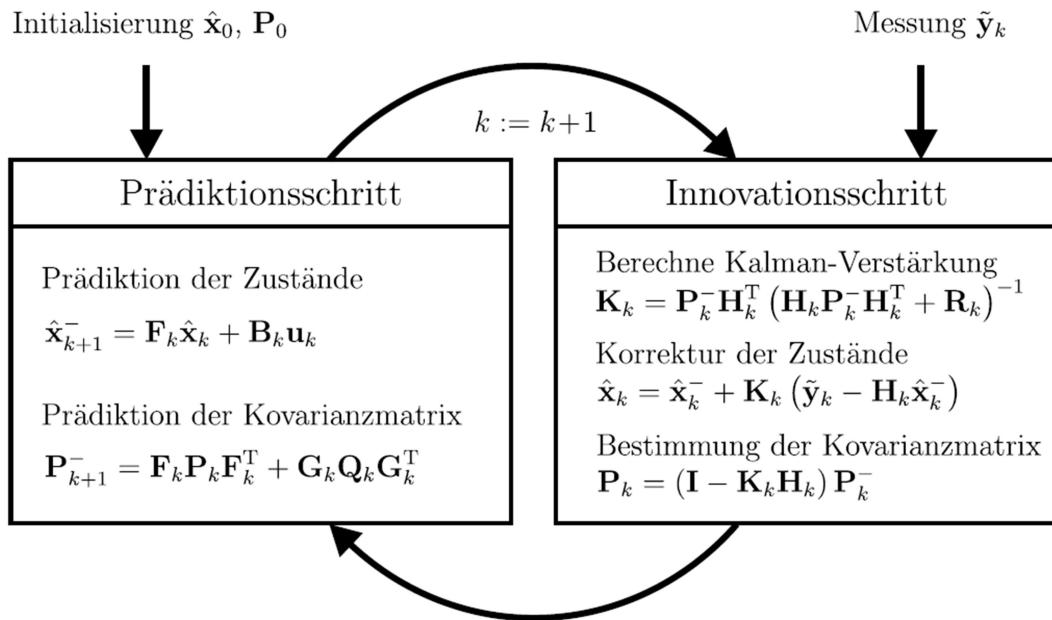


Abbildung 7.24: Ablaufdiagramm des Kalman-Filters (Quelle [101])

Dann errechnet sich die Innovation des Zustandsvektors wie folgt:

$$\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}_k^- + \mathbf{K}_k(\tilde{\mathbf{y}}_k - \hat{\mathbf{x}}_k^-) \quad (7.24)$$

Auch die Innovation der Fehlerkovarianzmatrix  $\mathbf{P}_k$  wird dadurch vereinfacht.

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k)\mathbf{P}_k^- \quad (7.25)$$

### Messfehler

Wie weiter oben (Kap. 7.1) zu lesen ist, weist die absolute Orientierung gelegentlich Ausreißer auf. Obwohl die relative Orientierung (Prädiktion) durch geringere Kovarianzen stärker gewichtet wird, kann die absolute (Innovation) das Ergebnis stark verzerrn. Um solche Ausreißer zu erkennen und zu verwerfen, wird zusätzlich ein Fehlergrenzwert verwendet, welcher nicht überschritten werden darf. Aus dem Residuum  $\mathbf{v}_k$  und den Kovarianzen der Messwertvorhersage  $\mathbf{S}_k$  wird ein quadratischer Fehlerwert errechnet [54].

$$\mathbf{v}_k = \tilde{\mathbf{y}}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^- = \tilde{\mathbf{y}}_k - \hat{\mathbf{x}}_k^- \quad (7.26)$$

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_k \mathbf{H}_k^T + \mathbf{R}_k = \mathbf{P}_k + \mathbf{R}_k \quad (7.27)$$

$$e^2 = \mathbf{v}_k \mathbf{S}_k^{-1} \mathbf{v}_k^T \quad (7.28)$$

Je vertrauenswürdiger der Messwert ist, also je kleiner die Kovarianz, desto stärker wird das Residuum im Fehlerwert  $e^2$  gewichtet. Dieser beschreibt dabei eine Chi-Quadrat-Verteilung [40] mit dem Parameter  $n = 3$  als Anzahl der Freiheitsgrade. Die Grenzen der Verteilung können aus einer zugehörigen Tabelle gelesen werden. Wenn der Fehler eine gegebene Grenze überschreitet, wird der Messwert als Ausreißer deklariert und verworfen. Die Innovation findet nicht statt, und der Folgezustand beruht einzig auf dem prädizierten Wert.

### 7.3.3 Ergebnisse

Für den Test des Filters wurden dieselben Datensätze verwendet wie in Kapitel 7.1.3 und 7.2.3. Der Vergleich soll zeigen, wie sehr sich die Anwendung des Filters auf die Genauigkeit auswirkt. Wie zu den vorigen Algorithmen finden sich weitere Ergebnisse im Anhang ab Seite 145 .

Wie die Abbildung 7.25 zeigt und auch schon in Kapitel 7.2.3 zu erkennen war, liegt bei Überflug A die relative Orientierung (rot) erstaunlich genau auf der wirklichen (schwarz). Bei Überflug B jedoch driftet die relative Orientierung aufgrund der geringeren Überlappung und der längeren Flugdauer stärker ab, wird jedoch, wie in Abbildung 7.26 zu sehen, durch die absolute Orientierung (blau) gestützt und ergibt durch das Navigationsfilter eine optimale Schätzung (grün). Die Fehler der verwendeten Beispiele sind im

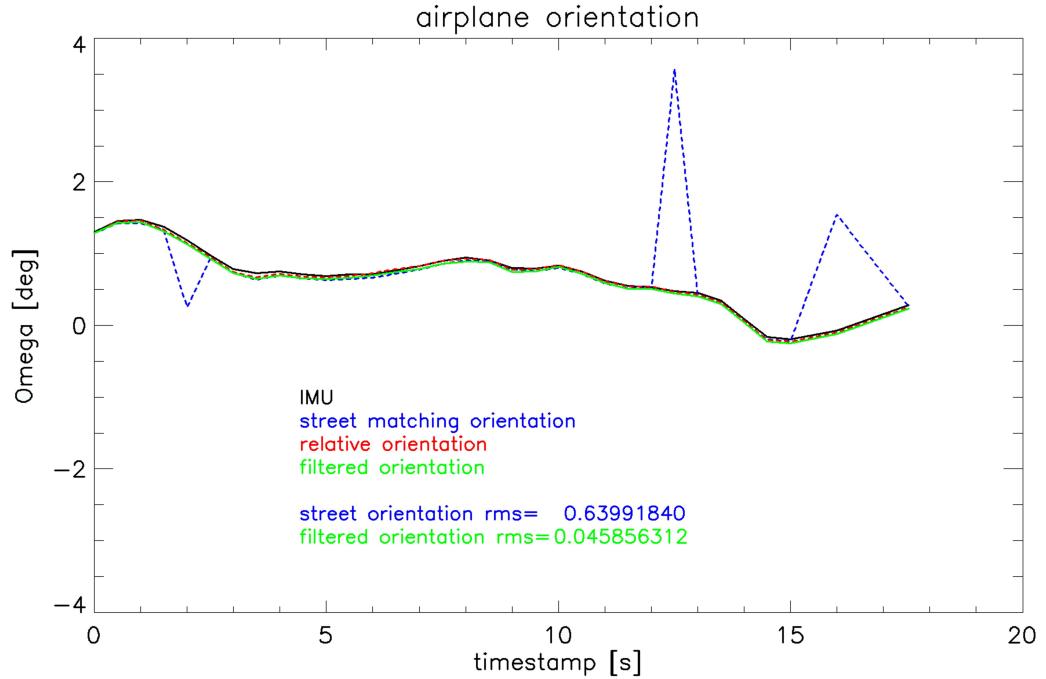


Abbildung 7.25:  $\omega$ -Winkel des Navigationsfilters (Überflug A)

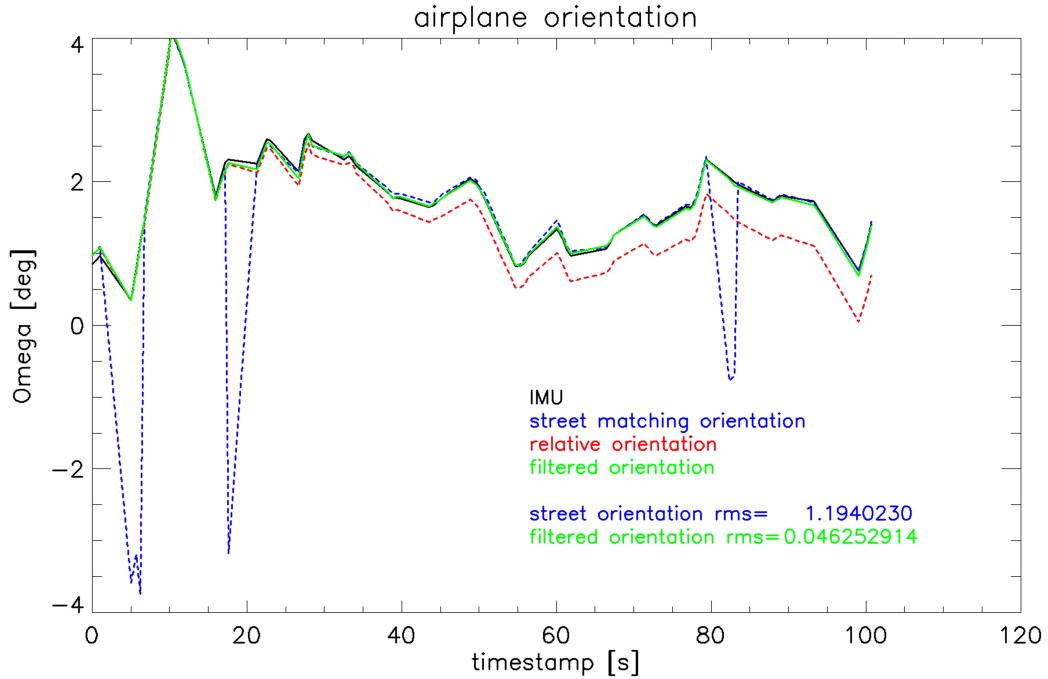
Vergleich zur alleinigen absoluten Orientierung sehr gering (Abb. 7.27 u. 7.28) und ergeben RMS-Werte von durchschnittlich 0,08 Grad (vorher 1,2 Grad), eine Verbesserung um das Fünfzehnfache.

### 7.3.4 Schwachstellen und Postprocessing

Auch wenn die meisten Ergebnisse des Navigationsfilters sehr genau sind, ist der Algorithmus in gewissen Situationen fehleranfällig. Wie schon Abschnitt 7.1.4 erwähnt, ist das Straßenmatching anfällig auf schlecht extrahierte Flächen oder andere lokale Maxima in der Optimierung. Auch wenn das Filter in der Lage ist, solche Ausreißer zu erkennen und zu verwerfen, muss es zunächst initialisiert werden.

Wenn die absolute Orientierung (nachfolgend AO) der ersten Aufnahmen falsch errechnet wird (Abb. 7.29), werden sämtliche korrekten Folgeorientierungen als Ausreißer angesehen, und das Filter hat keine Möglichkeit, sich dem wahren Wert anzunähern, da das Residuum zu groß ist. Die gefilterte Orientierung berechnet sich nur aus dem ersten Wert der absoluten Orientierung und den relativen Folgewerten, was im gezeigten Fall einen größeren RMS-Fehler ergibt als ohne Filter. Wie im Anhang ab Seite 160 zu sehen ist, wurden drei der elf getesteten Überflüge falsch initialisiert.

Auch wenn nur wenige Werte der absoluten Orientierung richtig konvergiert sind, kann der gesamte Flugstreifen in Hinblick auf die relative Orientierung (nachfolgend RO) korrigiert werden. Da die Ausreißer der AO, auch wenn es sehr viele sind, zufällig

Abbildung 7.26:  $\omega$ -Winkel des Navigationsfilters (Überflug B)

verteilte Werte annehmen, korrelieren diese nicht mit den Werten der RO. Durch eine erneute Implementierung eines *RANSAC*-Algorithmus kann der Offset der RO ausgeglichen werden. Das Verfahren wählt zunächst eine zufällig Teilmenge (*consensus set*) aus den RO-Werten aus und optimiert für diese die Winkeloffsets für den minimalen Abstand zur AO. Danach wird die Menge der RO nach weiteren Werten durchsucht, die mit den ermittelten Offsets einen bestimmten Fehlerwert zu den AO-Werten unterschreiten. All diese werden dem *consensus set* hinzugefügt. Danach beginnt der Algorithmus erneut, einen zufälligen *consensus set* auszuwählen. Nach einer gegebenen Anzahl von Iterationen wird das *consensus set* mit der maximalen Anzahl an Werten als Optimum akzeptiert. Mit diesem werden ein weiteres Mal die Offsets optimiert.

Das Postprocessing dauert je nach Anzahl der *RANSAC*-Iterationen und der Ausgleichsgenauigkeit nur Sekundenbruchteile und kann ohne weiteres während des Fluges durchgeführt werden. Es wurde auf drei falsch initialisierte Datensätze angewandt (Datensatz 8, 9 und 10) und konnte die Fehler in allen drei Fällen korrigieren (siehe Abb. 7.30). Die roten Kreuze zeigen an, welche Werte das maximale *consensus set* für die Optimierung bilden. Die ermittelten Fehlerwerte liegen durchschnittlich bei 0,17 Grad, sind also ähnlich genau wie die korrekt gefilterten Datensätze.

## 7 Optische Orientierung

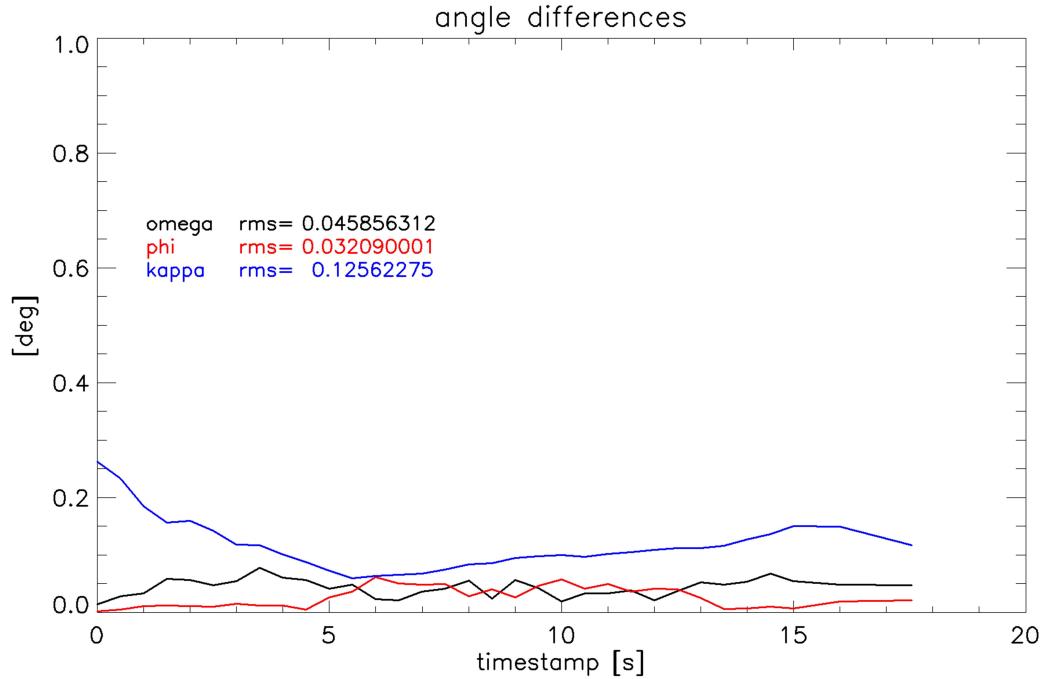


Abbildung 7.27: Winkeldifferenzen des Navigationsfilters (Überflug A)

Überflug	ungefiltert			gefiltert		
	$\omega$	$\phi$	$\kappa$	$\omega$	$\phi$	$\kappa$
1	0,134	0,022	0,114	0,151	0,027	0,098
2	0,390	0,640	1,012	0,189	0,118	0,264
3	0,101	0,189	0,279	0,180	0,144	0,111
4	1,186	0,823	1,292	0,046	0,231	0,145
5	0,145	0,884	0,372	0,083	0,208	0,169
6	0,089	0,194	0,188	0,139	0,190	0,149
7	0,630	0,913	2,227	0,046	0,032	0,126
8	2,473	3,117	6,596	0,981	3,278	8,698
9	2,058	2,566	1,946	4,309	4,970	7,566
10	2,848	1,233	3,076	3,513	0,727	3,776
11	4,435	4,936	2,069	0,254	0,310	0,135

Tabelle 7.1: RMS-Werte der elf Überflüge. (links ungefiltert, rechts Kalman-gefiltert, falsch initialisierte Überflüge sind kursiv dargestellt)

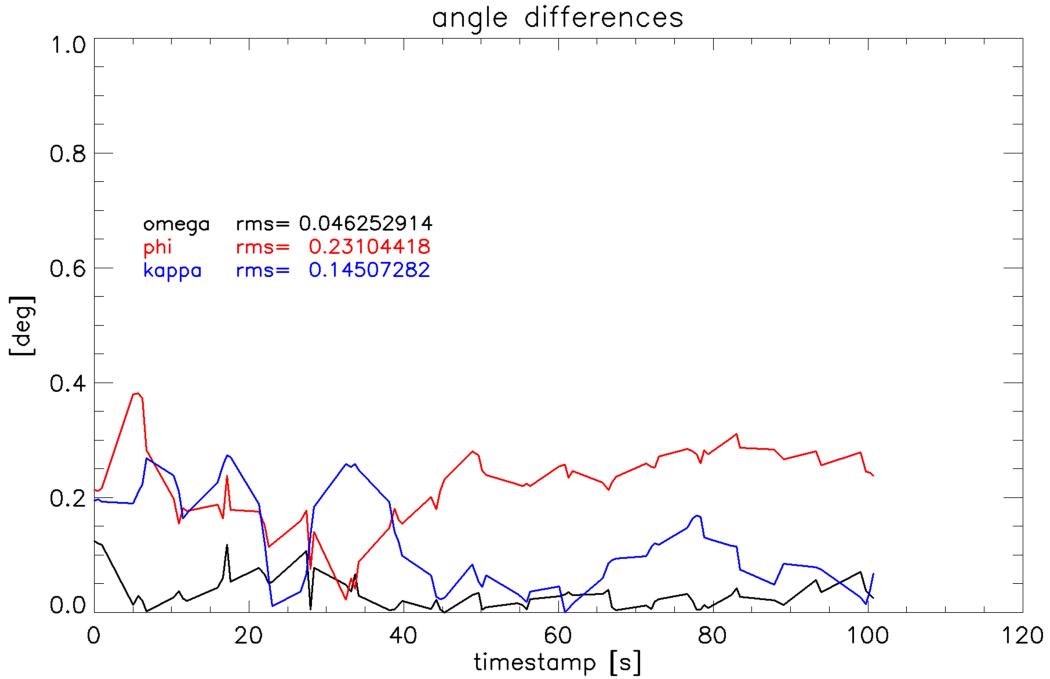


Abbildung 7.28: Winkeldifferenzen des Navigationsfilters (Überflug B)

Überflug	ungefiltert			gefiltert + RANSAC		
	$\omega$	$\phi$	$\kappa$	$\omega$	$\phi$	$\kappa$
1	0,134	0,022	0,114	0,151	0,027	0,098
2	0,390	0,640	1,012	0,189	0,118	0,264
3	0,101	0,189	0,279	0,180	0,144	0,111
4	1,186	0,823	1,292	0,046	0,231	0,145
5	0,145	0,884	0,372	0,083	0,208	0,169
6	0,089	0,194	0,188	0,139	0,190	0,149
7	0,630	0,913	2,227	0,046	0,032	0,126
8	2,473	3,117	6,596	0,174	0,133	0,138
9	2,058	2,566	1,946	0,147	0,223	0,145
10	2,848	1,233	3,076	0,136	0,188	0,215
11	4,435	4,936	2,069	0,254	0,310	0,135
gesamt	1,910	2,022	2,486	0,153	0,183	0,161

Tabelle 7.2: RMS-Werte der elf Überflüge (links ungefiltert, rechts Kalman- und RANSAC-gefiltert)

## 7.4 Zusammenfassung

Die Fusion der absoluten und relativen optischen Orientierung zeigt in den meisten Fällen eine deutliche Verringerung der Winkelfehler. In Tabelle 7.2 sind die RMS-Werte

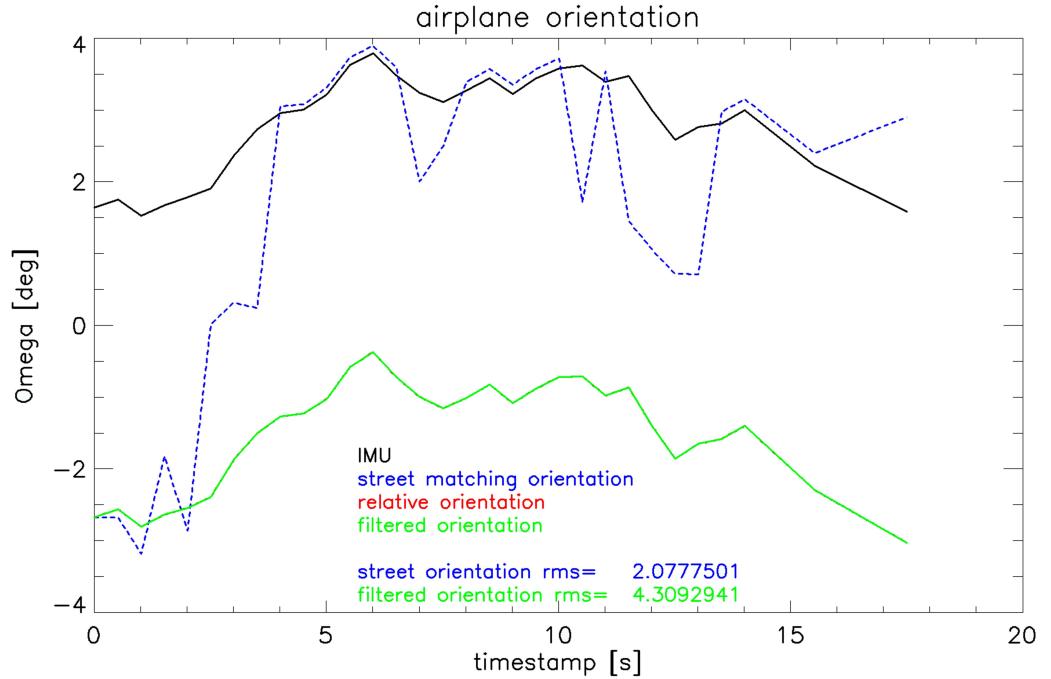


Abbildung 7.29:  $\omega$ -Winkel des Navigationsfilters (Überflug 9)

der gefilterten Daten im Vergleich zu den ungefilterten Datensätzen (siehe Tab. 7.1) dargestellt, doch sind diese nicht immer besser. Wie schon erwähnt, konnte sich das Navigationsfilter bei drei der elf Überflüge nicht richtig initialisieren, was in den beiden Fällen zu noch größeren Fehlern führte als vorher. Daher sind auch die Gesamt-RMS-Werte der ersten Tabelle nicht aussagekräftig. In Tabelle 7.2 die Überflüge 8 bis 10 RANSAC-gefiltert. Man kann eine Verringerung der Winkelfehler (RMS) auf weniger als ein Zehntel (von insgesamt 2,15 auf 0,166 Grad) beobachten. Dieser Fehler entspricht bei der Aufnahmehöhe von 1400 Metern einem Versatz von etwa 3,5 Metern am Boden. Nach [94] weisen allein schon die zugrundeliegenden NAVTEQ-Daten einen RMS von 1,7 Metern auf.

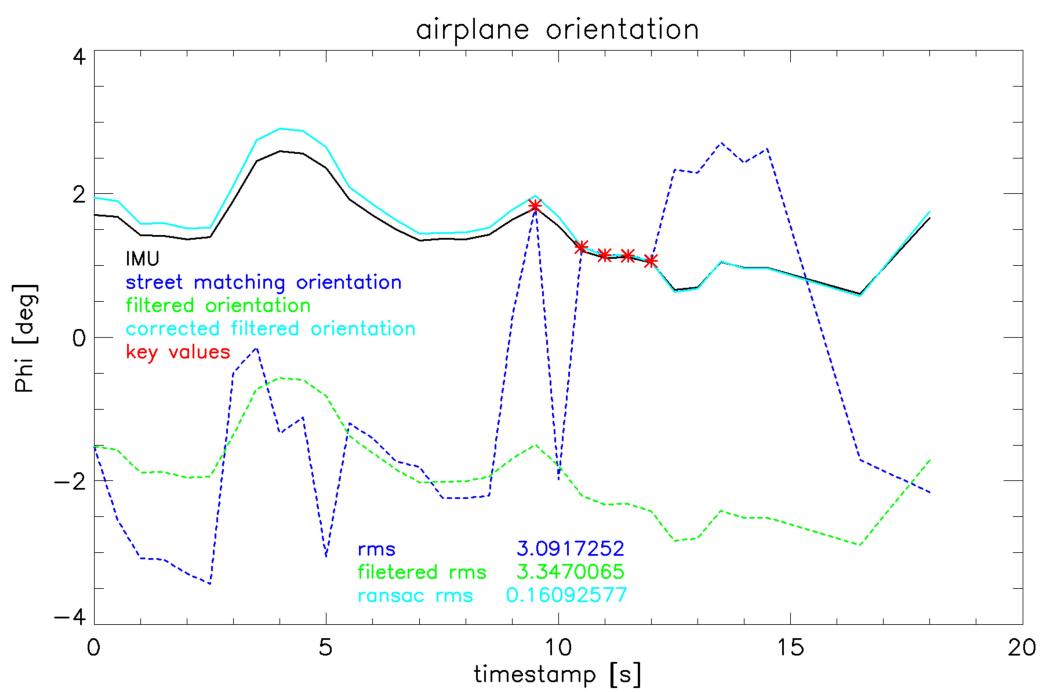


Abbildung 7.30: RANSAC-korrigierter  $\phi$ -Winkel des Navigationsfilters (Überflug 8)



## 8 Zusammenfassung und Ausblick

Die vorliegende Arbeit sollte sowohl einen umfangreichen Überblick über die luftgestützte Verkehrserfassung aus Sicht der Photogrammetrie und Bildverarbeitung geben (Kap. 1-4), als auch einen eigenen Beitrag zu dieser leisten (Kap. 5-7). Natürlich konnten dabei nicht alle Aspekte bis ins Detail berücksichtigt werden. Für sämtliche Verarbeitungsebenen wurden existierende Algorithmen vorgestellt, auf Schwachstellen untersucht und wo notwendig weiter- oder auch neu entwickelt. Nach kurzer Einleitung und Vorstellung der Hardware beginnt die eigentliche Arbeit in Kapitel 3 mit der Kalibrierung der inneren Orientierung und der Einbauwinkel der Kamera, wofür seit langem etablierte Verfahren implementiert und auf deren Genauigkeit bezüglich der Systeme *ANTAR* und *ARGOS* untersucht wurden. Dabei stellte sich heraus, dass das von *ANTAR* verwendete Inertialsystem nicht die angegebene Genauigkeit aufwies, was vermutlich auf die Vibrationen des Flugzeuges zurückzuführen ist. An dieser Stelle könnten weitere Untersuchungen Aufschluss geben.

In Kapitel 4 wurde beschrieben, wie die Luftaufnahmen in Kartenbilder transformiert werden, um auf ihnen Straßenkarten abzubilden, welche den Suchraum für die Fahrzeugerfassung eingrenzen. Dabei zeigten sich erneut Fehler, welche jedoch auf die ungenauen Koordinaten der Straßenelemente zurückzuführen sind. Es wurde ein Korrekturalgorithmus implementiert, welcher die Positionen der Straßenelemente in den meisten Fällen verbessert. Doch ist dieser, da er auf der Basis von Farbbereichen arbeitet, teilweise von den Beleuchtungsbedingungen abhängig. Eine Erweiterung um die Erkennung von Fahrbahnmarkierungen könnte die Zuverlässigkeit noch weiter erhöhen.

Kapitel 5 zeigt einen zweistufigen Erkennungsalgorithmus, welcher entwickelt wurde, um auf den sehr großen Aufnahmen die Vorteile eines schnellen und eines genauen Algorithmus zu vereinen. Das speziell geformte Kantenfilter erkennt im ersten Schritt je nach Bildqualität zwischen 60 und 90 Prozent der Fahrzeuge mit hoher Anzahl von *False Positives*, braucht jedoch im günstigsten Fall nur eine Sekunde pro Aufnahme. Im zweiten Schritt werden die Hypothesen durch eine *Support Vector Machine* überprüft, und ein Großteil der Fehlerkennungen wird verworfen. Im ungünstigsten Fall braucht die Vorselektion jedoch sieben Sekunden pro Aufnahme und ist somit nur bei geringer Aufnahmefrequenz und ausreichender Flugzeit echtzeitfähig. Großen Anteil an der Berechnungsgeschwindigkeit hat auch die Anzahl der Straßenelemente, was besonders bei Stadtaufnahmen ins Gewicht fällt.

In Kapitel 6 wurde ein Algorithmus implementiert, welcher auf Basis von räumlicher Distanz und Ähnlichkeit erkannte Objekte in benachbarten Bildern assoziiert. Tracking ist eigentlich nicht der treffende Begriff, da die Objekte aufgrund des Abbruchs der Bildaufnahme nicht weiter verfolgt werden können. Die Assoziation erfolgt mittels Singulärwertzerlegung nach einer Erweiterung des von Scott und Longuet-Higgins vorgestellten

## 8 Zusammenfassung und Ausblick

Matchingverfahrens für Stereobilder. Wenn man nach dem Matching die Paare nochmals auf glaubwürdige Distanzen und Ähnlichkeiten überprüft, ergeben sich Geschwindigkeitsabweichungen von höchstens zehn km/h zur *Ground Truth*, oft aber erheblich weniger. Eine weiterer Forschungsanreiz wäre, die Ähnlichkeit nicht aufgrund einfacher Kreuzkorrelation zu bestimmen, sondern, wie in Kapitel 5 beschrieben, komplexere Deskriptoren zu verwenden und zu vergleichen. Dieser Schritt könnte noch bessere Ergebnisse liefern. Wenn die Bildaufnahme in naher Zukunft kontinuierlich geschieht, könnten die Fahrzeuge mit einem Multi-Kalman-Filter oder einem Partikel-Filter über mehrere Aufnahmen getrackt werden.

Die Arbeit schließt mit Kapitel 7, in welchem ein alternativer Ansatz zur Orientierungsbestimmung vorgestellt wird. Es wird schrittweise gezeigt, welche Algorithmen und Verbesserungen sich wie auf die Winkelbestimmung auswirken. Um das Inertialsystem zu kompensieren, werden Straßen extrahiert und mit den ins Bild projizierten Straßenelementen gematcht. Ein Ausgleichsalgorithmus optimiert die Rotationswinkel des Systems mit einer Genauigkeit von 0,1 bis 0,2 Grad. Durch Passpunktverfolgung wird eine zusätzliche relative Orientierung bestimmt, welche in einem Kalman-Filter mit der Straßenorientierung fusioniert wird, um Ausreißer zu erkennen. Dabei treten in manchen Fällen Probleme bei der Initialisierung des Filters auf, wenn die Anfangsorientierung falsch ermittelt wurde. Eine Verbesserung wäre einerseits wie in Kapitel 4 durch eine aufwändigere Straßenextraktion möglich. Im gezeigten Ansatz wird für falsch initialisierte Datensätze nach dem Filtern per *RANSAC* ein optimales Offset ermittelt, welches die Rotationsfehler auch in diesen Fällen korrekt bestimmt. Somit konnte die Orientierung des Flugzeuges für alle elf vorhandenen Datensätze bestimmt werden.

Interessant wäre beispielsweise zu untersuchen, inwiefern andere Objekte mit bekannten Koordinaten zur automatischen Orientierungsbestimmung herangezogen werden können, wie es z.B. in [30] auf Gebäudemodelle angewandt wurde. Ist die Orientierungsbestimmung in Gebieten ohne Straßen in ähnlicher Weise durch das Matching von Orthofotos und Höhenmodellen möglich? Würde eine zuverlässigere Straßenextraktion die Bestimmung der relativen optischen Orientierung oder die Anwendung eines *RANSAC* überflüssig machen? Kann die Genauigkeit der Orientierung durch andere Datenquellen (Atkis, Tele Atlas, OpenStreetMap) noch verbessert werden? Wie genau lässt sich die Flugzeugposition in der Ausgleichsrechnung bestimmen, wenn keine GPS-Daten verwendet werden?

Zur Innovation dieser Arbeit zählt der zweistufige Fahrzeugerkennungsalgorithmus, der im Vergleich zu anderen Ansätzen wie [98] oder [43] sich dadurch abgrenzt, dass er nicht nur Zuverlässigkeit sondern auch Geschwindigkeit optimiert, was bei den gegebenen Datenmengen von großer Bedeutung ist. Interessant wäre eine Untersuchung der Zuverlässigkeit und Geschwindigkeit sämtlicher genannter Erkennungsalgorithmen auf ein und demselben Datensatz.

Die zweite Innovation dieser Arbeit ist die Entwicklung einer vollautomatischen optischen Orientierungsbestimmung, welche zwar ähnlich wie in [30] Modelle in die Aufnahmen projiziert, jedoch durch die Verwendung von Straßen anstatt von Gebäuden in weit mehr Gegenden, insbesondere zur Verkehrserfassung, anwendbar ist.

Wenn auch zum Ende dieser Arbeit einige Fragen offen bleiben, wurden doch die wich-

tigsten Werkzeuge und Algorithmen zur luftgestützten Verkehrslageerfassung aufgezeigt, weiterentwickelt und ausgewertet. Insbesondere das optische Navigationsfilter im Kapitel 7 ein neuartiger Ansatz zur automatischen Orientierungsbestimmung und sollte als Grundlage für weitere Forschungen in dieser Richtung genutzt werden. Die Datenfusion aus absoluter Orientierung an bekannten Landmarken und relativer Orientierung durch Passpunktverfolgung scheint ein vielversprechender Ansatz zu sein.



# **Anhang A**

## **Ergebnisse der absoluten optischen Orientierung**

## Überflug 1

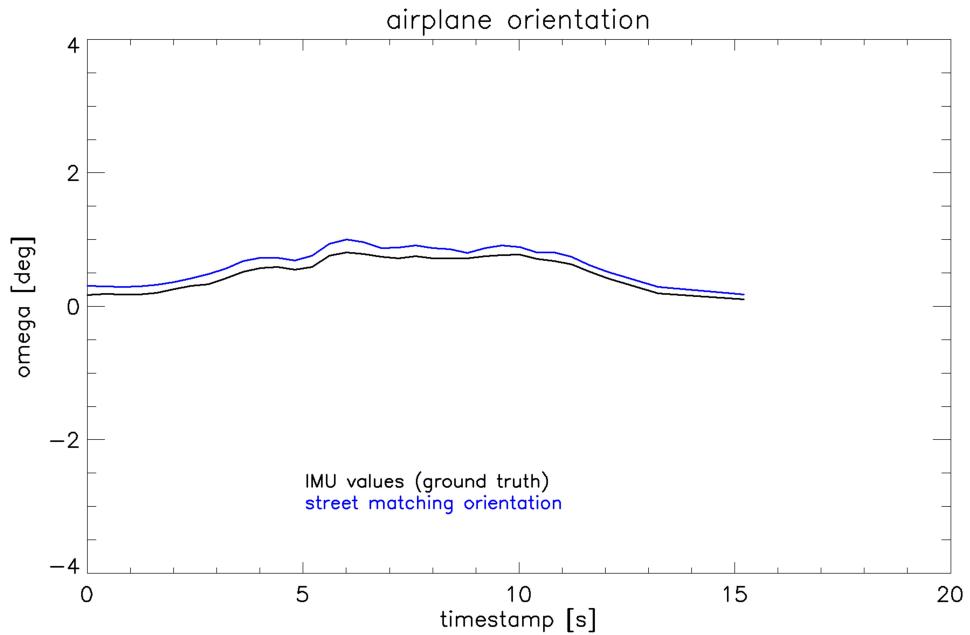


Abbildung 1:  $\omega$ -Winkel der absoluten Orientierung (Überflug 1)

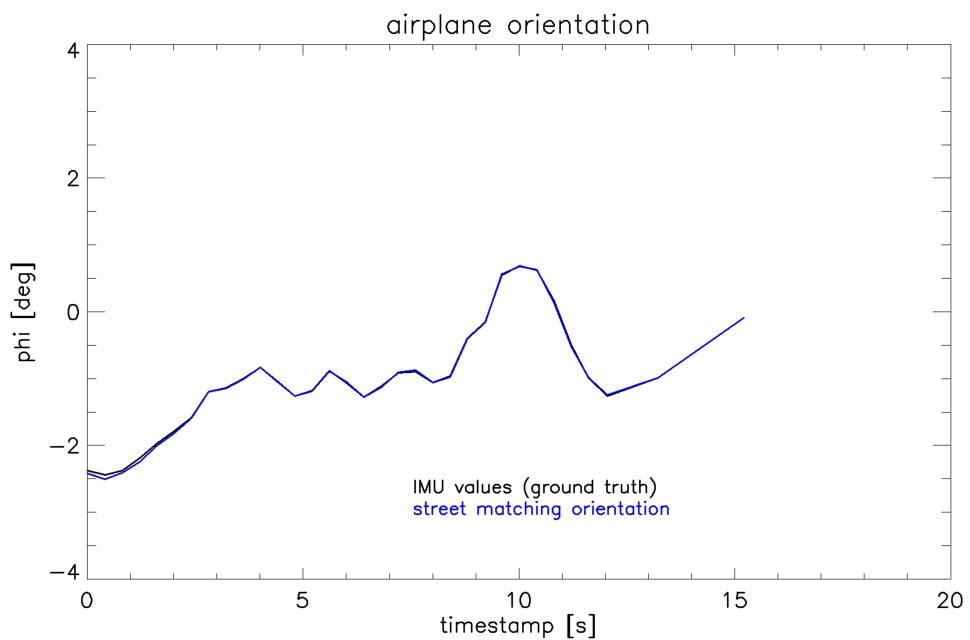


Abbildung 2:  $\phi$ -Winkel der absoluten Orientierung (Überflug 1)

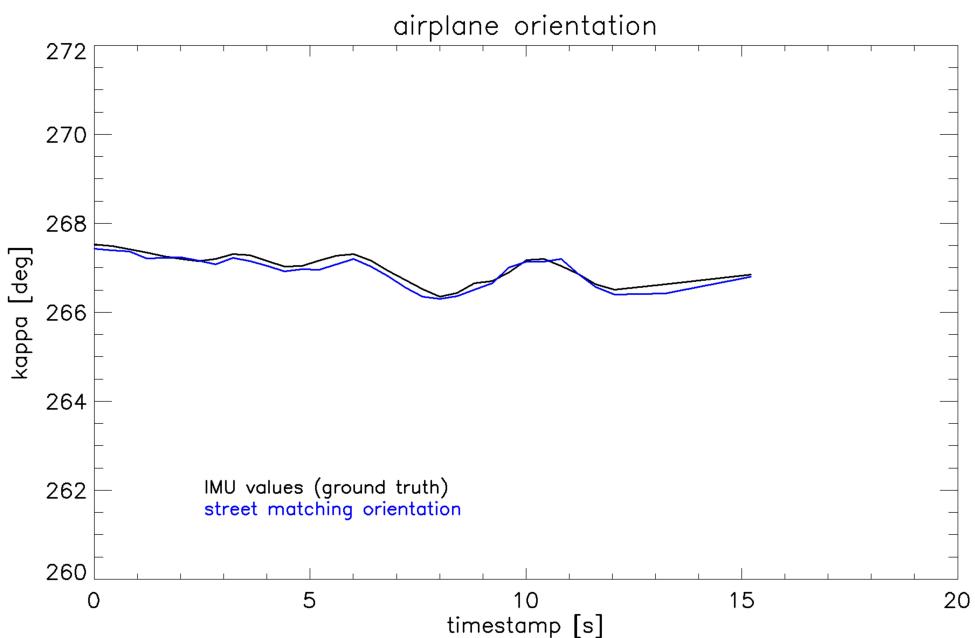


Abbildung 3:  $\kappa$ -Winkel der absoluten Orientierung (Überflug 1)

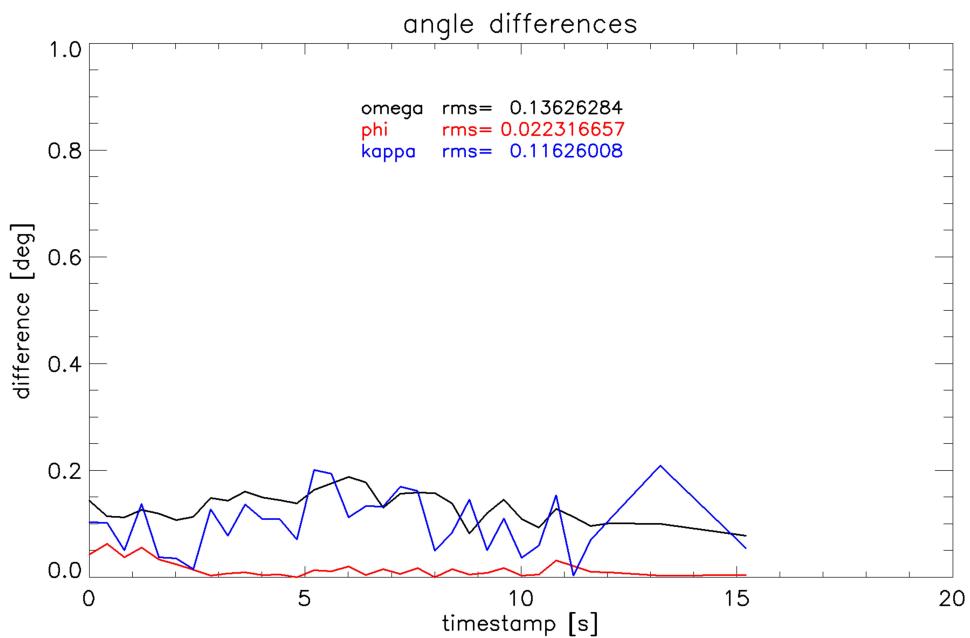


Abbildung 4: Winkelfehler der absoluten Orientierung (Überflug 1)

## Überflug 2

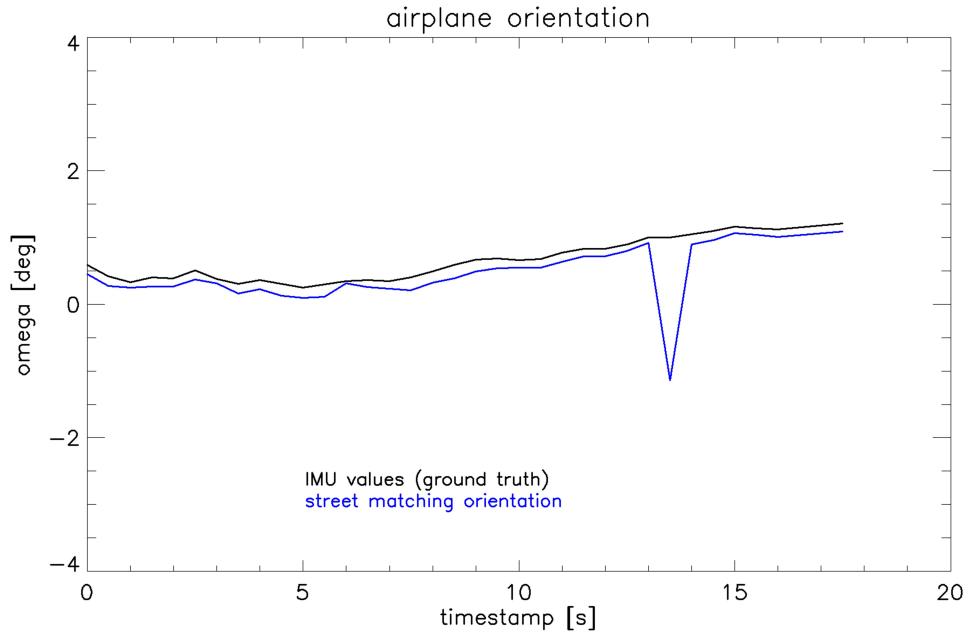


Abbildung 5:  $\omega$ -Winkel der absoluten Orientierung (Überflug 2)

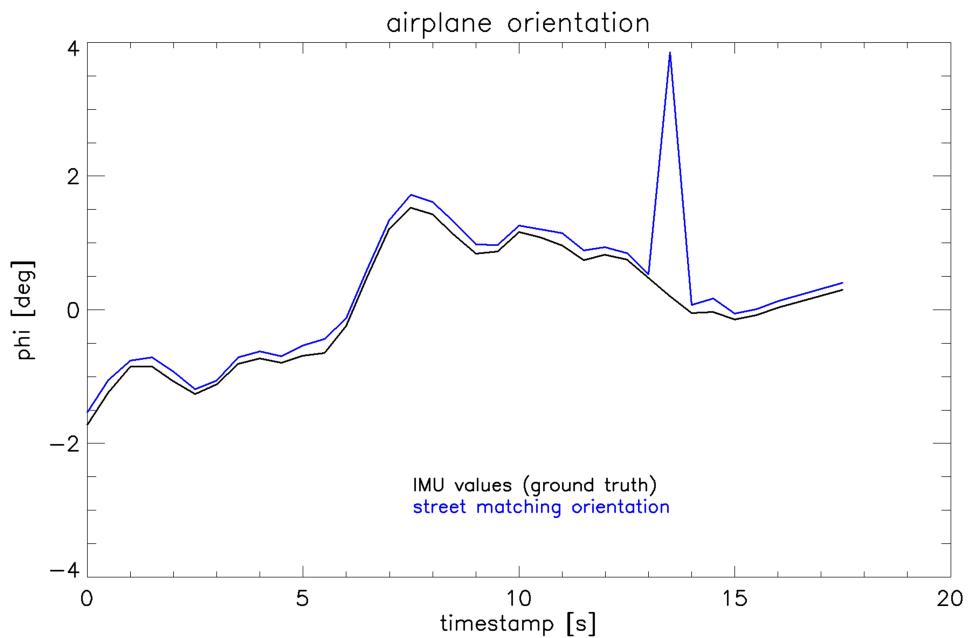


Abbildung 6:  $\phi$ -Winkel der absoluten Orientierung (Überflug 2)

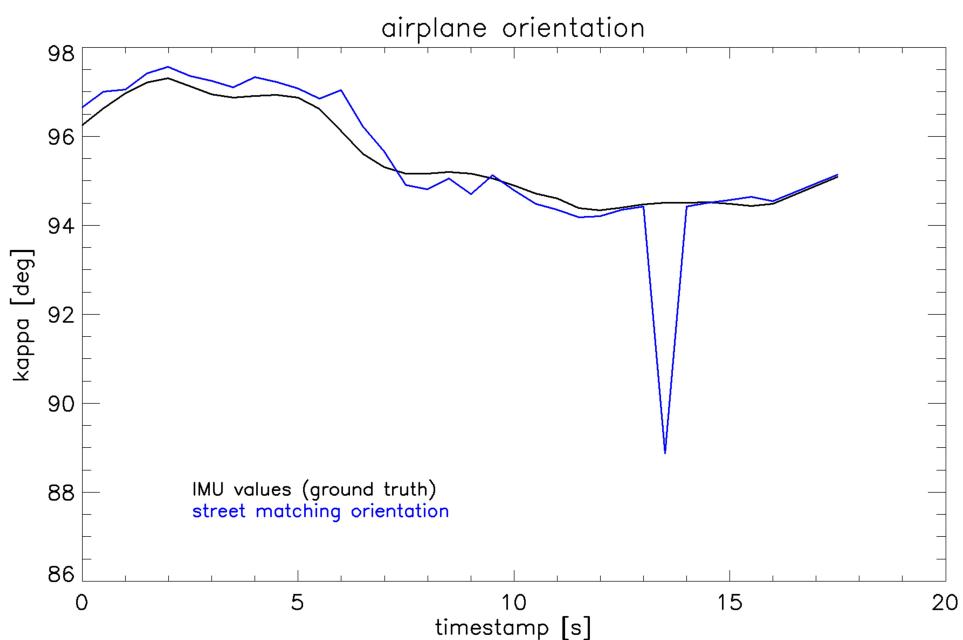


Abbildung 7:  $\kappa$ -Winkel der absoluten Orientierung (Überflug 2)

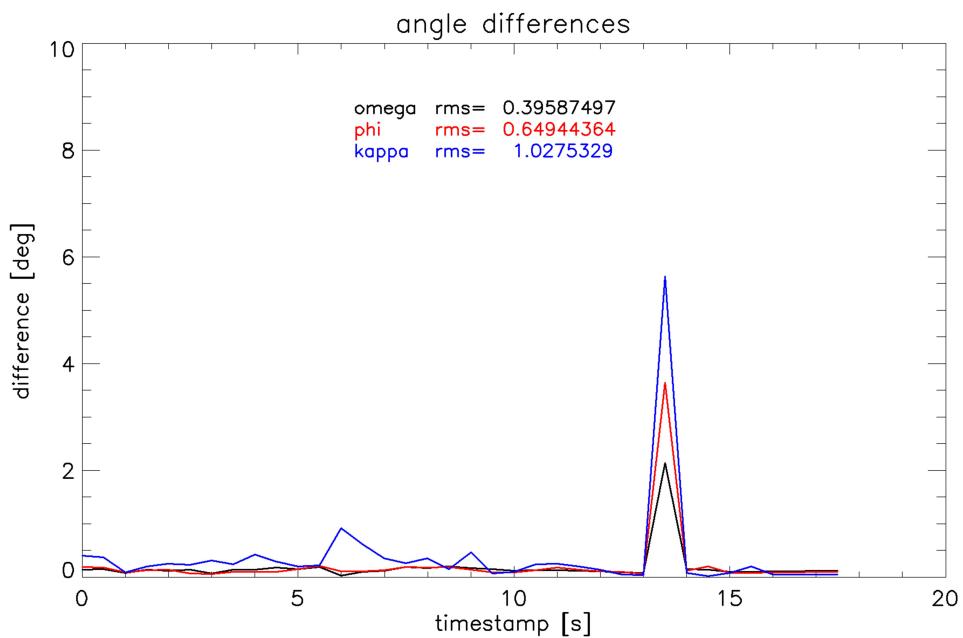


Abbildung 8: Winkelfehler der absoluten Orientierung (Überflug 2)

### Überflug 3

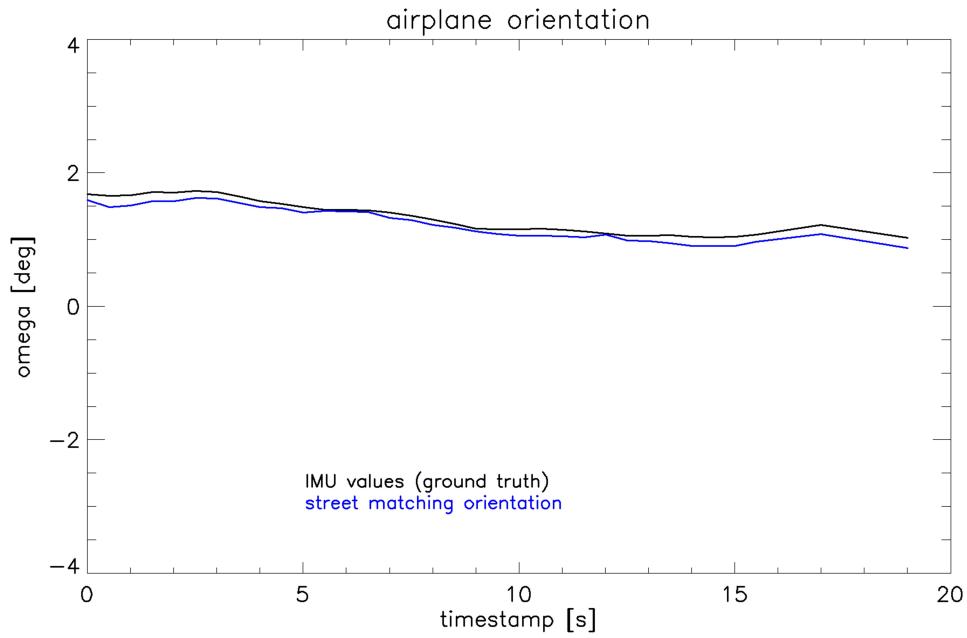


Abbildung 9:  $\omega$ -Winkel der absoluten Orientierung (Überflug 3)

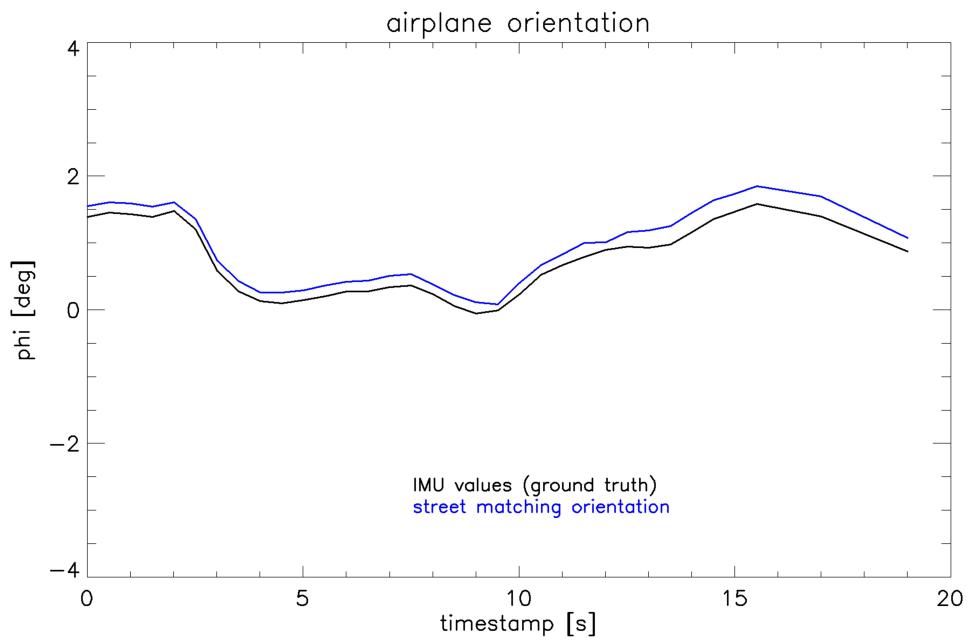


Abbildung 10:  $\phi$ -Winkel der absoluten Orientierung (Überflug 3)

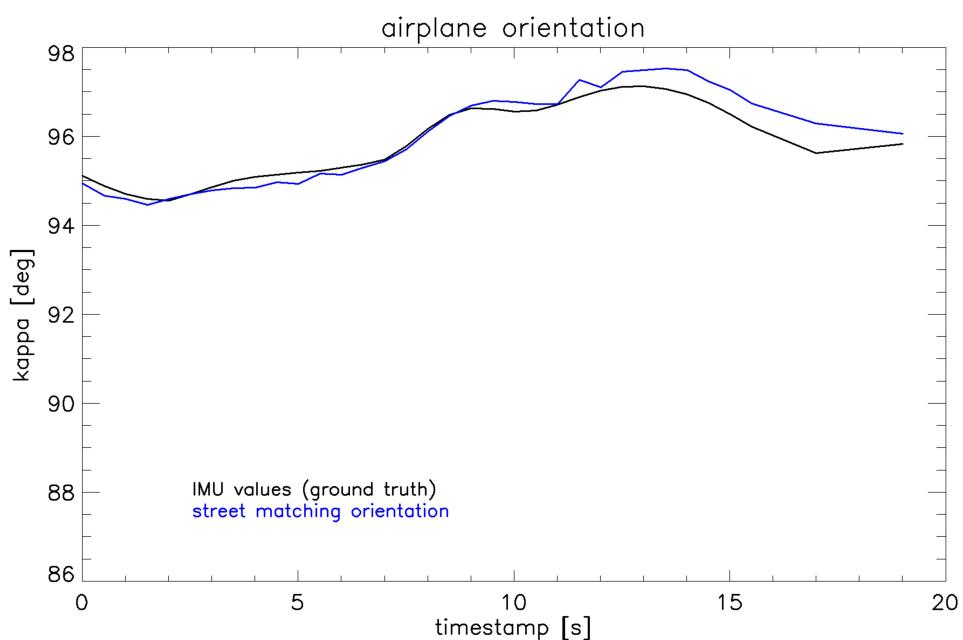


Abbildung 11:  $\kappa$ -Winkel der absoluten Orientierung (Überflug 3)

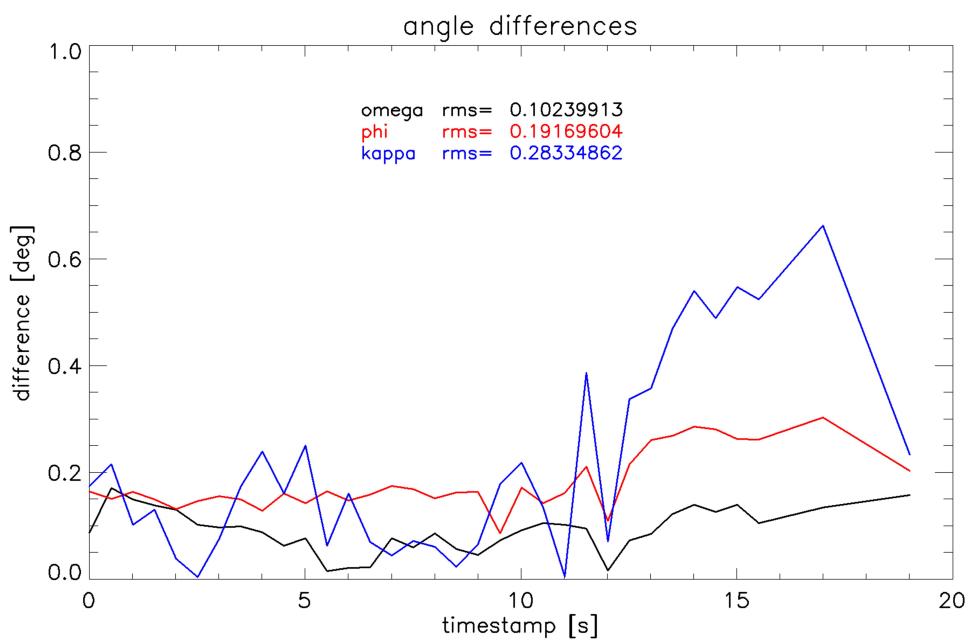


Abbildung 12: Winkelfehler der absoluten Orientierung (Überflug 3)

### Überflug 4

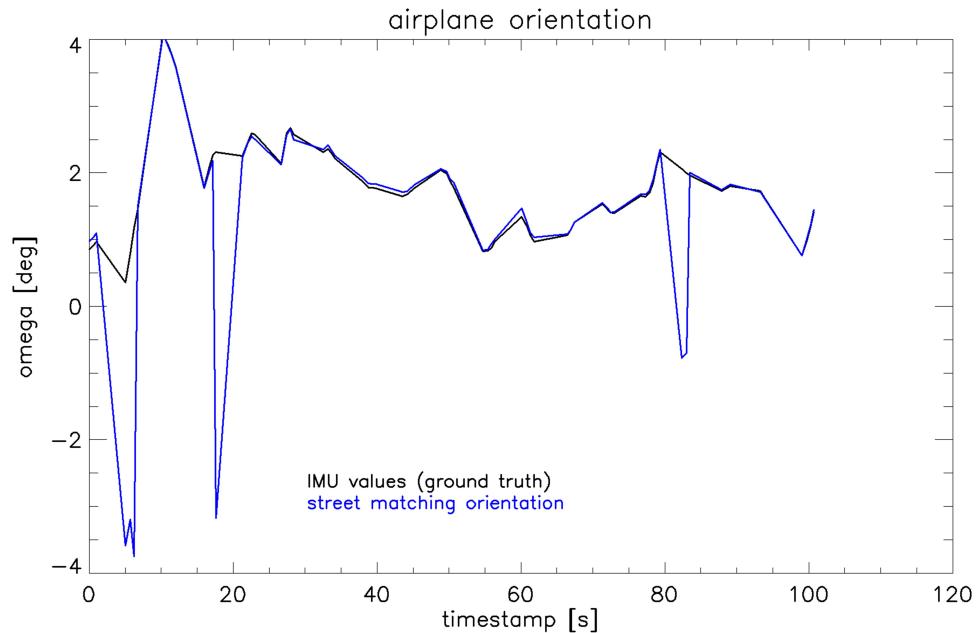


Abbildung 13:  $\omega$ -Winkel der absoluten Orientierung (Überflug 4)

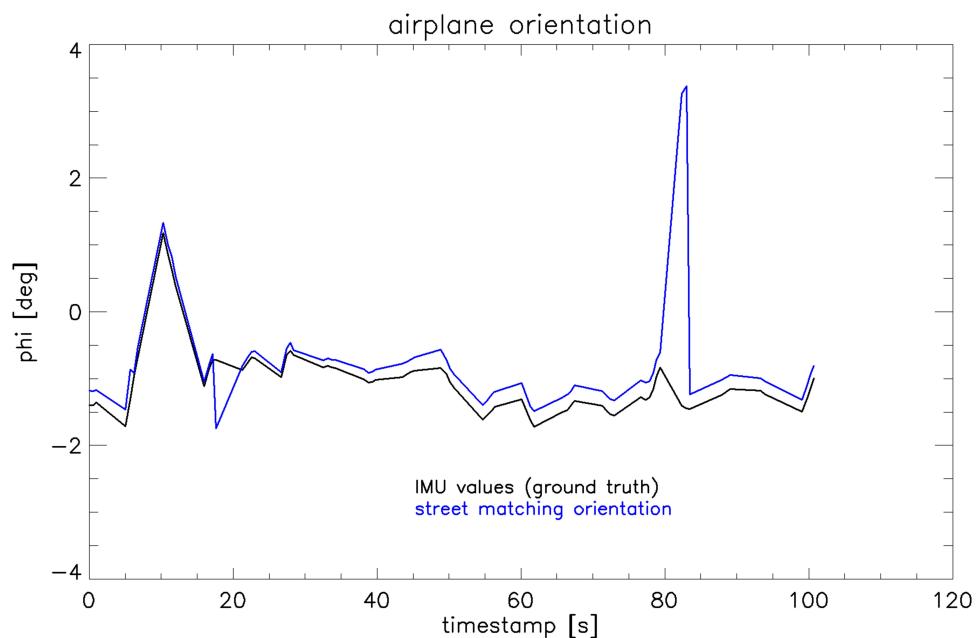


Abbildung 14:  $\phi$ -Winkel der absoluten Orientierung (Überflug 4)

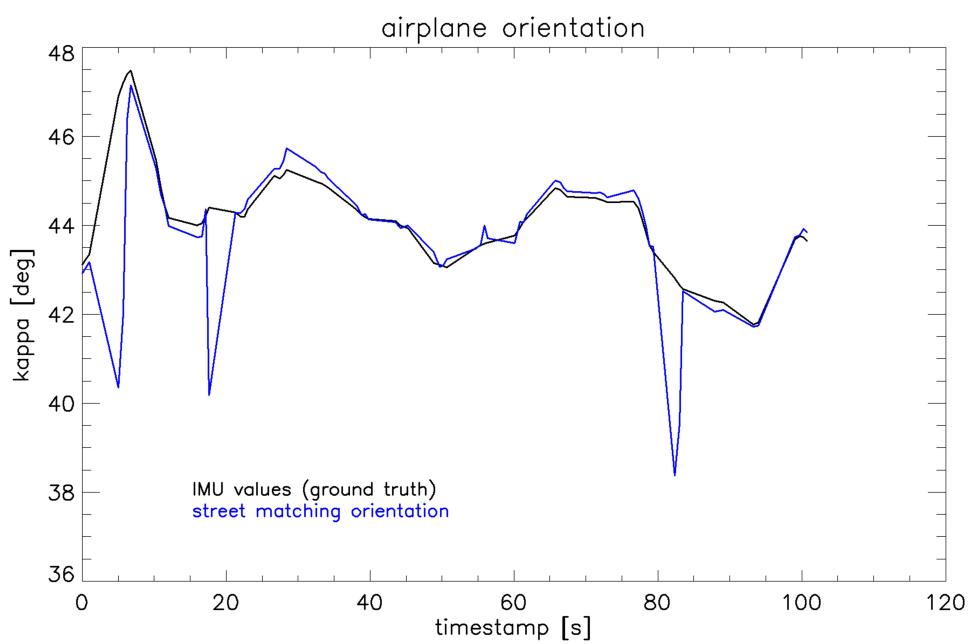


Abbildung 15:  $\kappa$ -Winkel der absoluten Orientierung (Überflug 4)

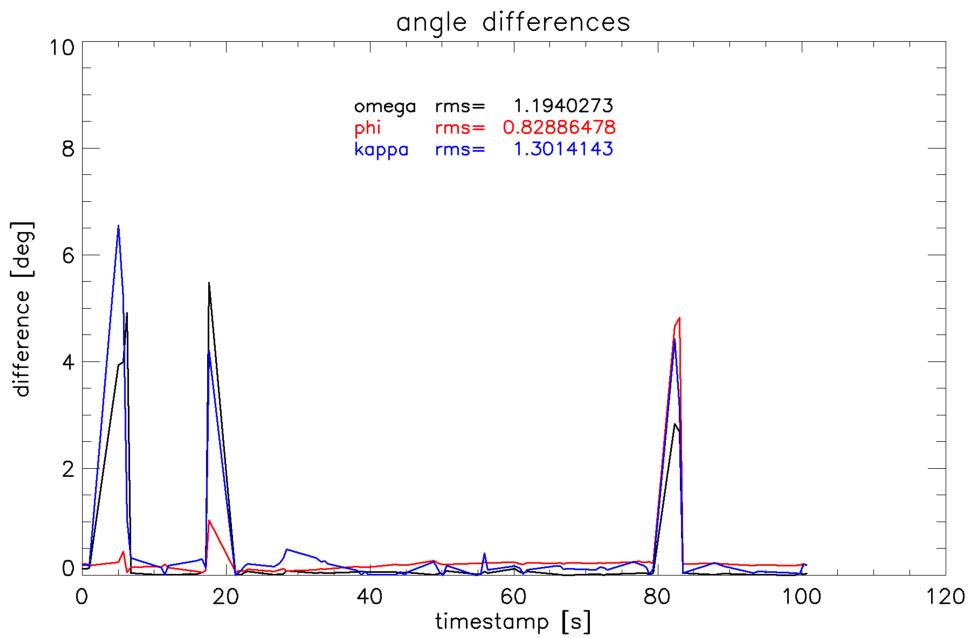


Abbildung 16: Winkelfehler der absoluten Orientierung (Überflug 4)

### Überflug 5

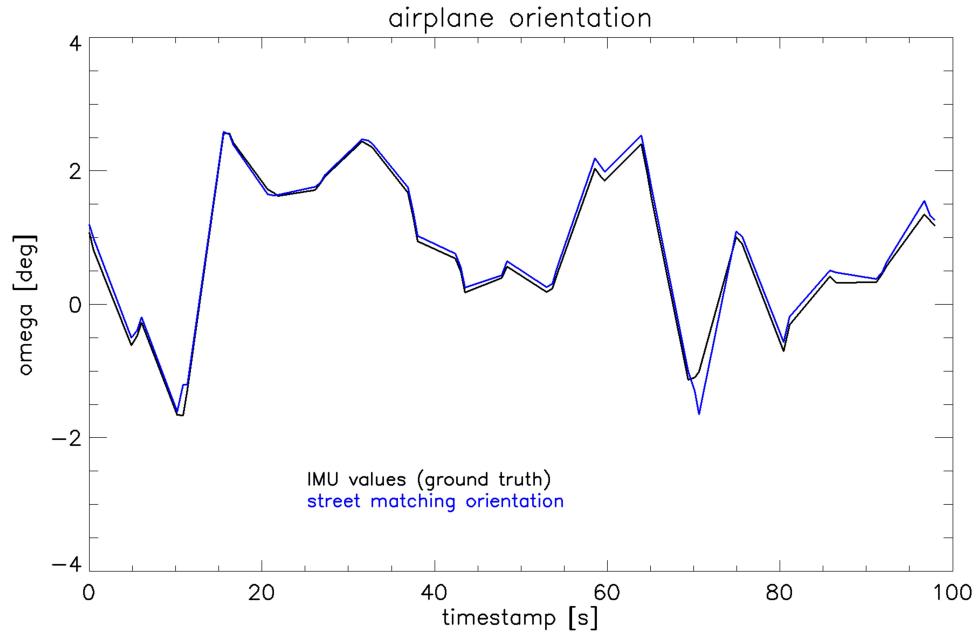


Abbildung 17:  $\omega$ -Winkel der absoluten Orientierung (Überflug 5)

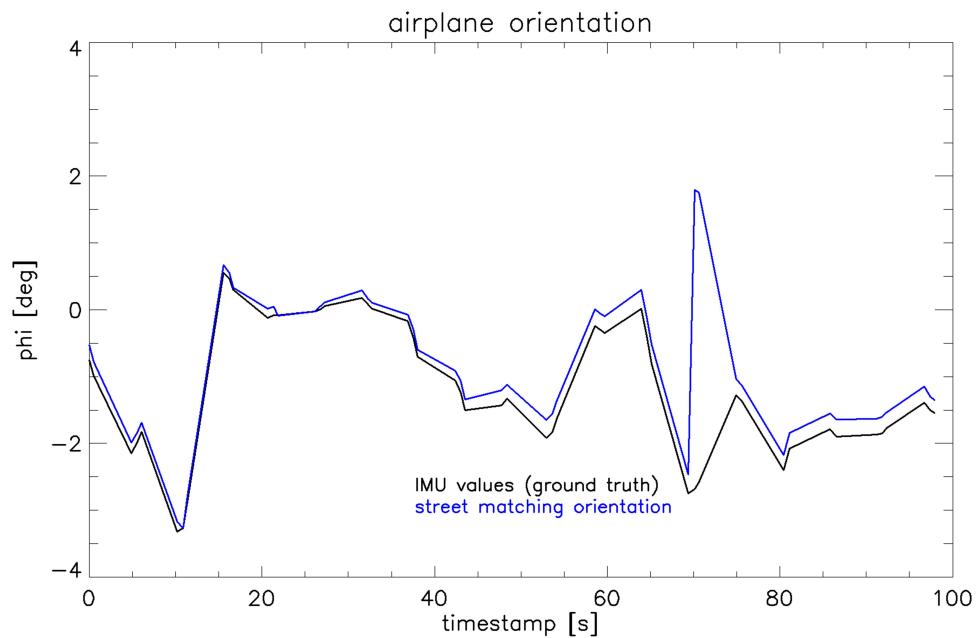


Abbildung 18:  $\phi$ -Winkel der absoluten Orientierung (Überflug 5)

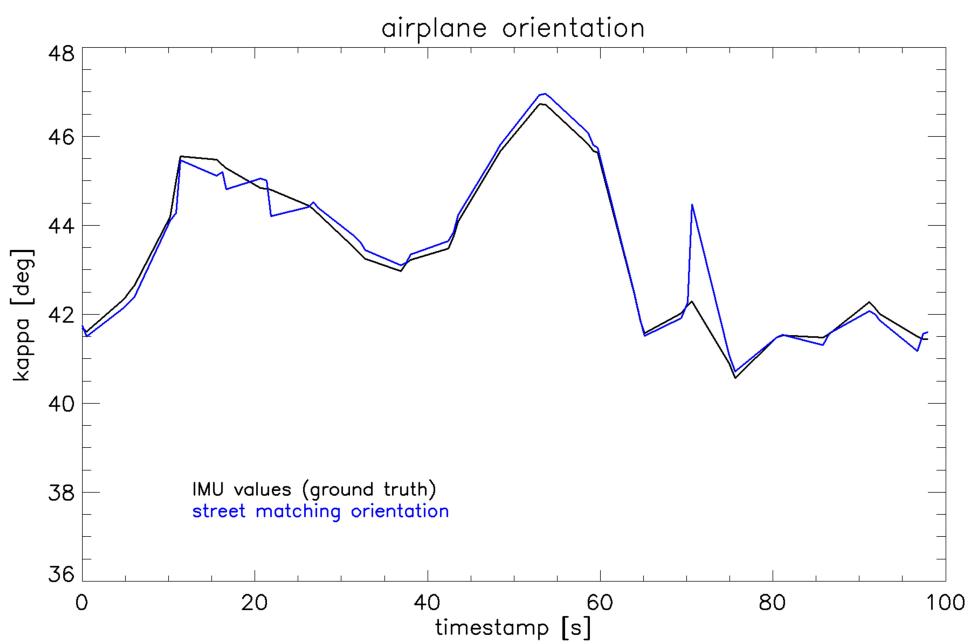


Abbildung 19:  $\kappa$ -Winkel der absoluten Orientierung (Überflug 5)

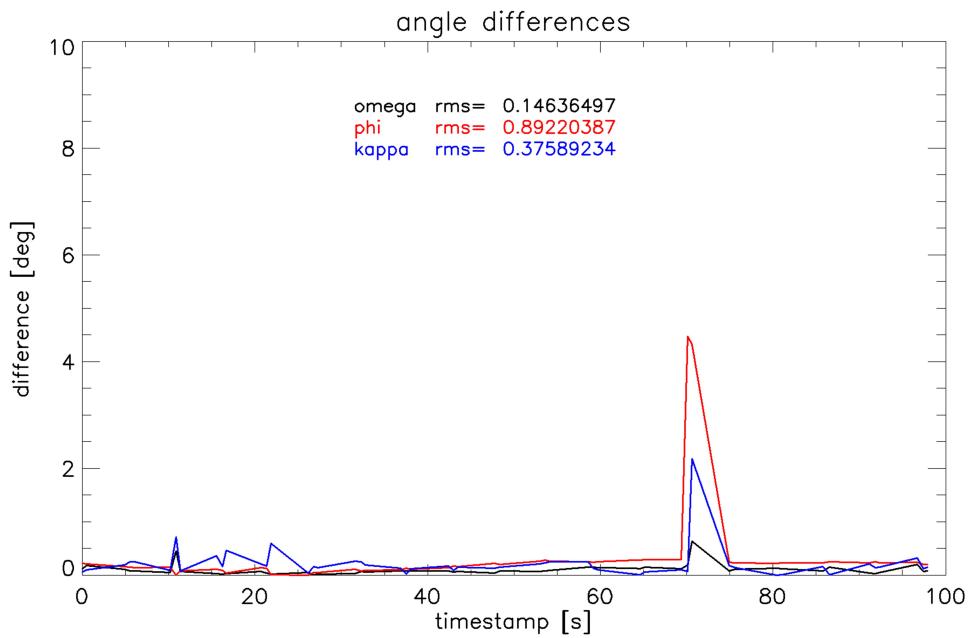


Abbildung 20: Winkelfehler der absoluten Orientierung (Überflug 5)

## Überflug 6

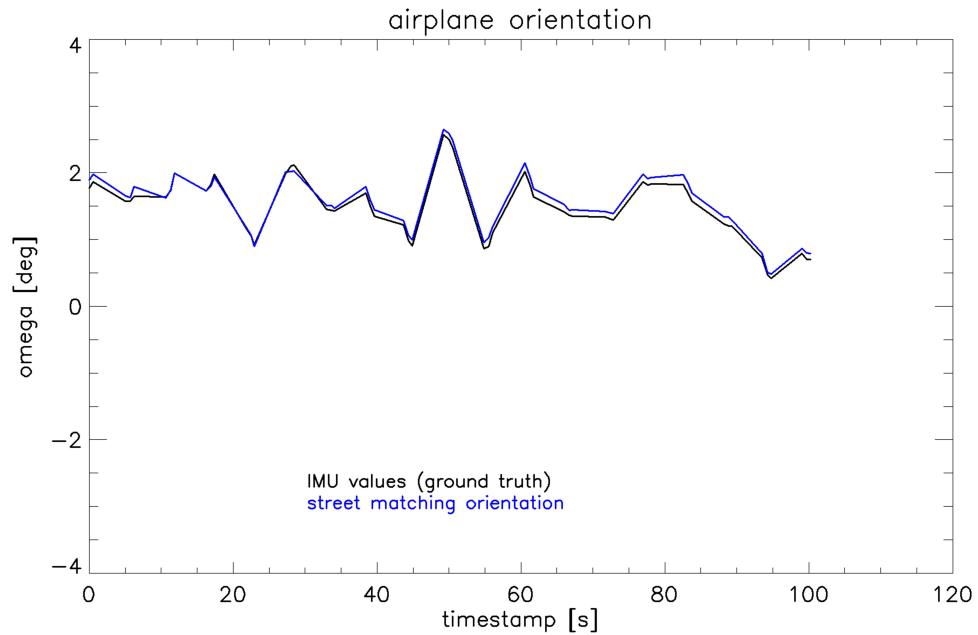


Abbildung 21:  $\omega$ -Winkel der absoluten Orientierung (Überflug 6)

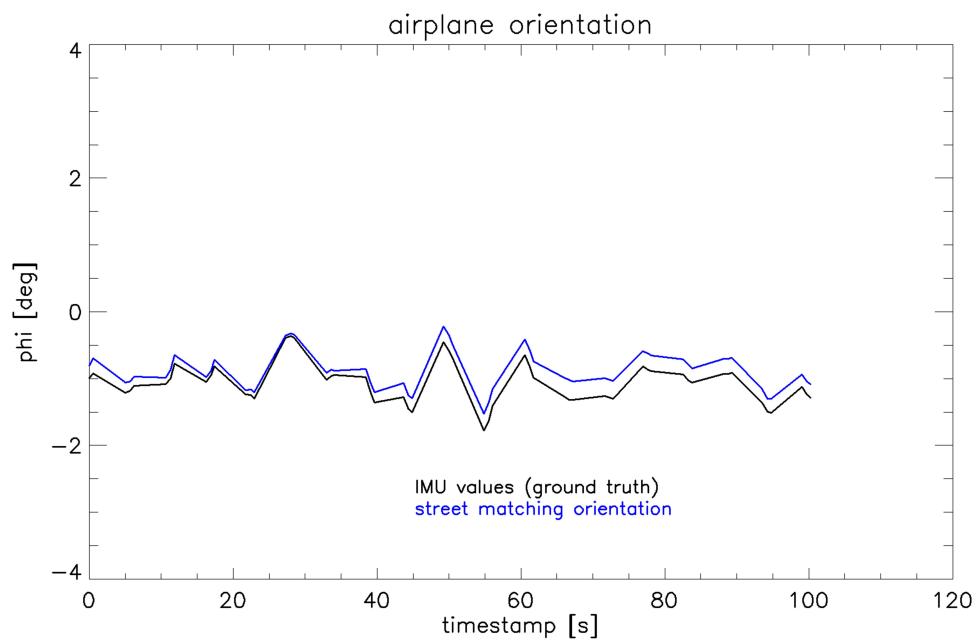


Abbildung 22:  $\phi$ -Winkel der absoluten Orientierung (Überflug 6)

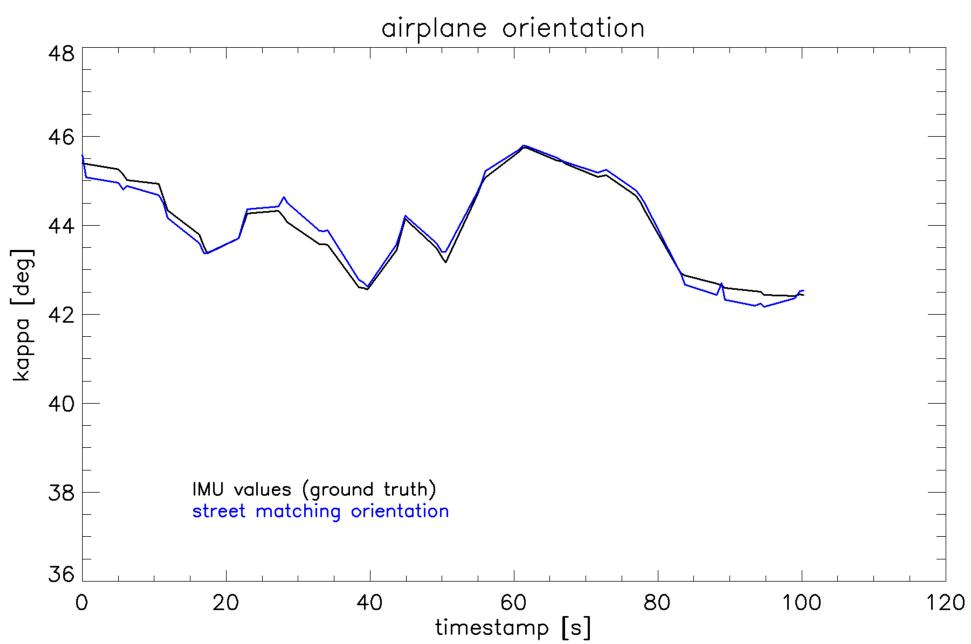


Abbildung 23:  $\kappa$ -Winkel der absoluten Orientierung (Überflug 6)

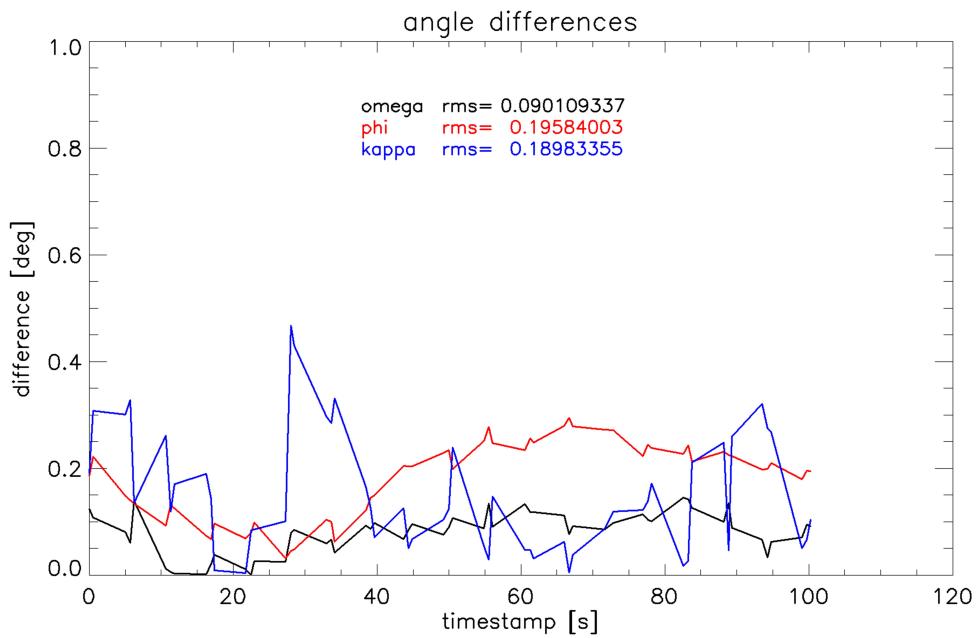


Abbildung 24: Winkelfehler der absoluten Orientierung (Überflug 6)

## 8 Zusammenfassung und Ausblick

### Überflug 7

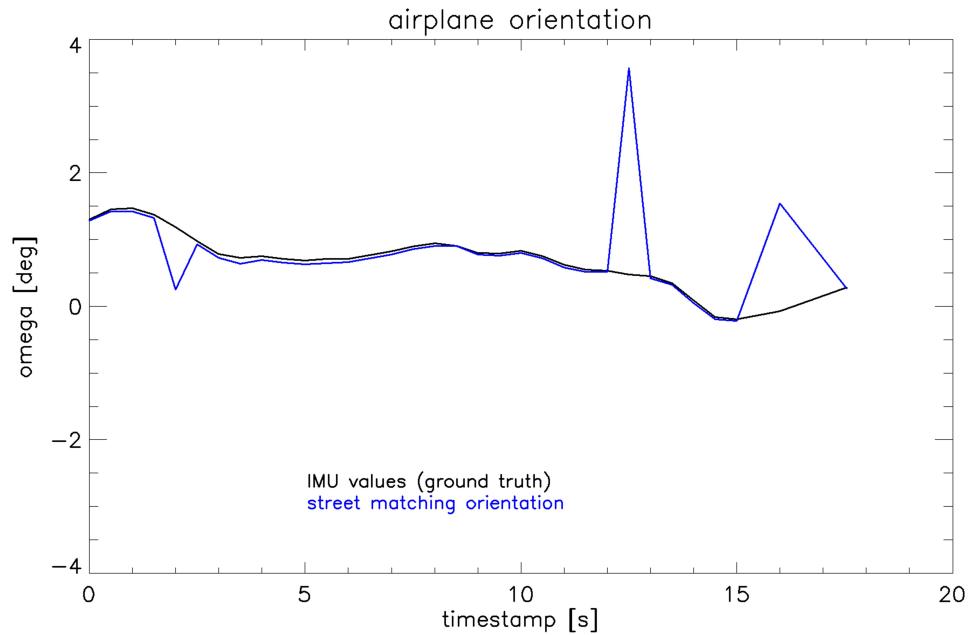


Abbildung 25:  $\omega$ -Winkel der absoluten Orientierung (Überflug 7)

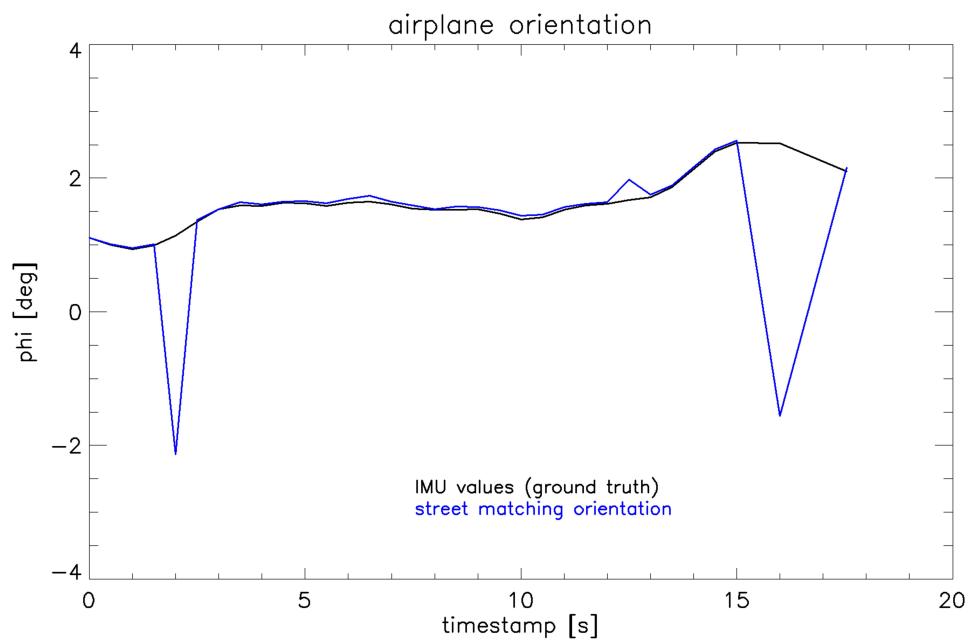


Abbildung 26:  $\phi$ -Winkel der absoluten Orientierung (Überflug 7)

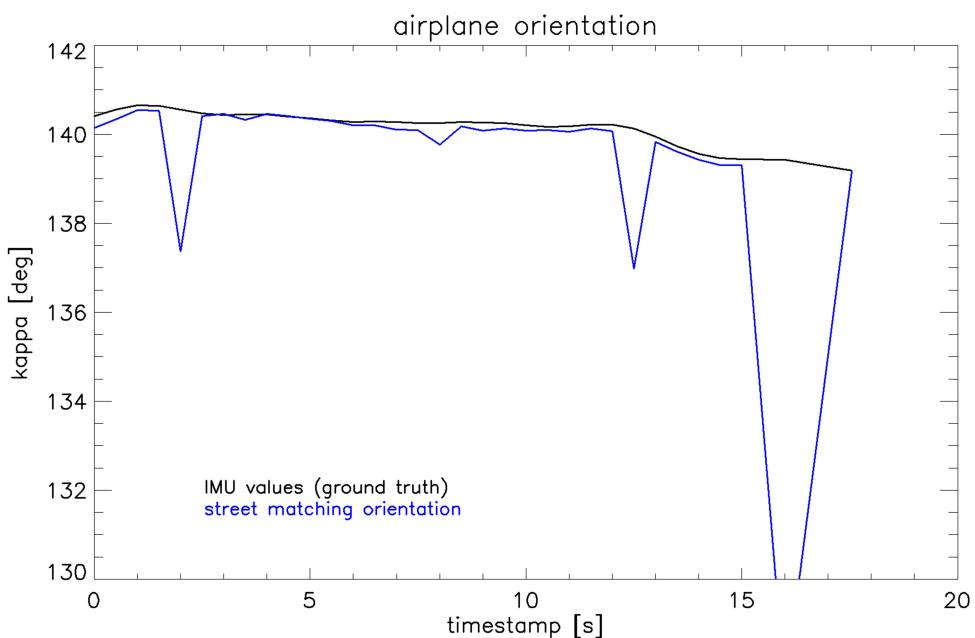


Abbildung 27:  $\kappa$ -Winkel der absoluten Orientierung (Überflug 7)

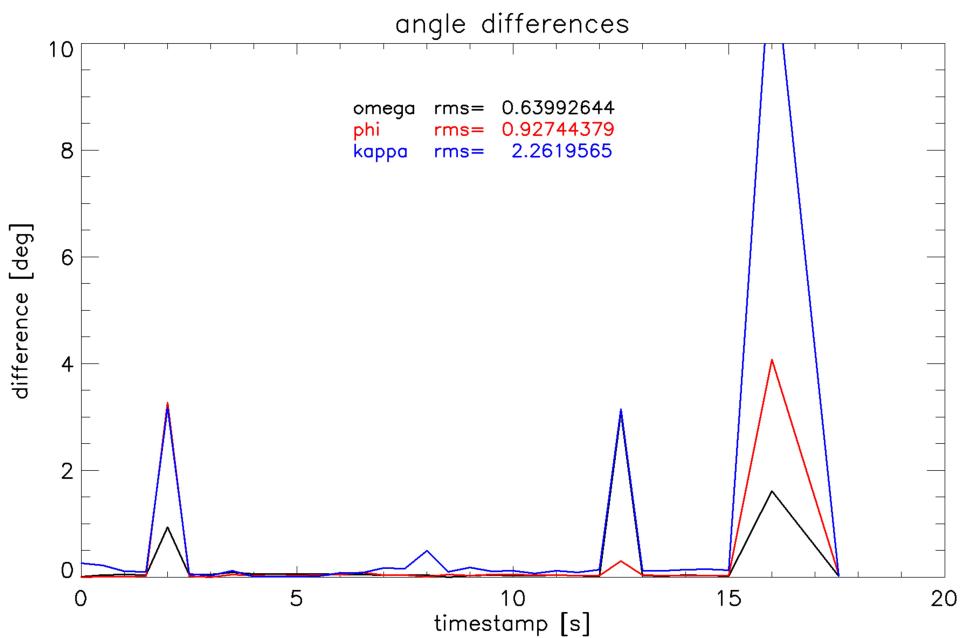


Abbildung 28: Winkelfehler der absoluten Orientierung (Überflug 7)

## Überflug 8

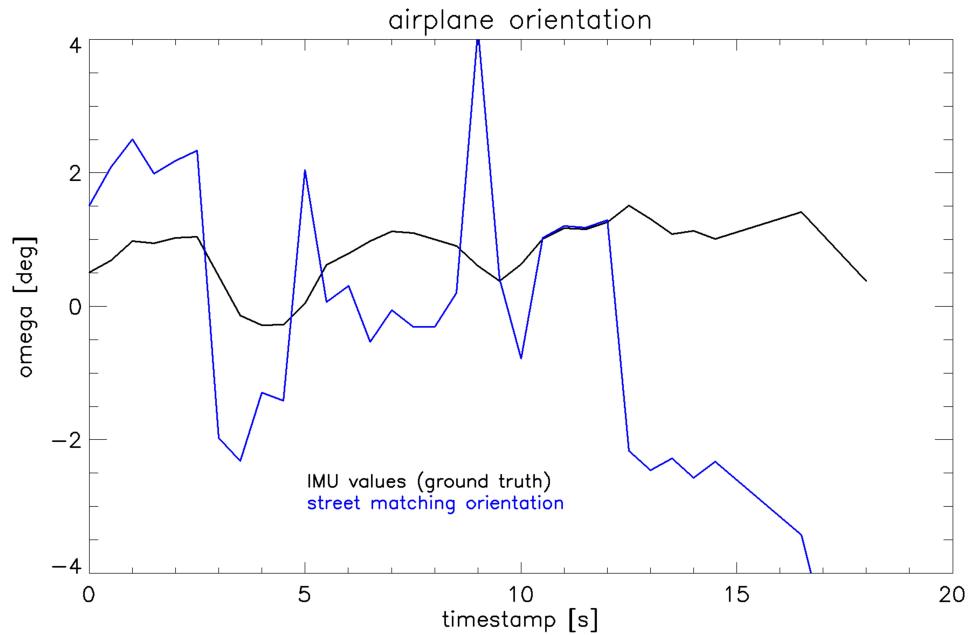


Abbildung 29:  $\omega$ -Winkel der absoluten Orientierung (Überflug 8)

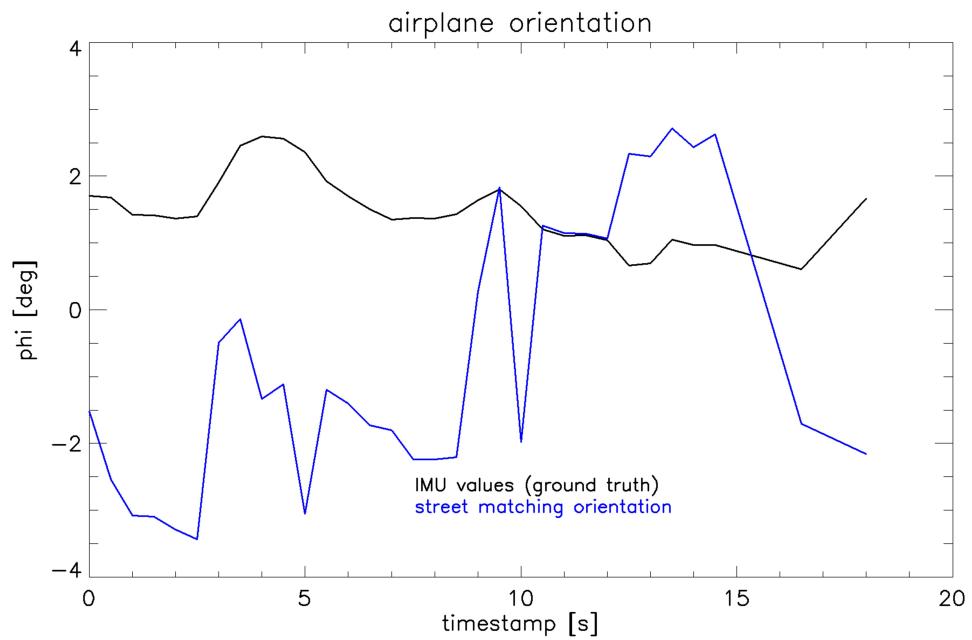


Abbildung 30:  $\phi$ -Winkel der absoluten Orientierung (Überflug 8)

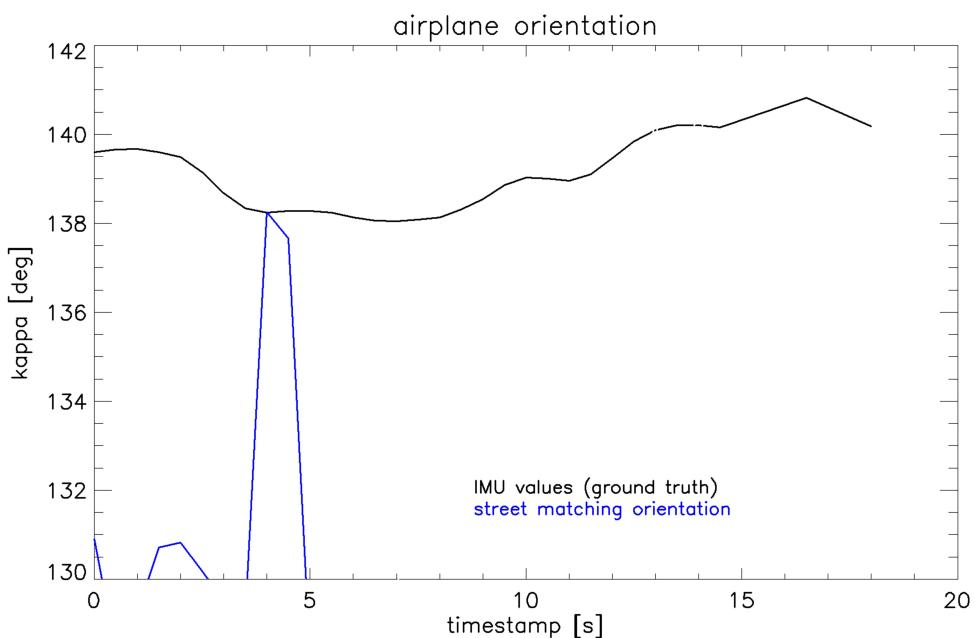


Abbildung 31:  $\kappa$ -Winkel der absoluten Orientierung (Überflug 8)

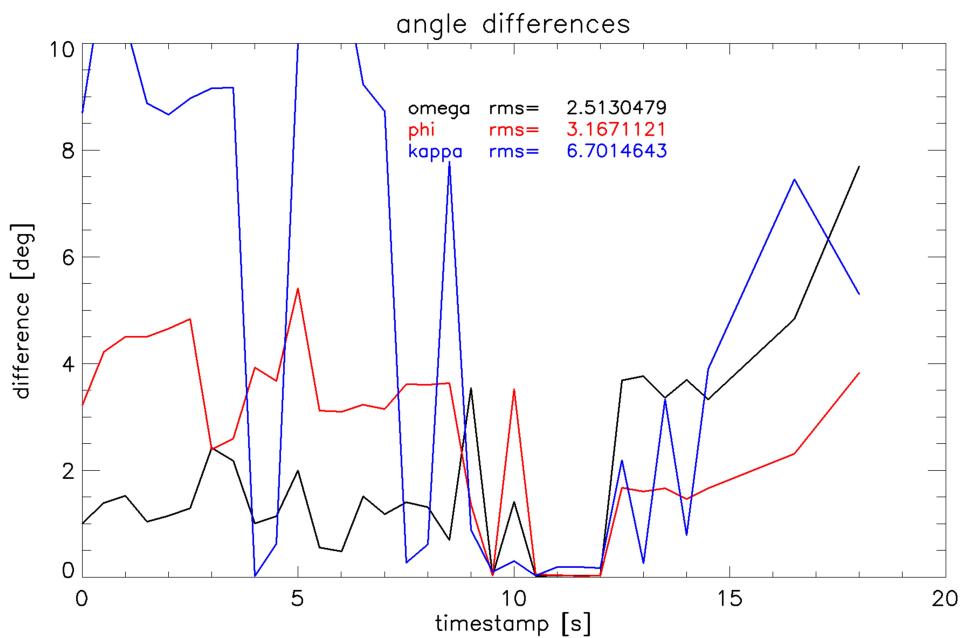


Abbildung 32: Winkelfehler der absoluten Orientierung (Überflug 8)

### Überflug 9

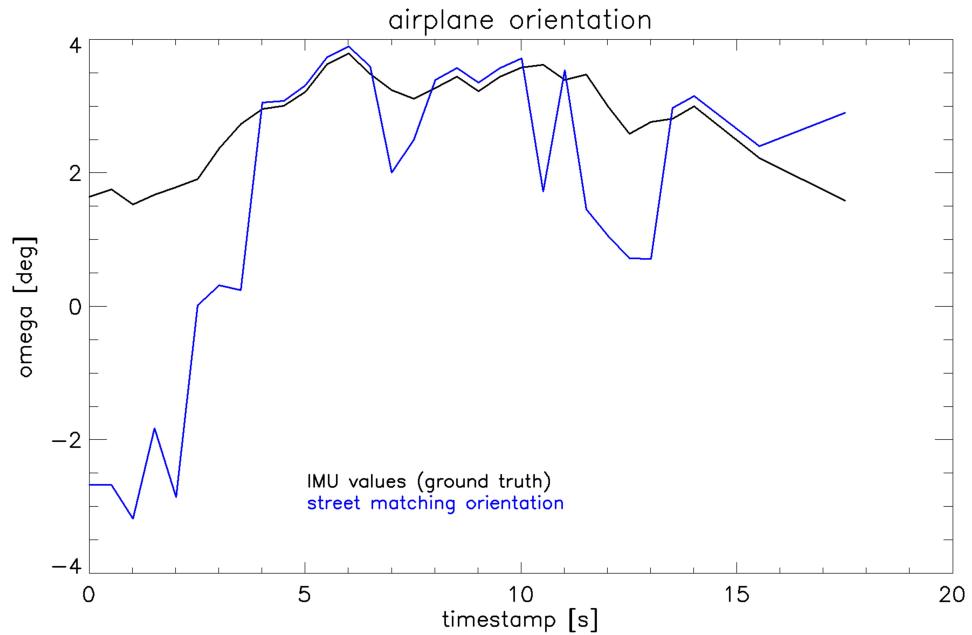


Abbildung 33:  $\omega$ -Winkel der absoluten Orientierung (Überflug 9)

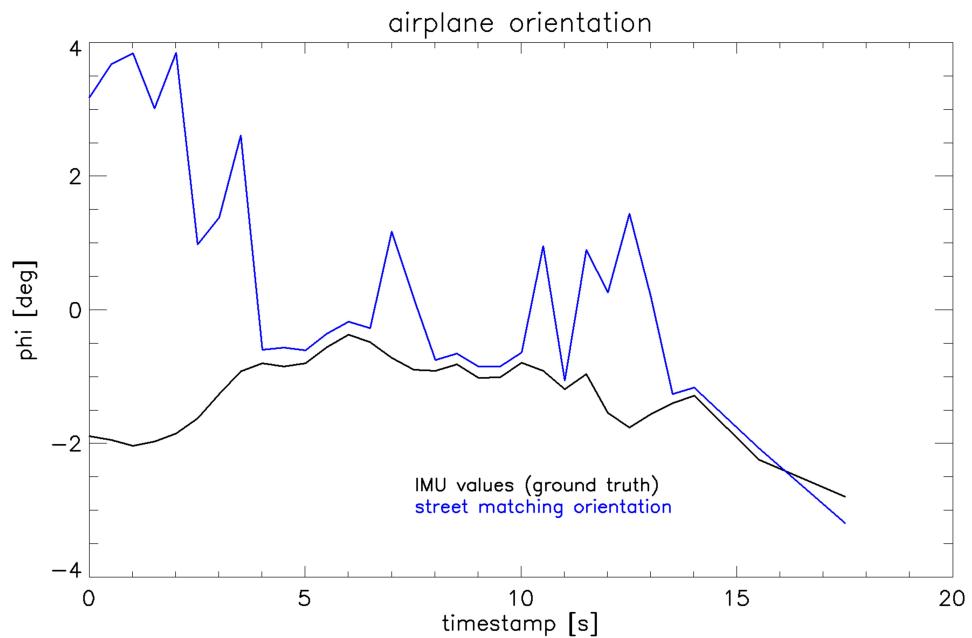


Abbildung 34:  $\phi$ -Winkel der absoluten Orientierung (Überflug 9)

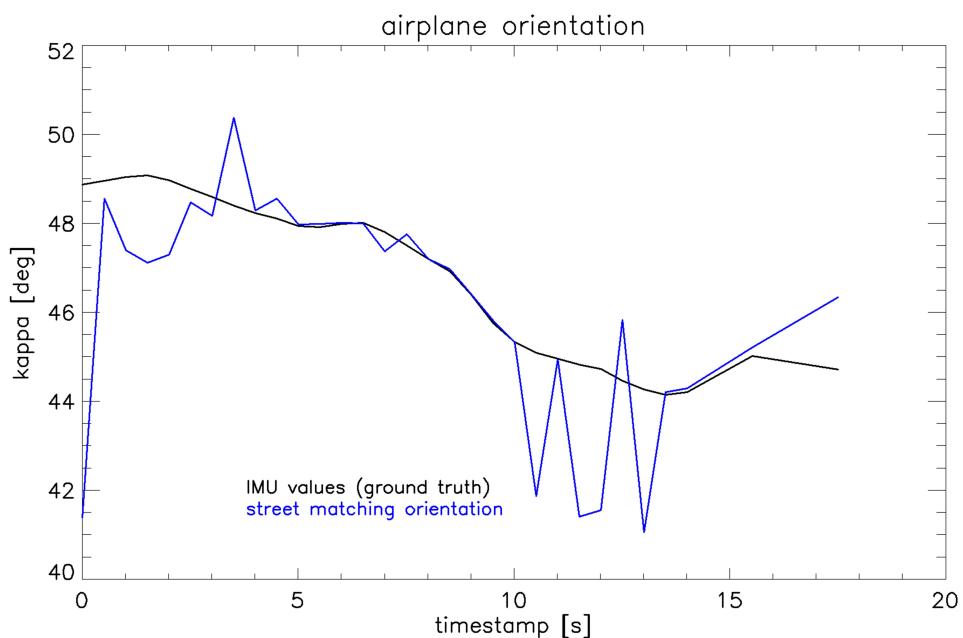


Abbildung 35:  $\kappa$ -Winkel der absoluten Orientierung (Überflug 9)

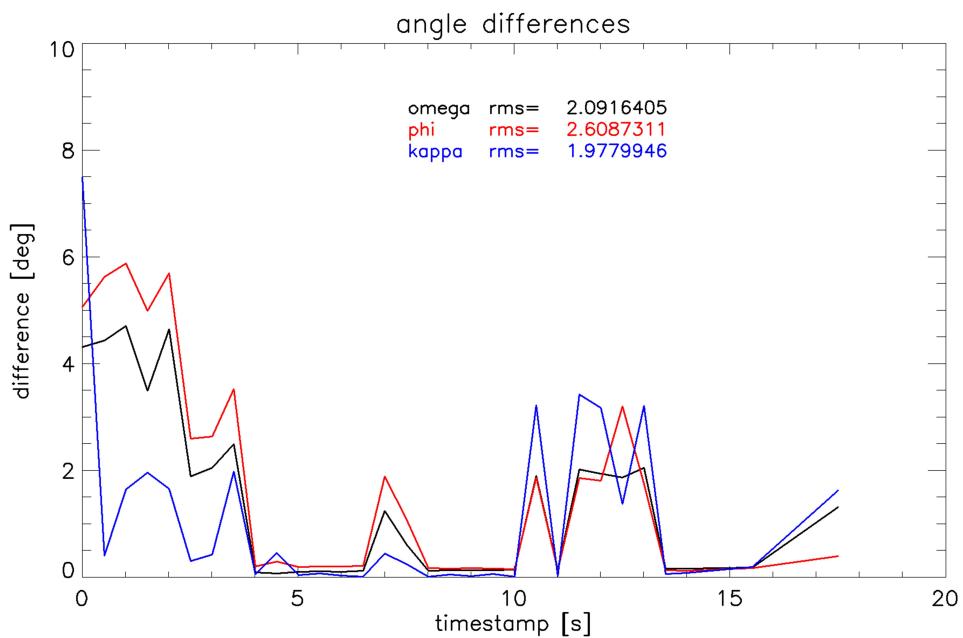


Abbildung 36: Winkelfehler der absoluten Orientierung (Überflug 9)

### Überflug 10

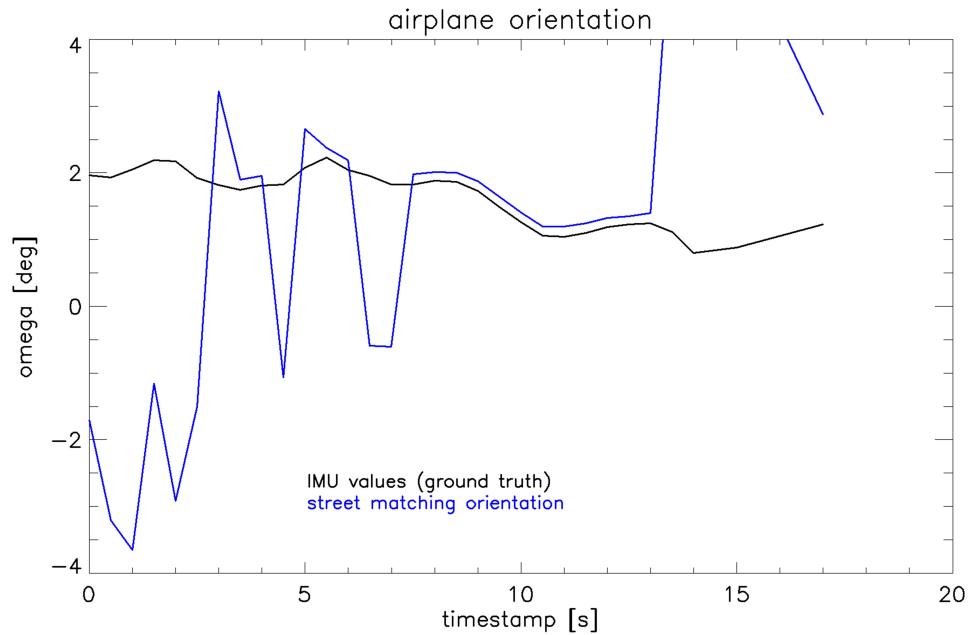


Abbildung 37:  $\omega$ -Winkel der absoluten Orientierung (Überflug 10)

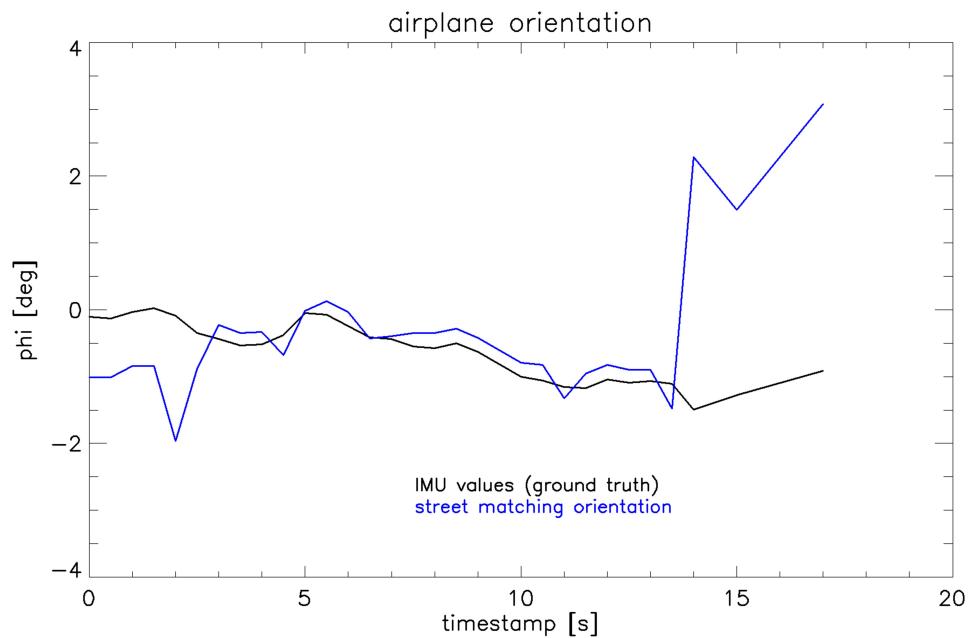


Abbildung 38:  $\phi$ -Winkel der absoluten Orientierung (Überflug 10)

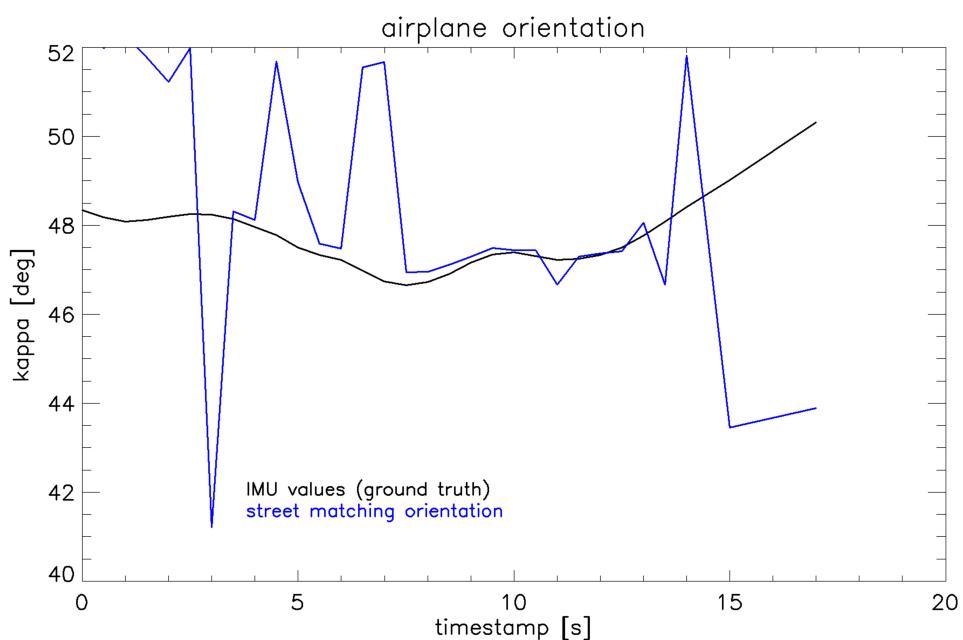


Abbildung 39:  $\kappa$ -Winkel der absoluten Orientierung (Überflug 10)

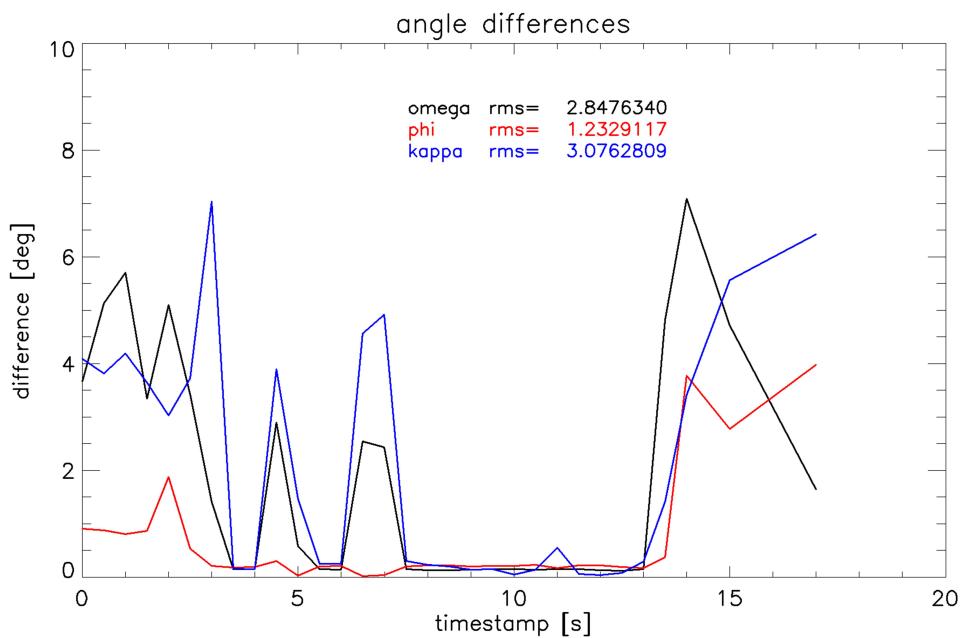


Abbildung 40: Winkelfehler der absoluten Orientierung (Überflug 10)

### Überflug 11

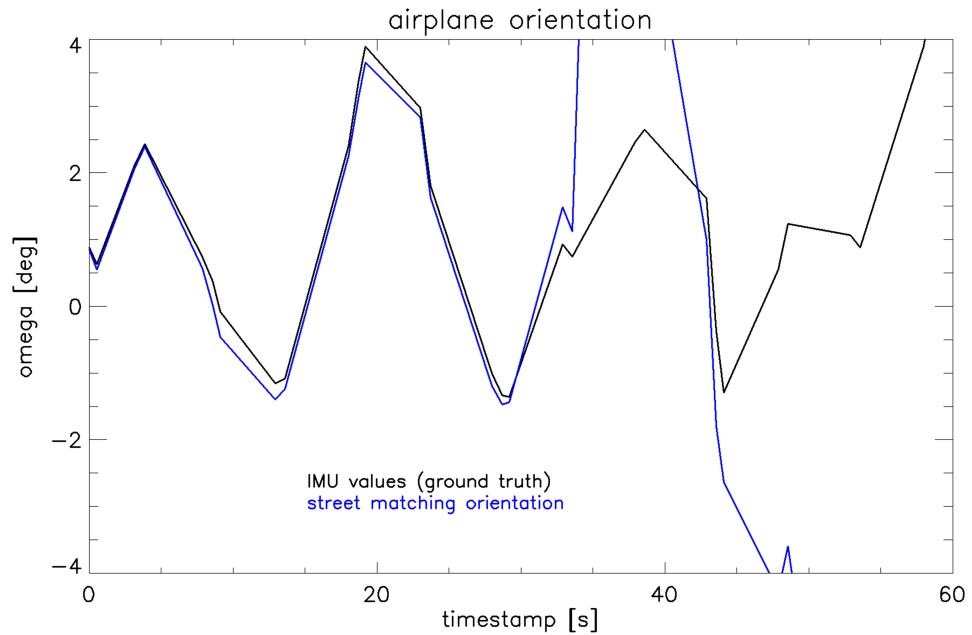


Abbildung 41:  $\omega$ -Winkel der absoluten Orientierung (Überflug 11)

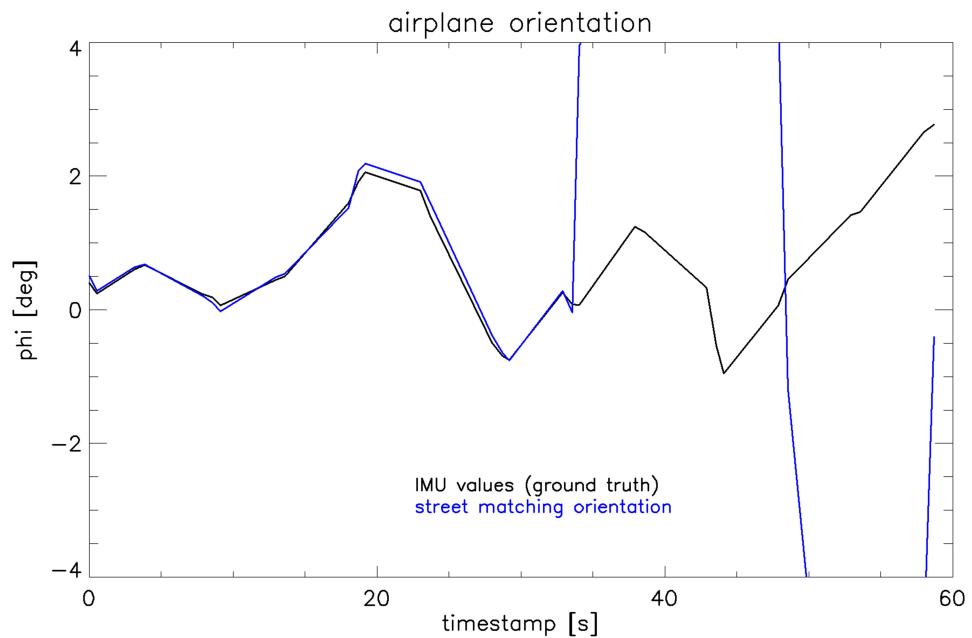


Abbildung 42:  $\phi$ -Winkel der absoluten Orientierung (Überflug 11)

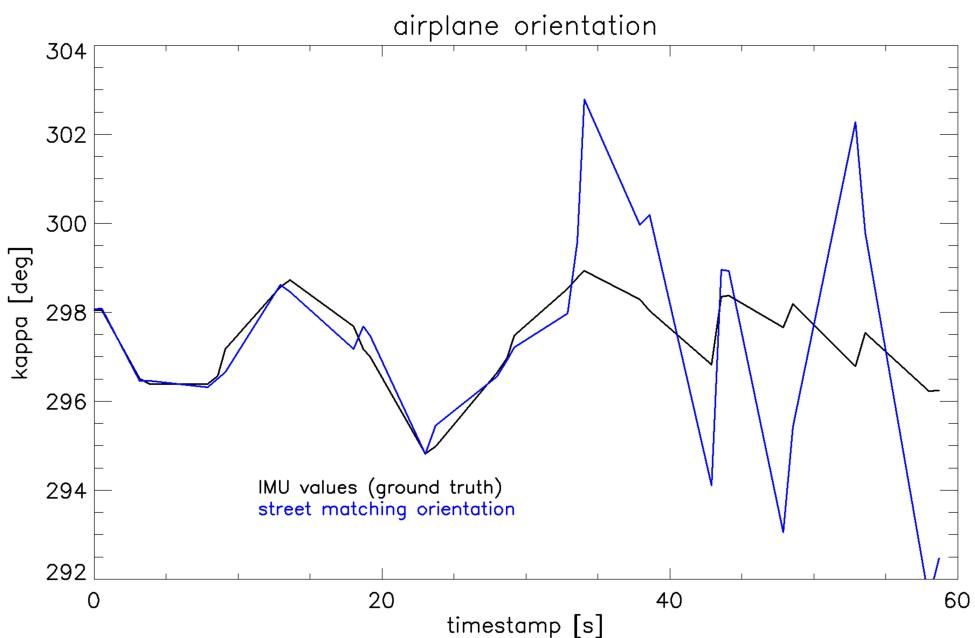


Abbildung 43:  $\kappa$ -Winkel der absoluten Orientierung (Überflug 11)

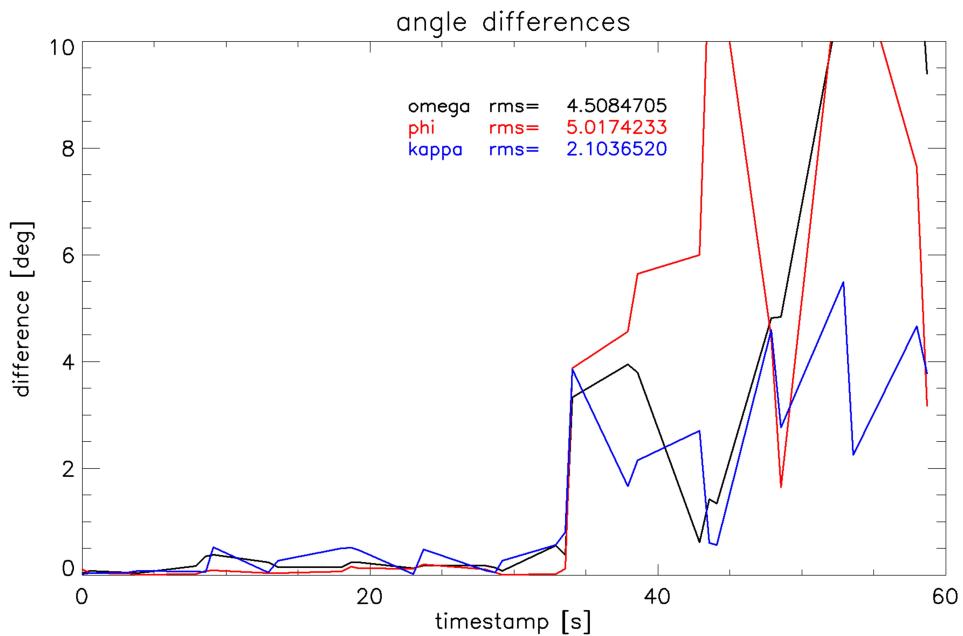


Abbildung 44: Winkelfehler der absoluten Orientierung (Überflug 11)

## **Ergebnisse der relativen optischen Orientierung**

## Überflug 1

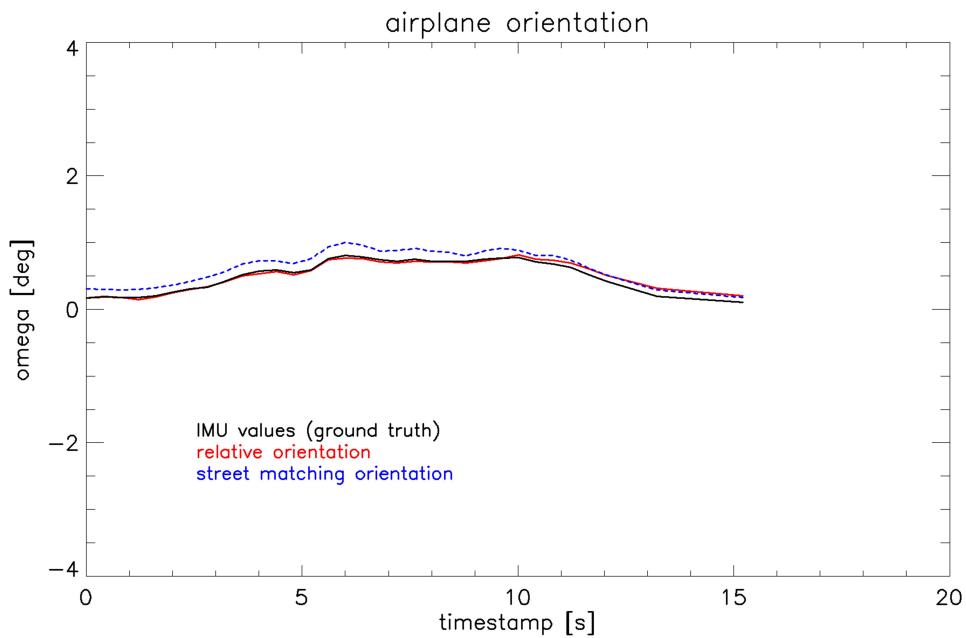


Abbildung 45:  $\omega$ -Winkel der relativen Orientierung (Überflug 1)

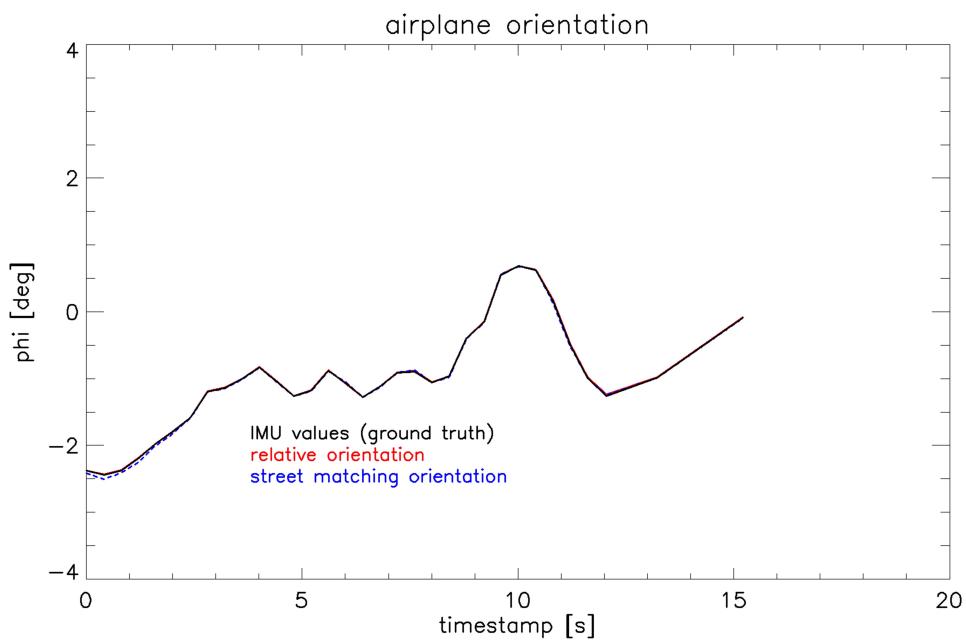


Abbildung 46:  $\phi$ -Winkel der relativen Orientierung (Überflug 1)

## 8 Zusammenfassung und Ausblick

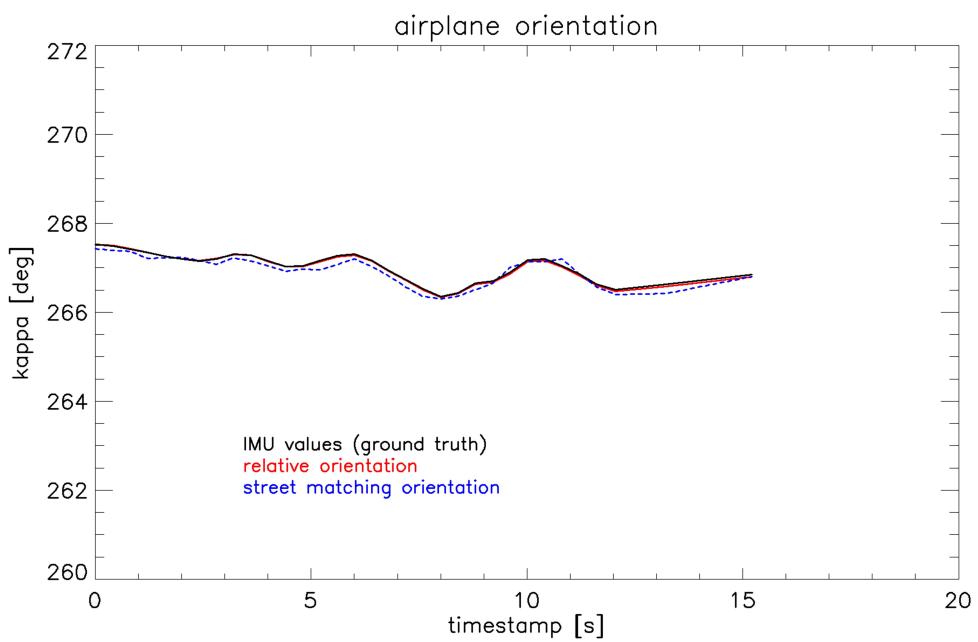


Abbildung 47:  $\kappa$ -Winkel der relativen Orientierung (Überflug 1)

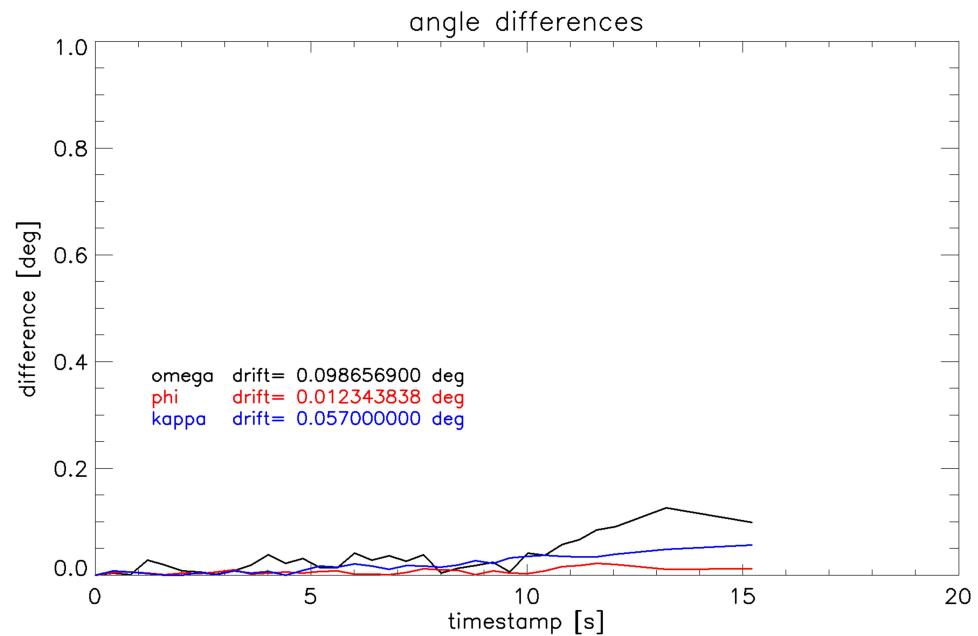


Abbildung 48: Winkelfehler der relativen Orientierung (Überflug 1)

## Überflug 2

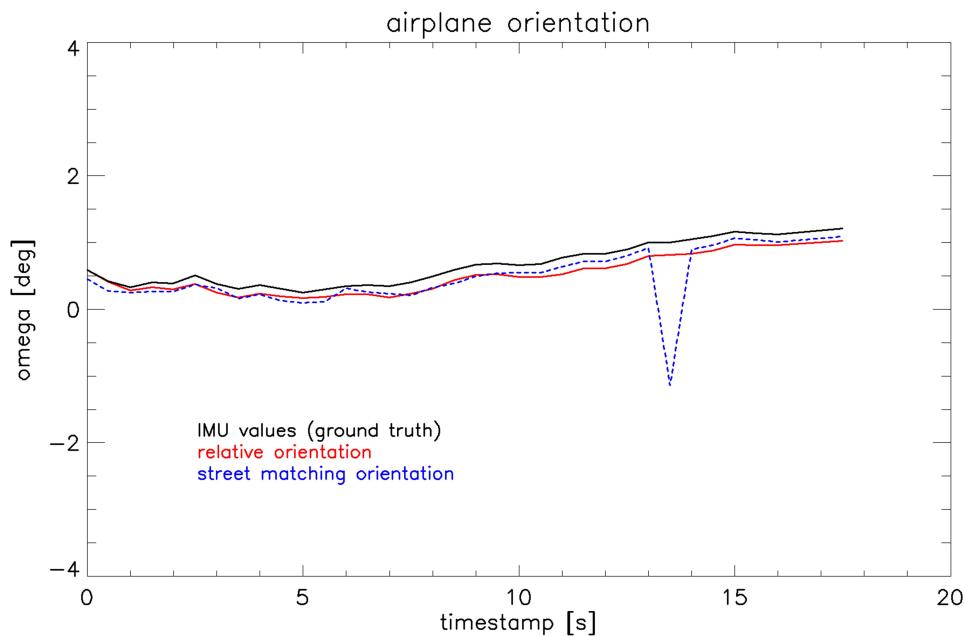


Abbildung 49:  $\omega$ -Winkel der relativen Orientierung (Überflug 2)

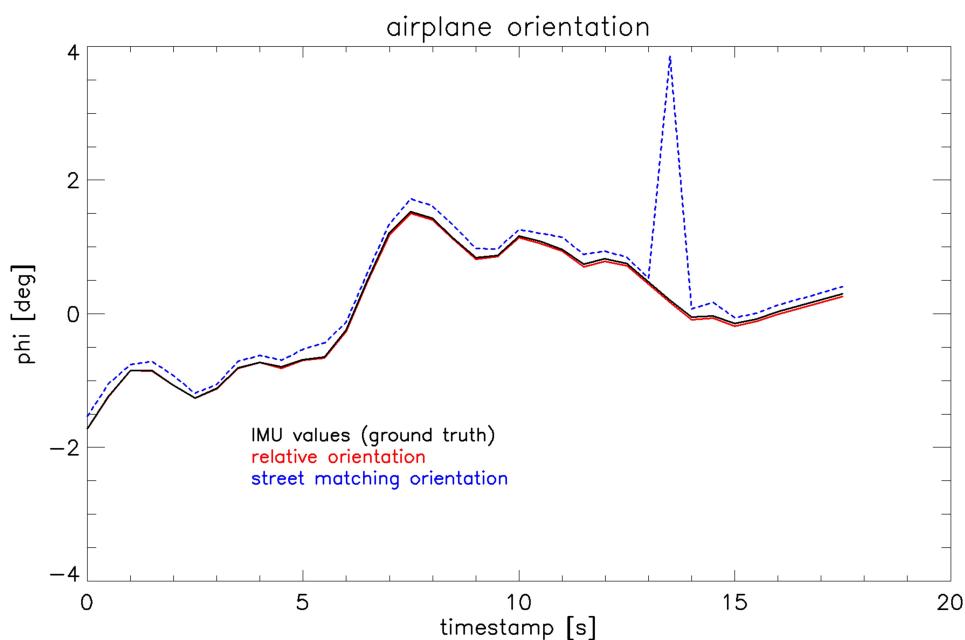


Abbildung 50:  $\phi$ -Winkel der relativen Orientierung (Überflug 2)

## 8 Zusammenfassung und Ausblick

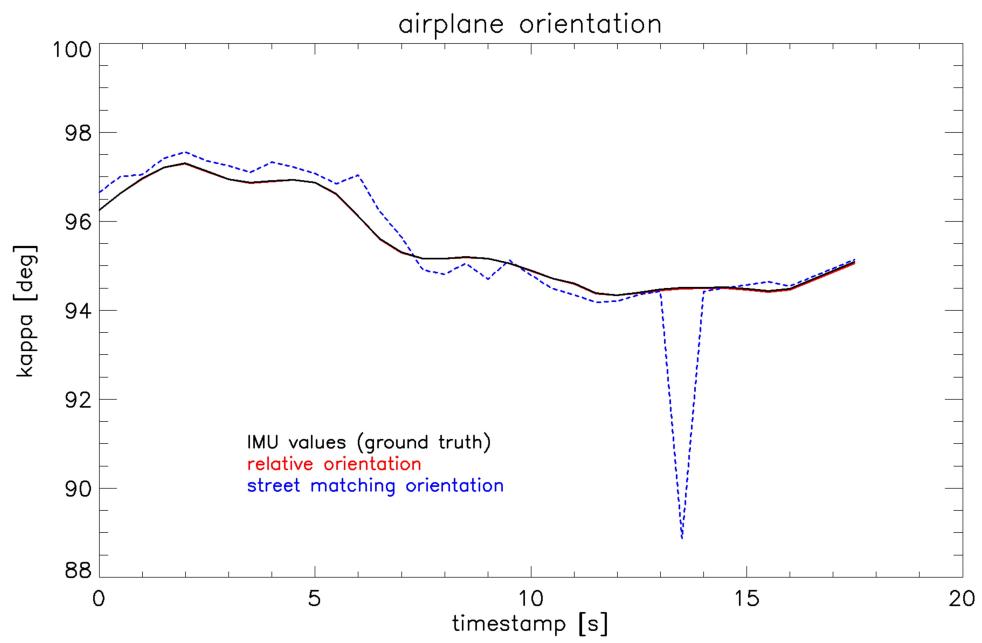


Abbildung 51:  $\kappa$ -Winkel der relativen Orientierung (Überflug 2)

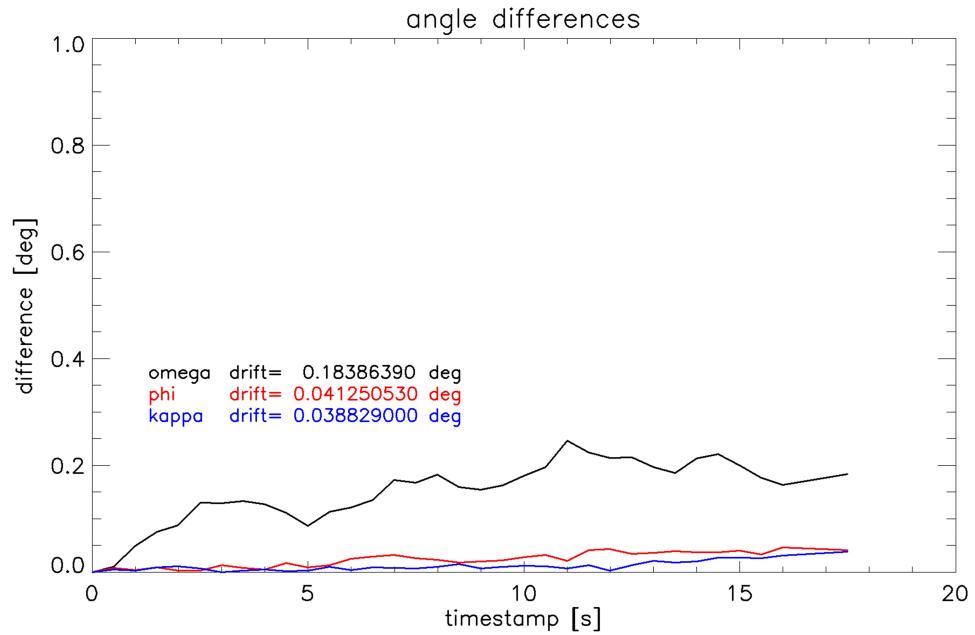


Abbildung 52: Winkelfehler der relativen Orientierung (Überflug 2)

### Überflug 3

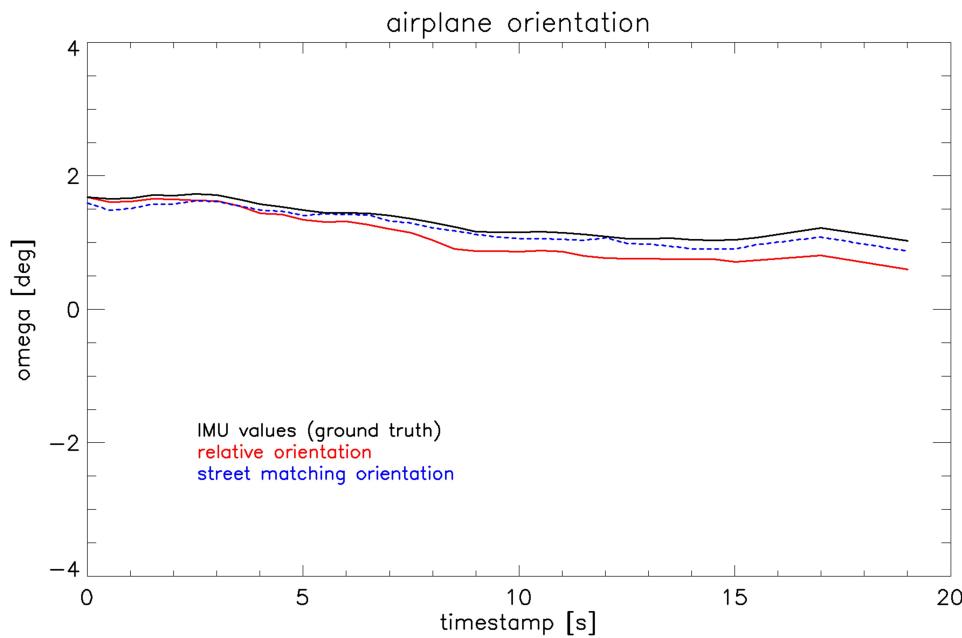


Abbildung 53:  $\omega$ -Winkel der relativen Orientierung (Überflug 3)

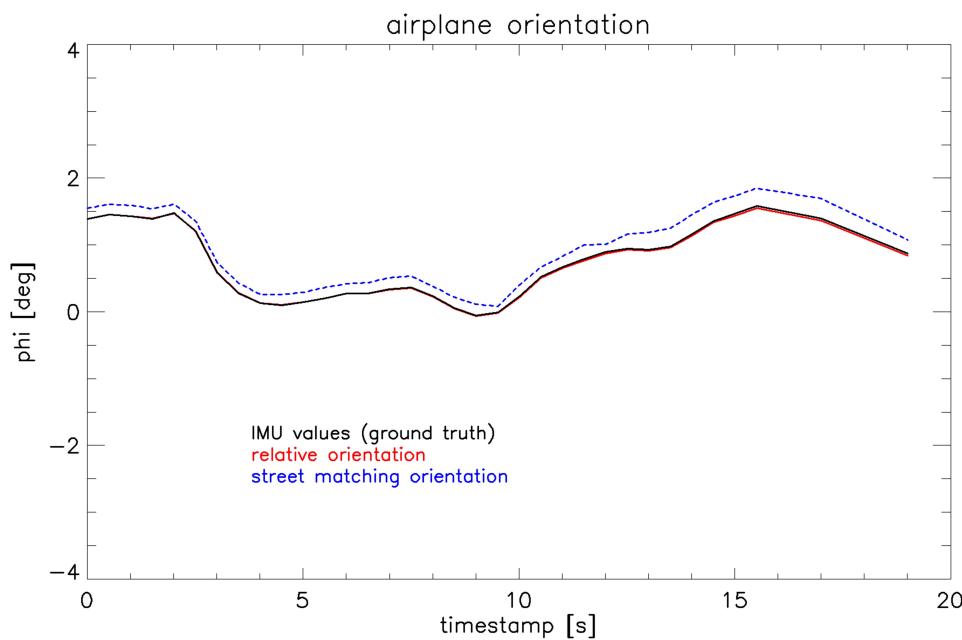


Abbildung 54:  $\phi$ -Winkel der relativen Orientierung (Überflug 3)

## 8 Zusammenfassung und Ausblick

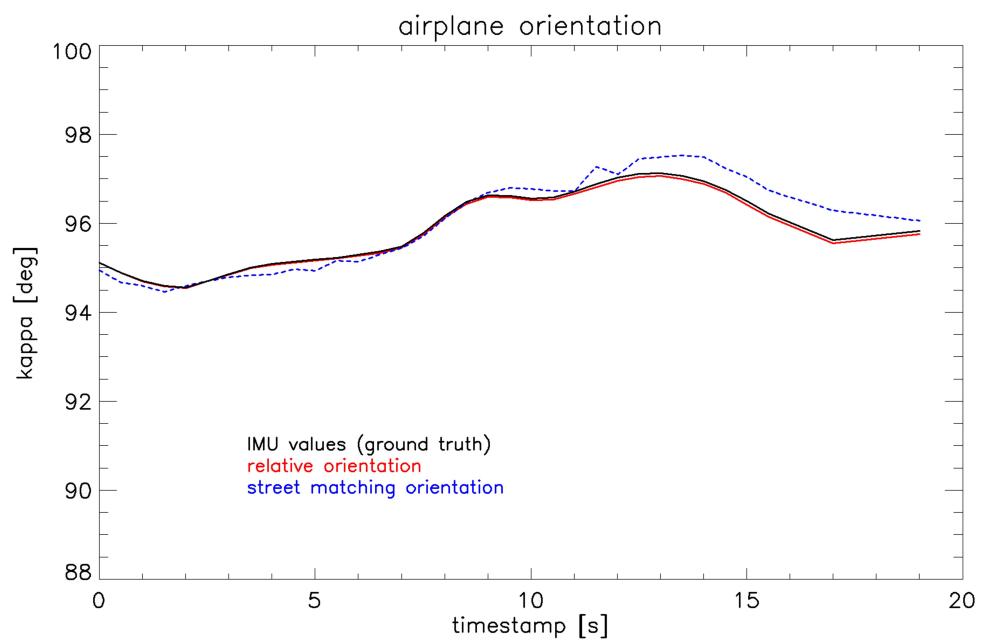


Abbildung 55:  $\kappa$ -Winkel der relativen Orientierung (Überflug 3)

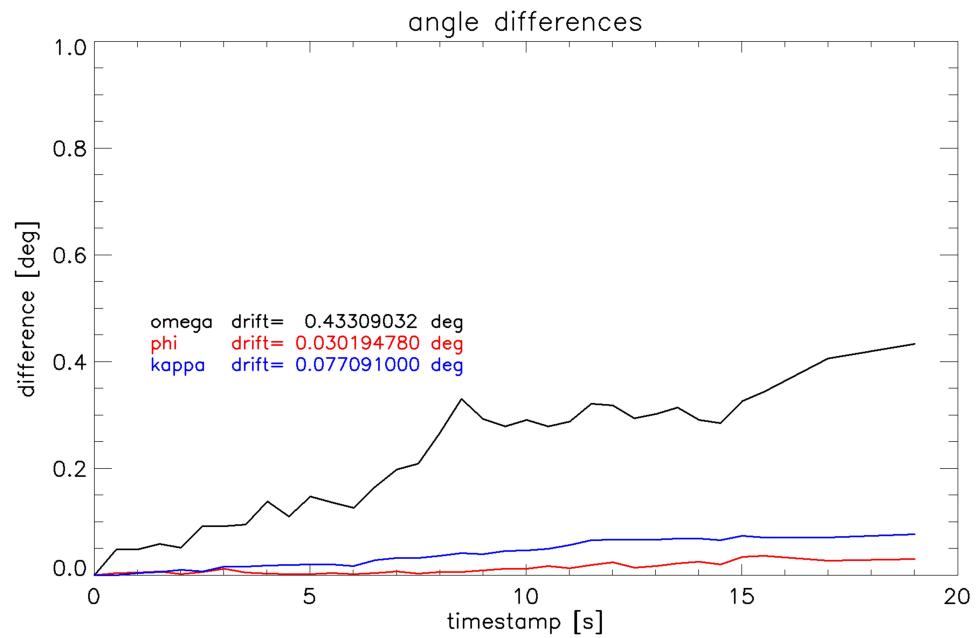


Abbildung 56: Winkelfehler der relativen Orientierung (Überflug 3)

## Überflug 4

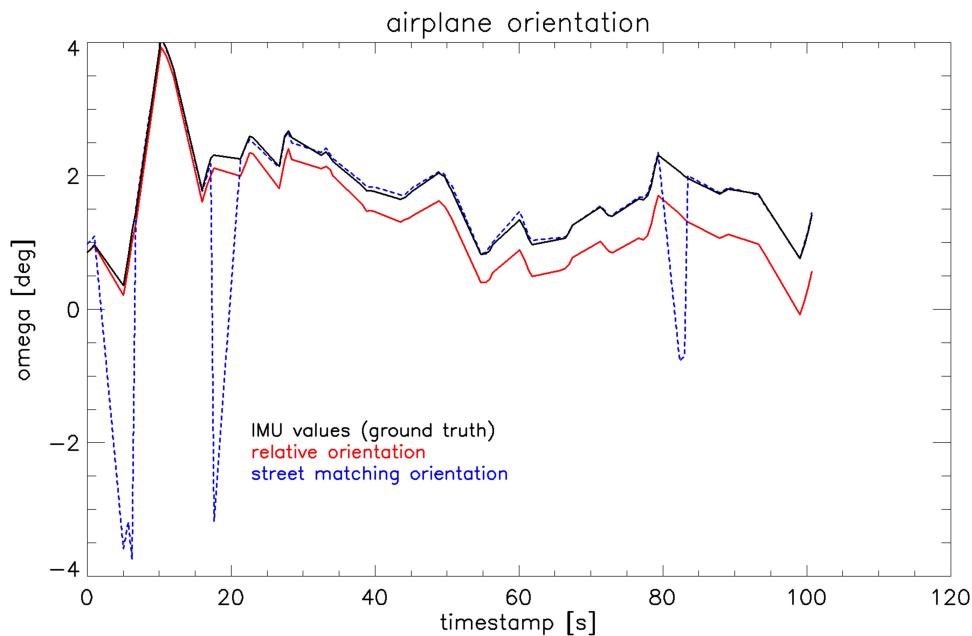


Abbildung 57:  $\omega$ -Winkel der relativen Orientierung (Überflug 4)

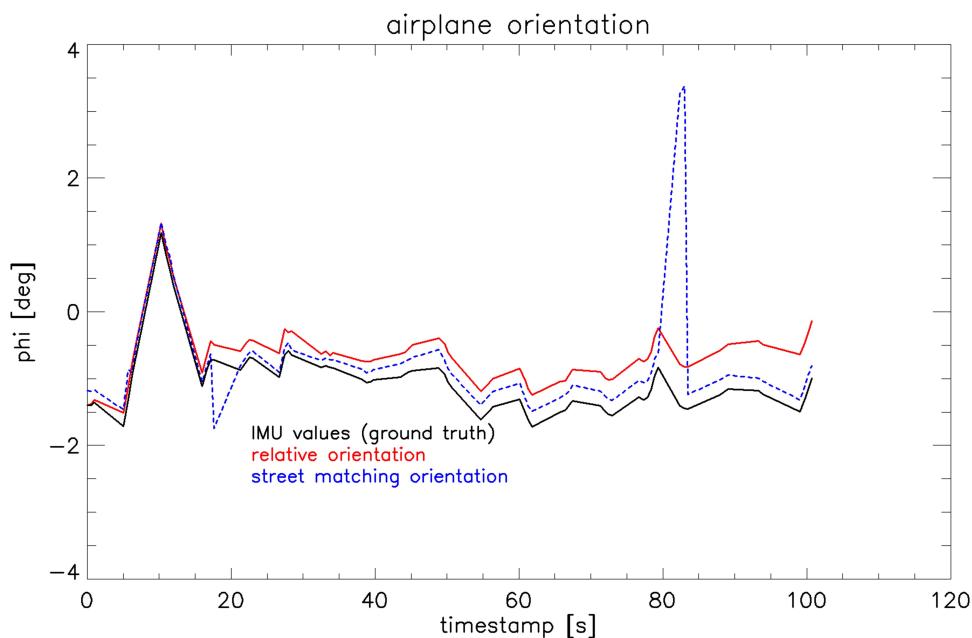


Abbildung 58:  $\phi$ -Winkel der relativen Orientierung (Überflug 4)

## 8 Zusammenfassung und Ausblick

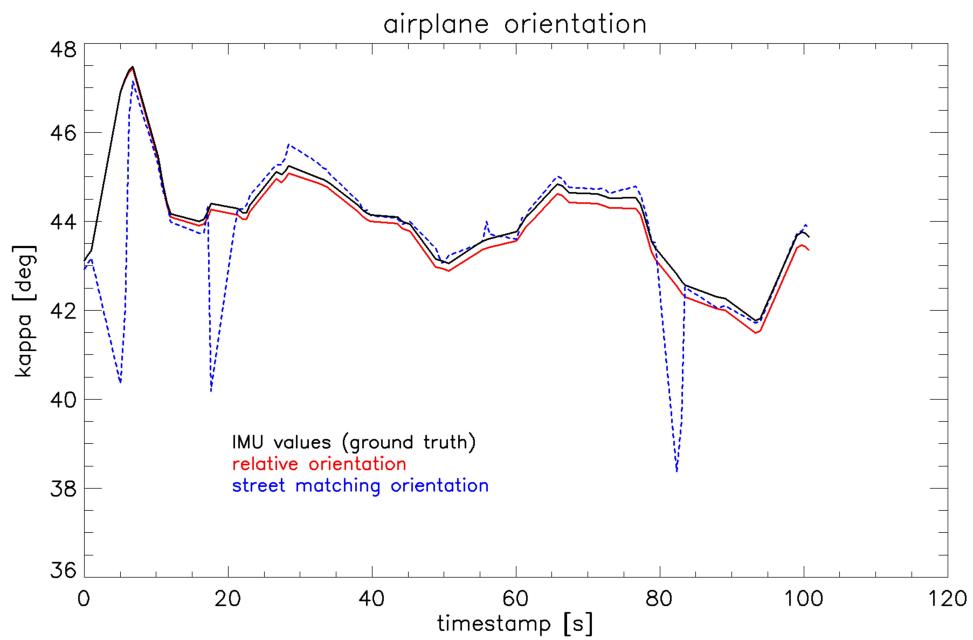


Abbildung 59:  $\kappa$ -Winkel der relativen Orientierung (Überflug 4)

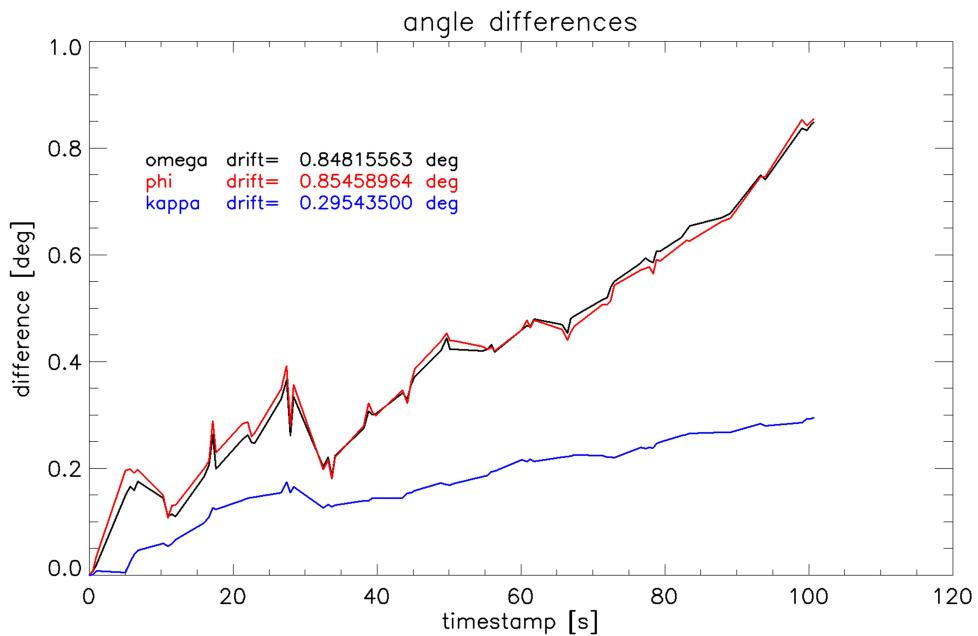


Abbildung 60: Winkelfehler der relativen Orientierung (Überflug 4)

## Überflug 5

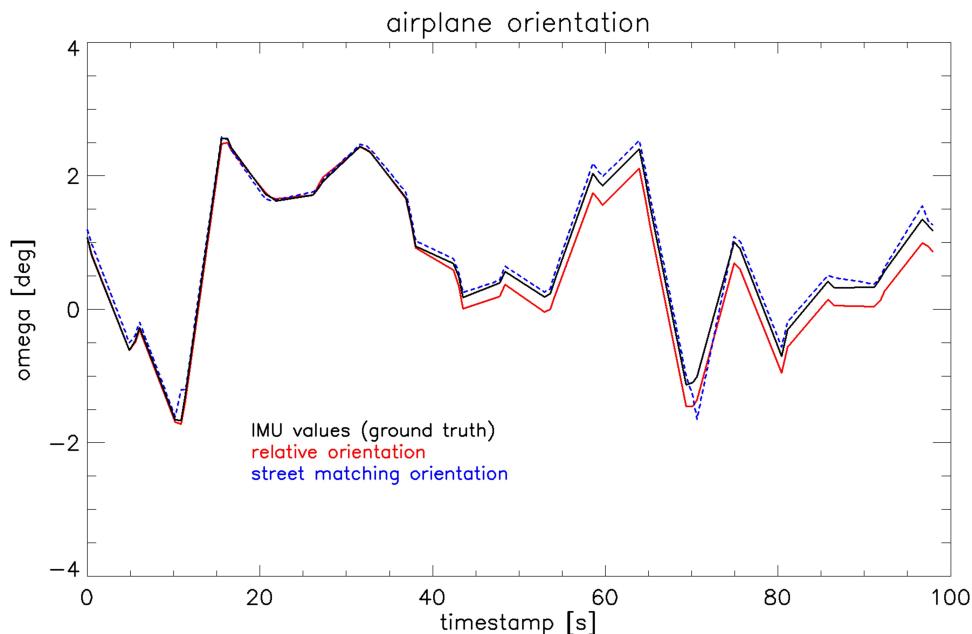


Abbildung 61:  $\omega$ -Winkel der relativen Orientierung (Überflug 5)

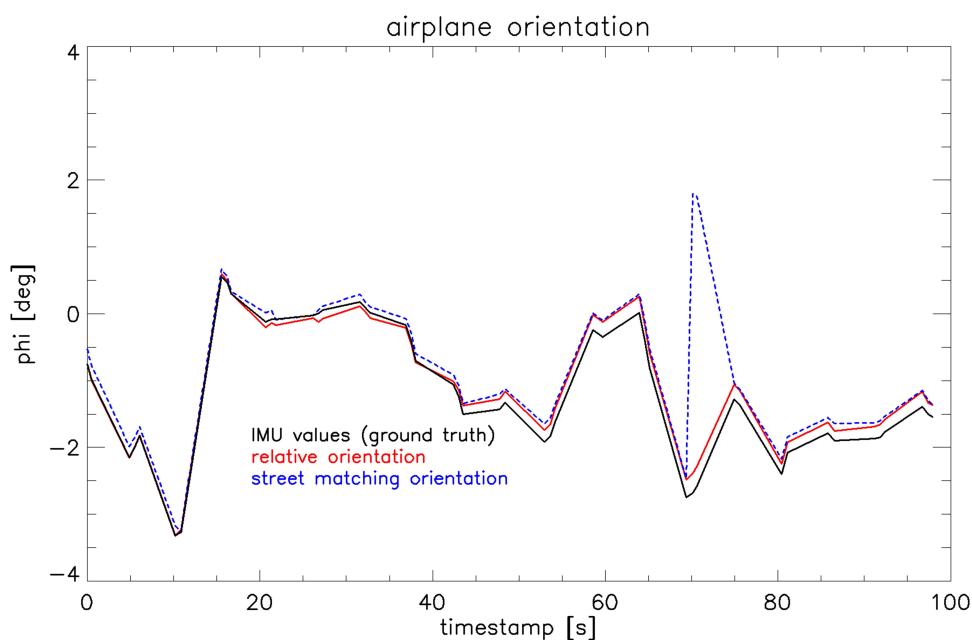


Abbildung 62:  $\phi$ -Winkel der relativen Orientierung (Überflug 5)

## 8 Zusammenfassung und Ausblick

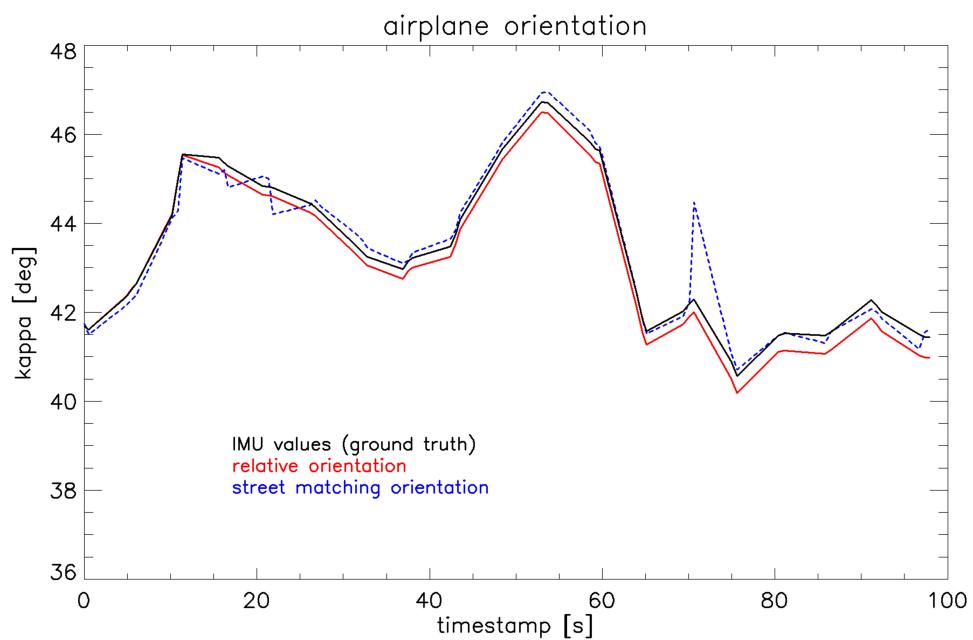


Abbildung 63:  $\kappa$ -Winkel der relativen Orientierung (Überflug 5)

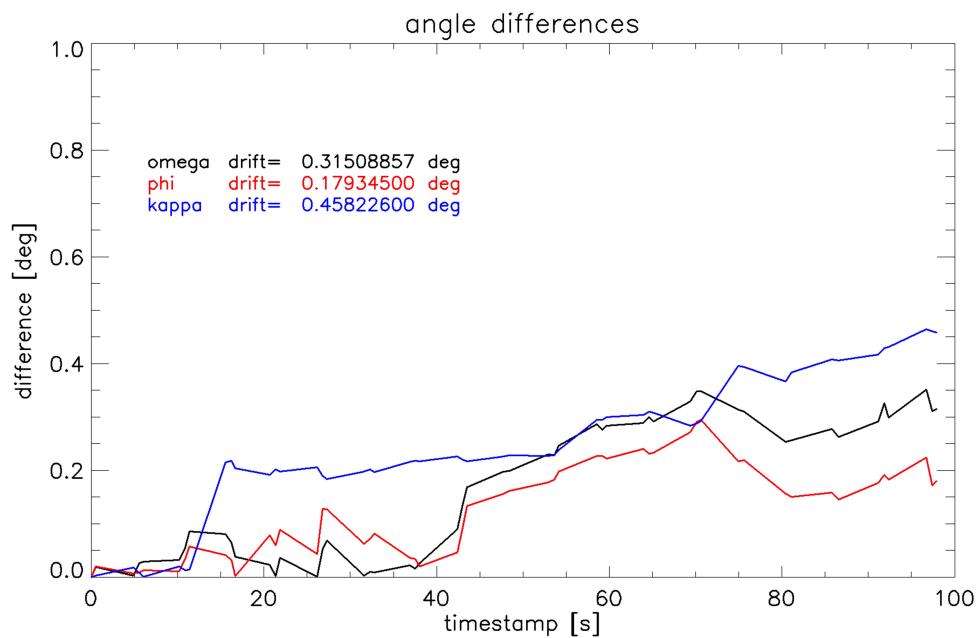


Abbildung 64: Winkelfehler der relativen Orientierung (Überflug 5)

## Überflug 6

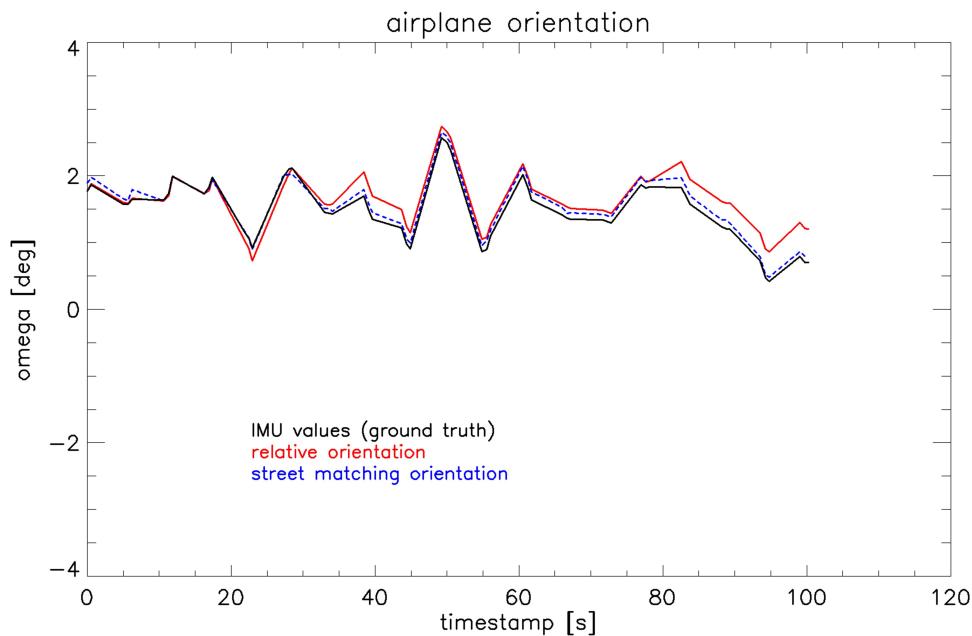


Abbildung 65:  $\omega$ -Winkel der relativen Orientierung (Überflug 6)

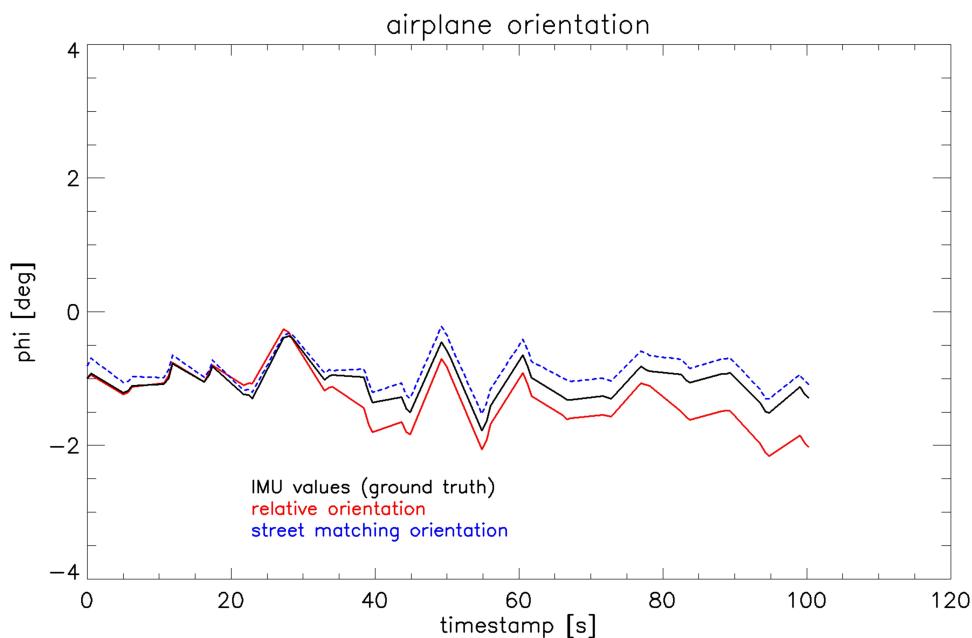


Abbildung 66:  $\phi$ -Winkel der relativen Orientierung (Überflug 6)

## 8 Zusammenfassung und Ausblick

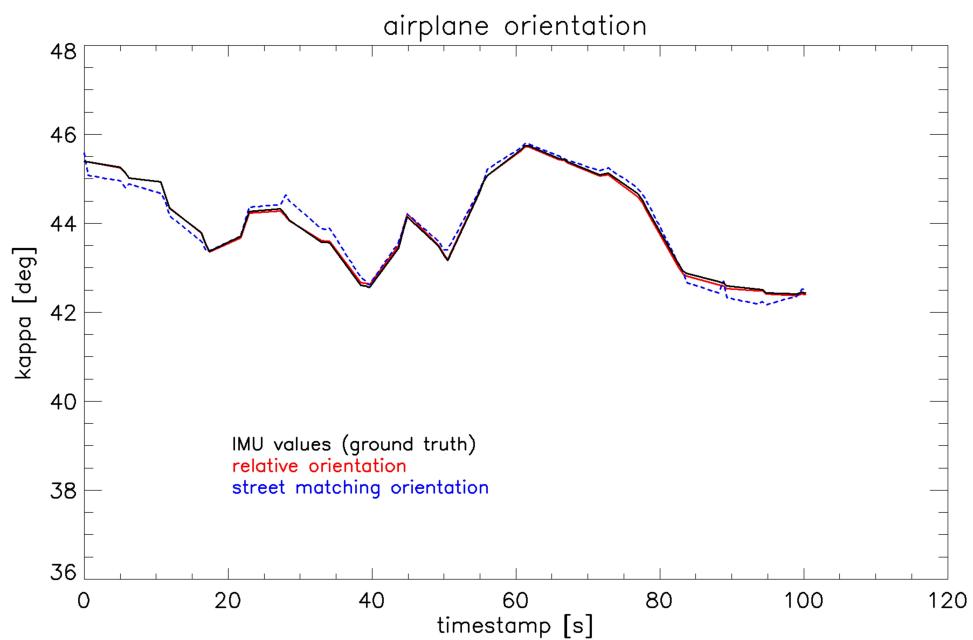


Abbildung 67:  $\kappa$ -Winkel der relativen Orientierung (Überflug 6)

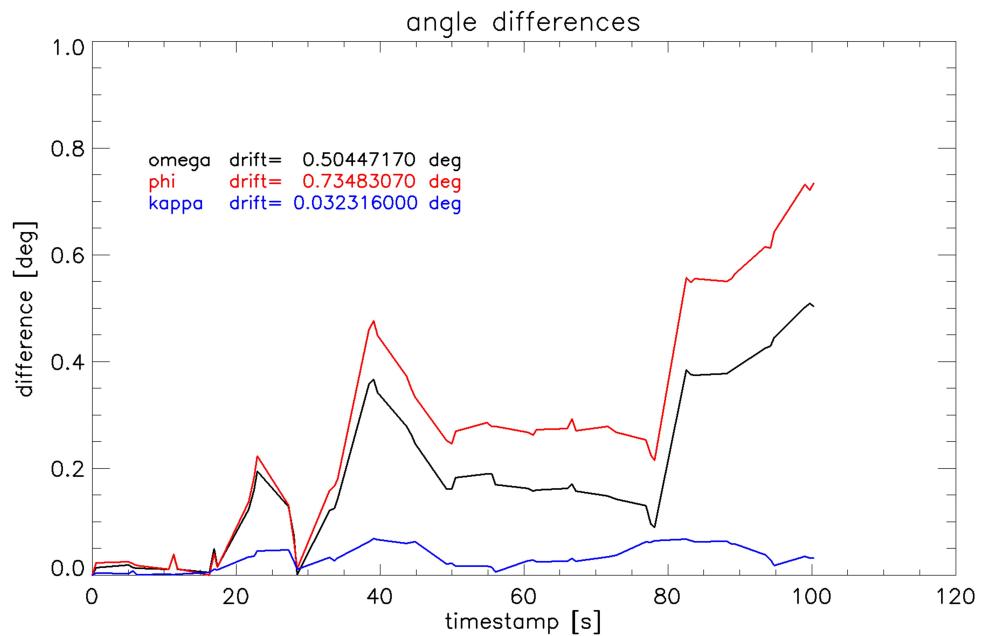


Abbildung 68: Winkelfehler der relativen Orientierung (Überflug 6)

## Überflug 7

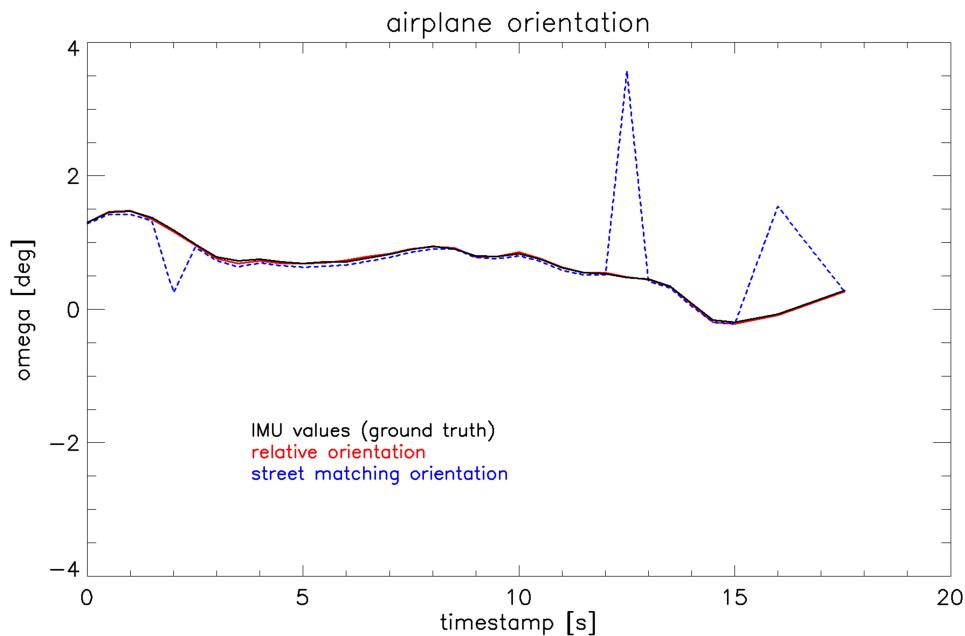


Abbildung 69:  $\omega$ -Winkel der relativen Orientierung (Überflug 7)

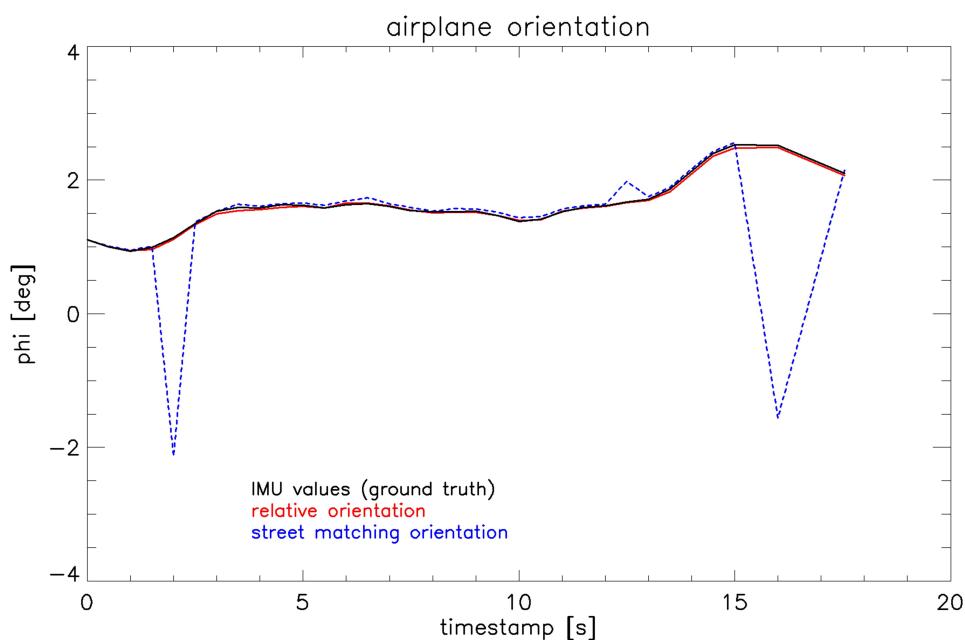


Abbildung 70:  $\phi$ -Winkel der relativen Orientierung (Überflug 7)

## 8 Zusammenfassung und Ausblick

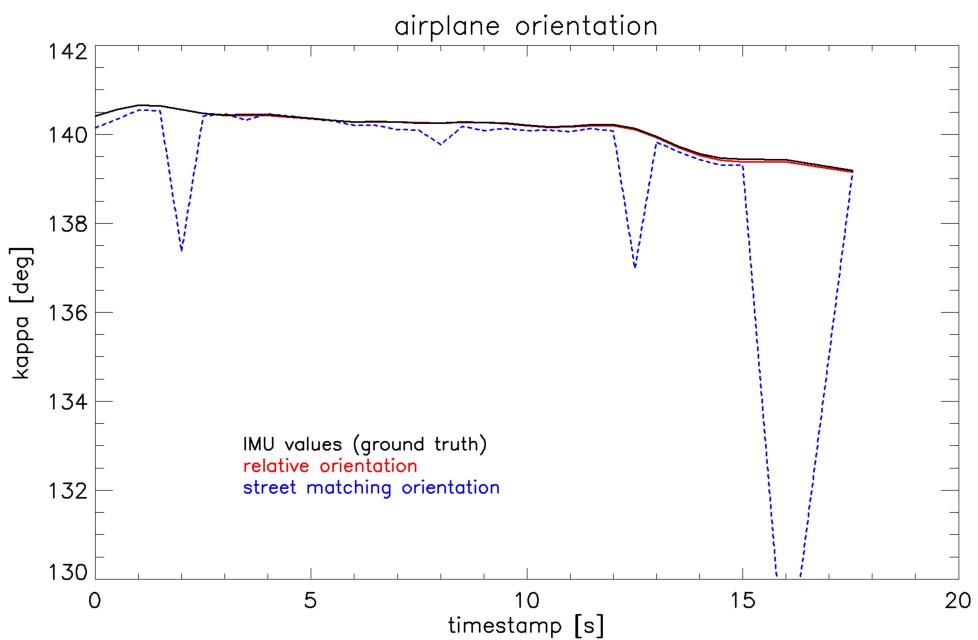


Abbildung 71:  $\kappa$ -Winkel der relativen Orientierung (Überflug 7)

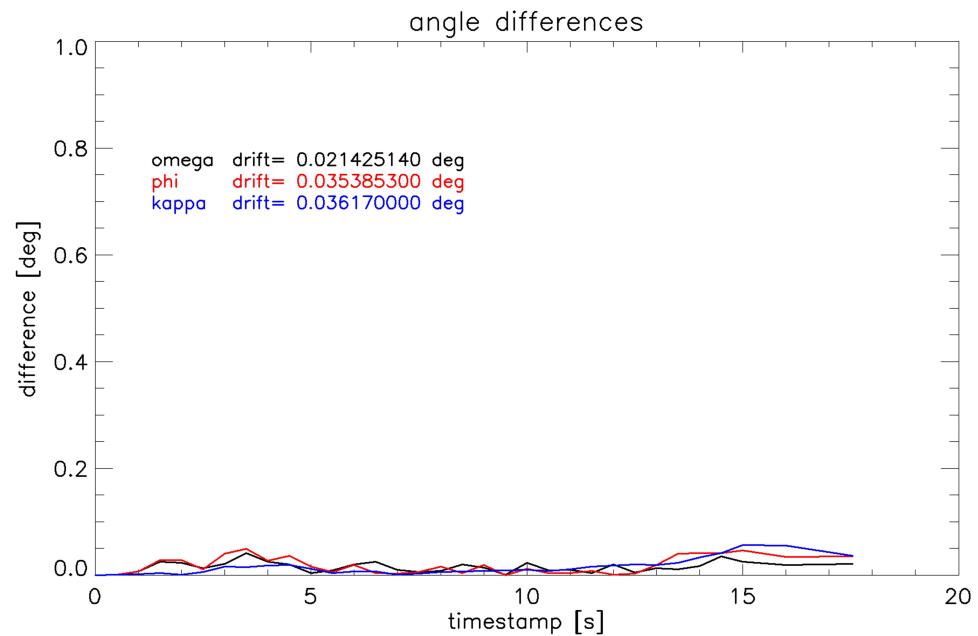


Abbildung 72: Winkelfehler der relativen Orientierung (Überflug 7)

## Überflug 8

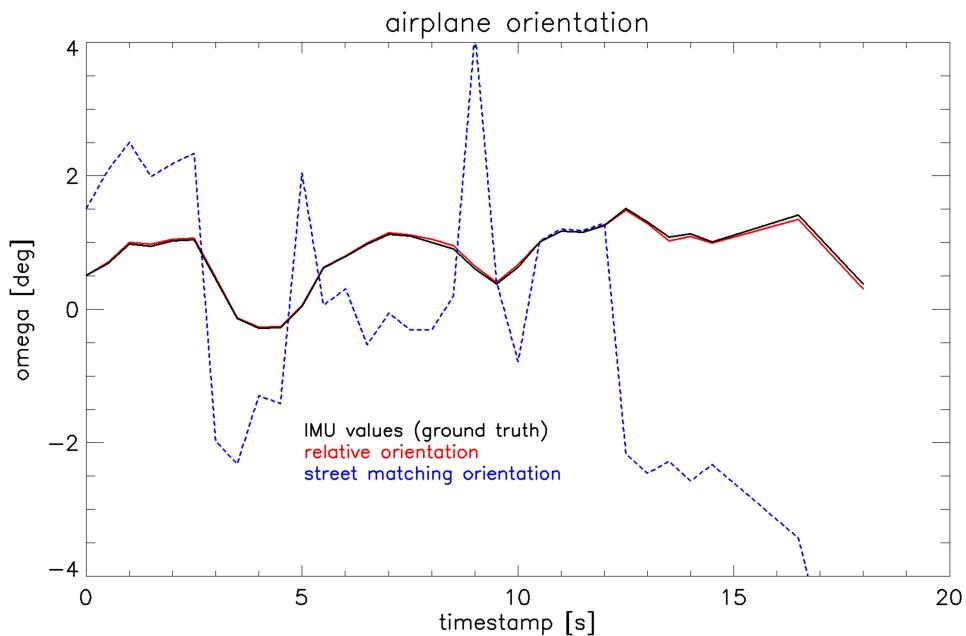


Abbildung 73:  $\omega$ -Winkel der relativen Orientierung (Überflug 8)

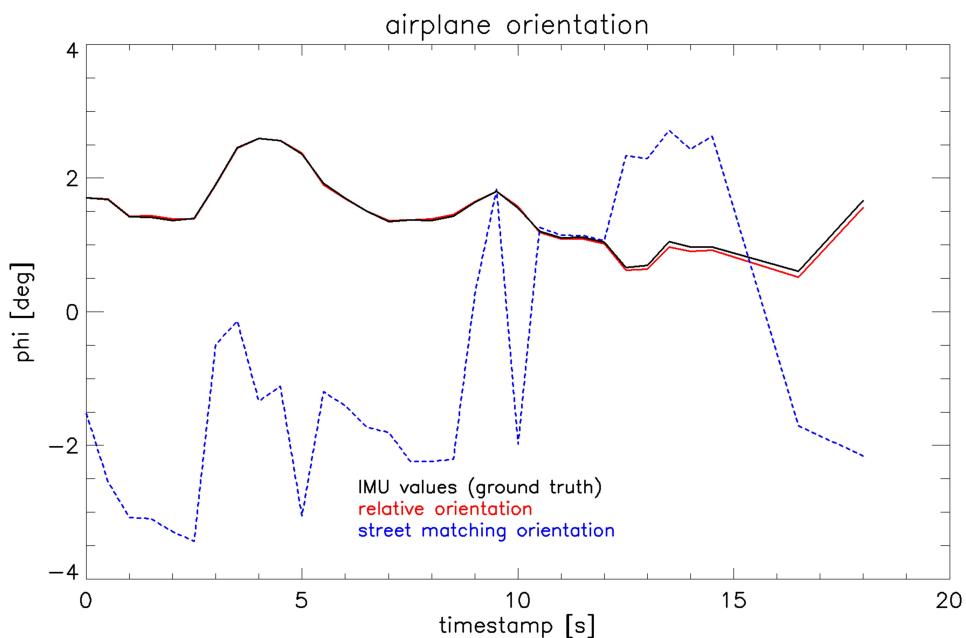


Abbildung 74:  $\phi$ -Winkel der relativen Orientierung (Überflug 8)

## 8 Zusammenfassung und Ausblick

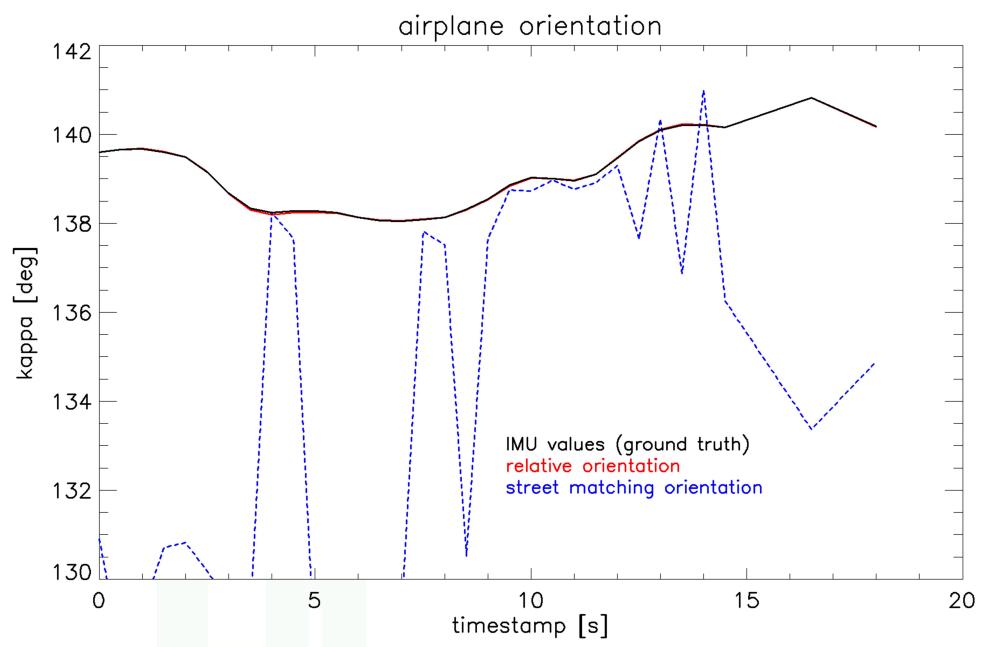


Abbildung 75:  $\kappa$ -Winkel der relativen Orientierung (Überflug 8)

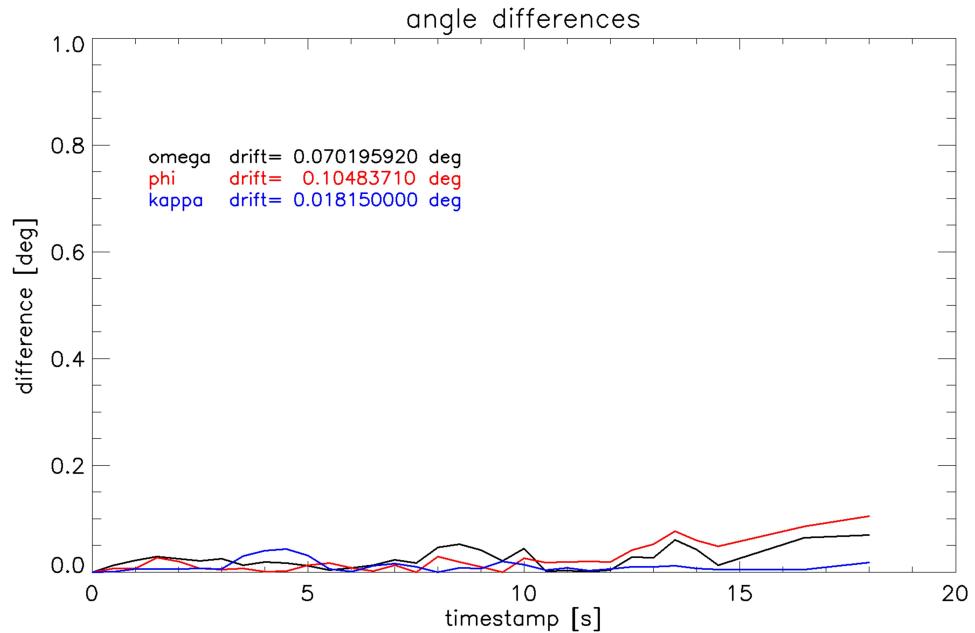


Abbildung 76: Winkelfehler der relativen Orientierung (Überflug 8)

## Überflug 9

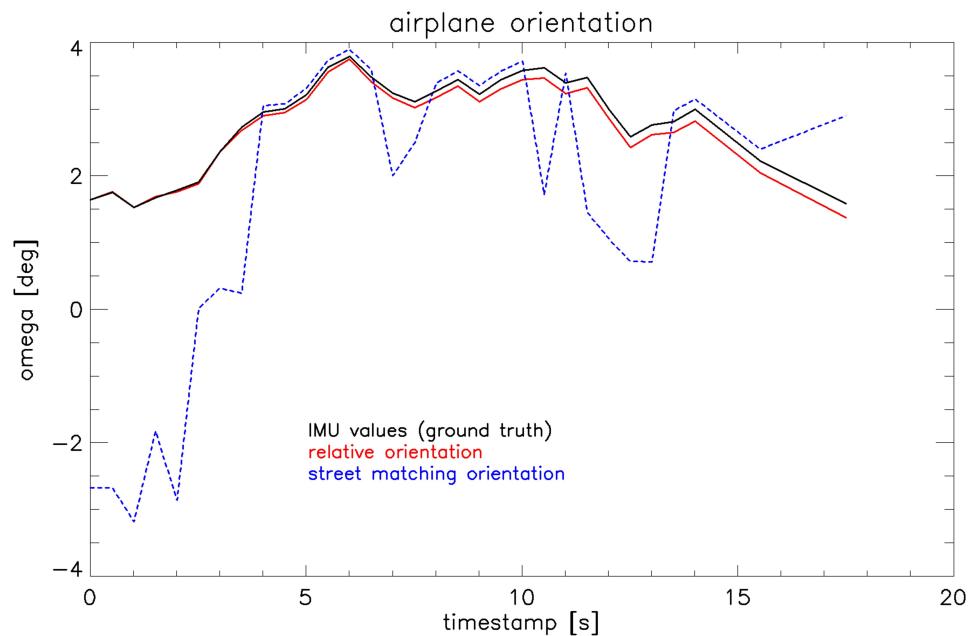


Abbildung 77:  $\omega$ -Winkel der relativen Orientierung (Überflug 9)

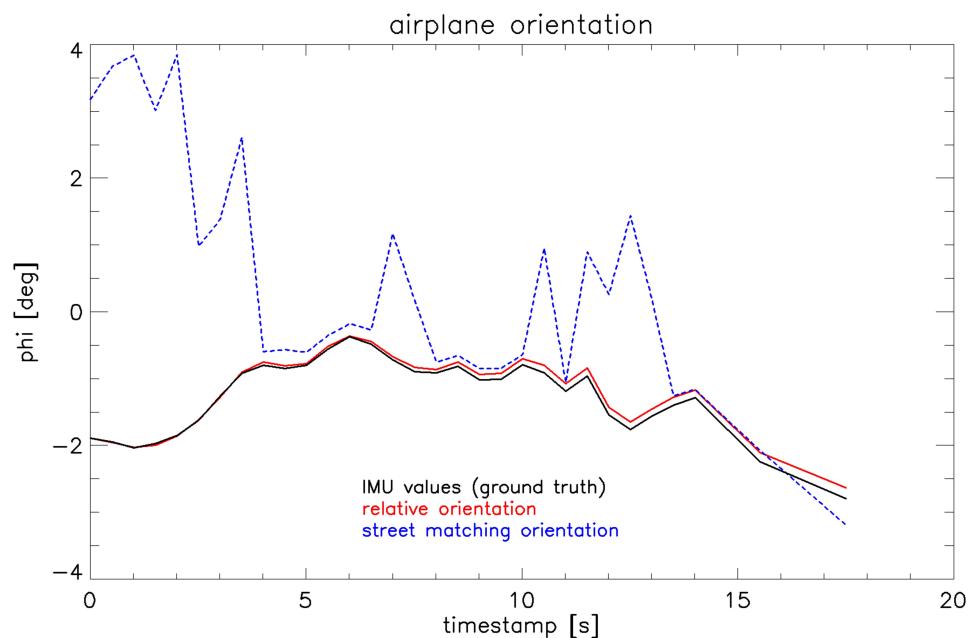


Abbildung 78:  $\phi$ -Winkel der relativen Orientierung (Überflug 9)

## 8 Zusammenfassung und Ausblick

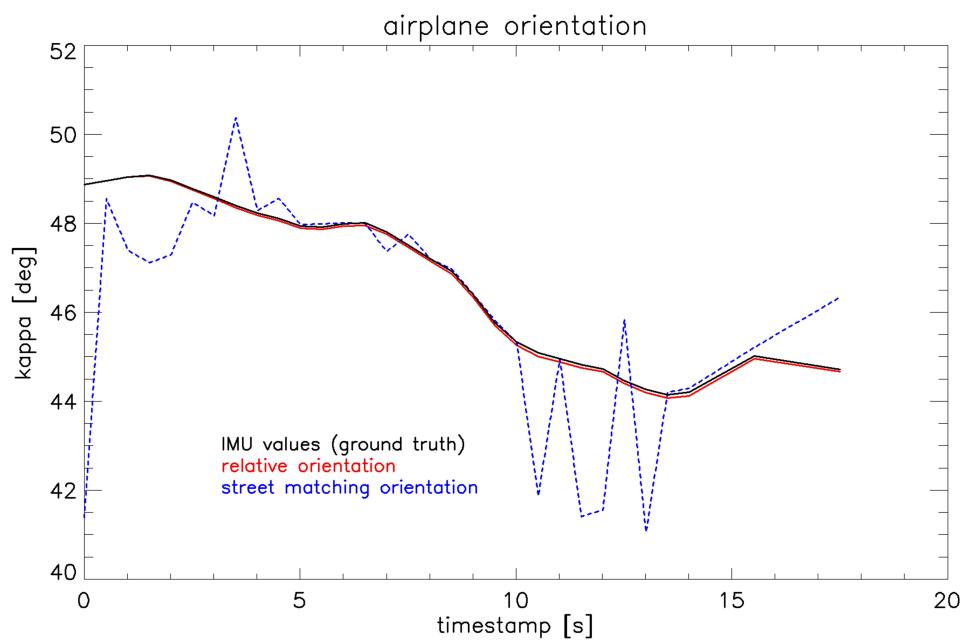


Abbildung 79:  $\kappa$ -Winkel der relativen Orientierung (Überflug 9)

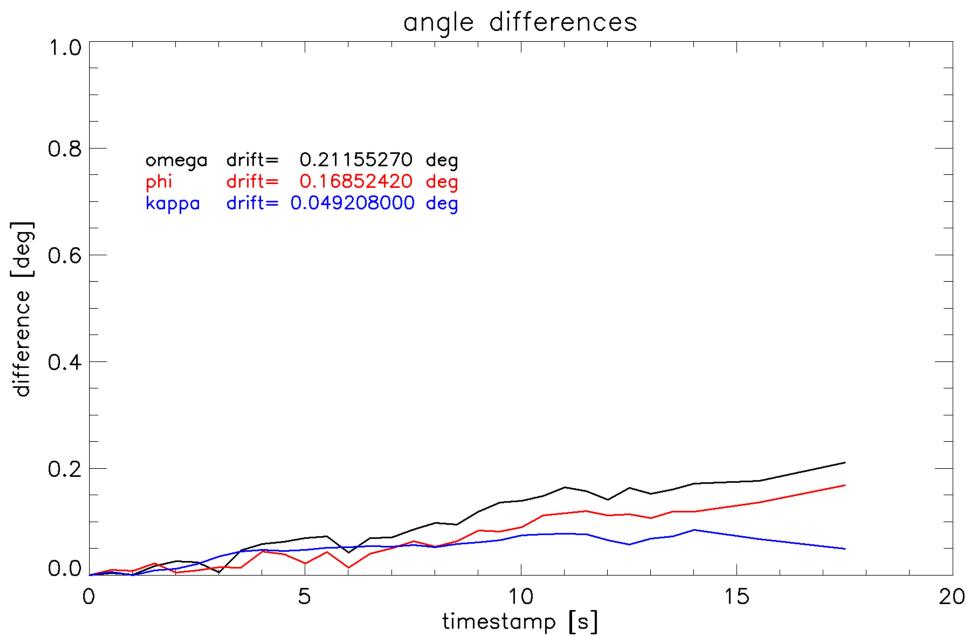


Abbildung 80: Winkelfehler der relativen Orientierung (Überflug 9)

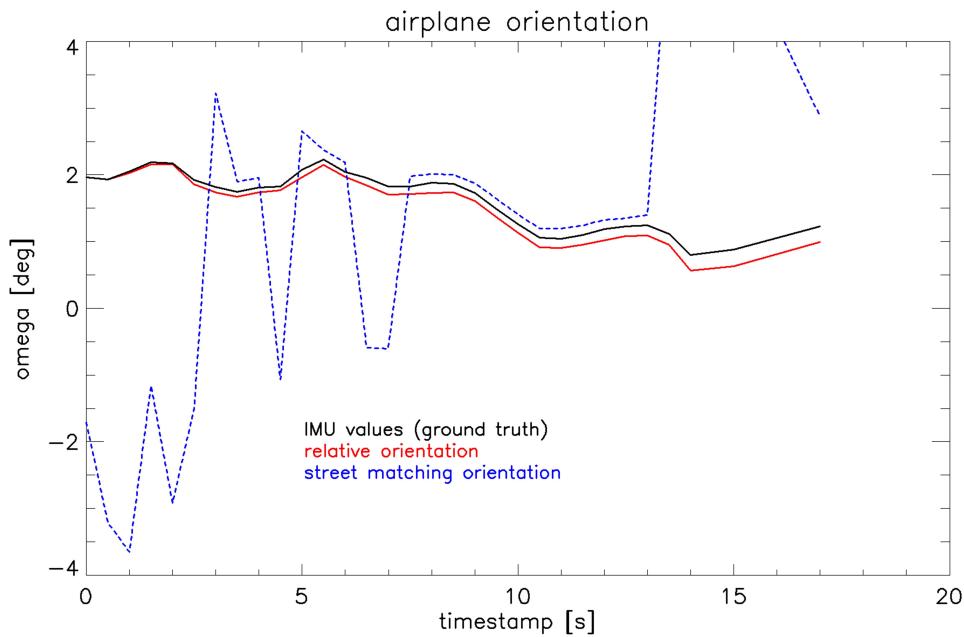


Abbildung 81:  $\omega$ -Winkel der relativen Orientierung (Überflug 10)

### Überflug 10

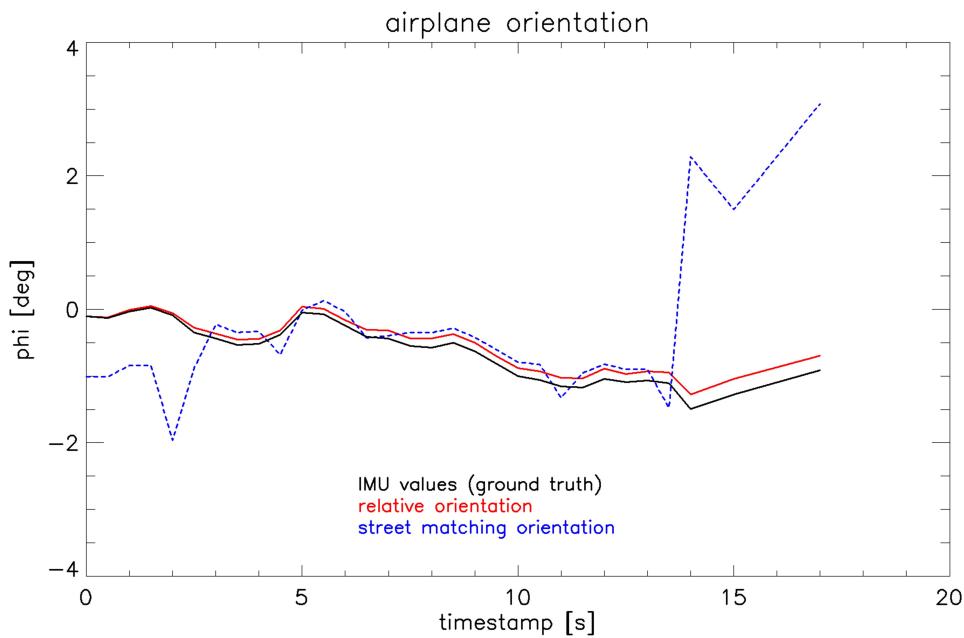


Abbildung 82:  $\phi$ -Winkel der relativen Orientierung (Überflug 10)

## 8 Zusammenfassung und Ausblick

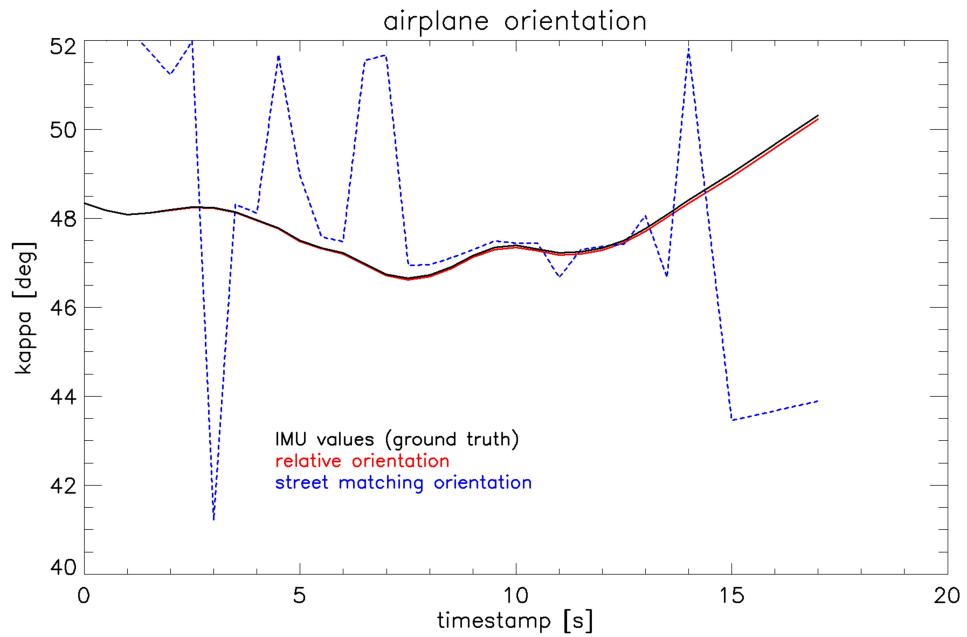


Abbildung 83:  $\kappa$ -Winkel der relativen Orientierung (Überflug 10)

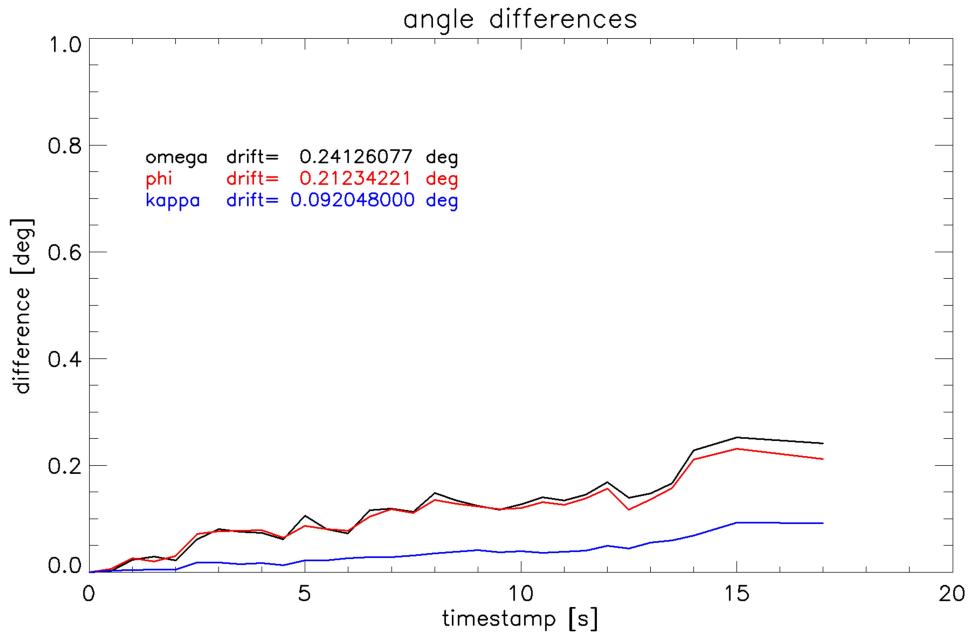


Abbildung 84: Winkelfehler der relativen Orientierung (Überflug 10)

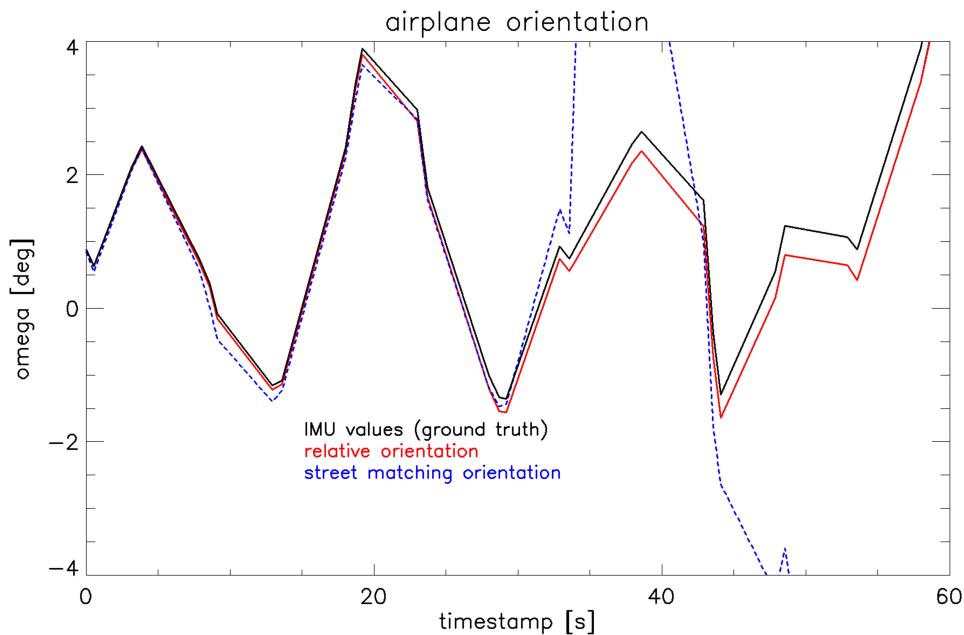


Abbildung 85:  $\omega$ -Winkel der relativen Orientierung (Überflug 11)

### Überflug 11

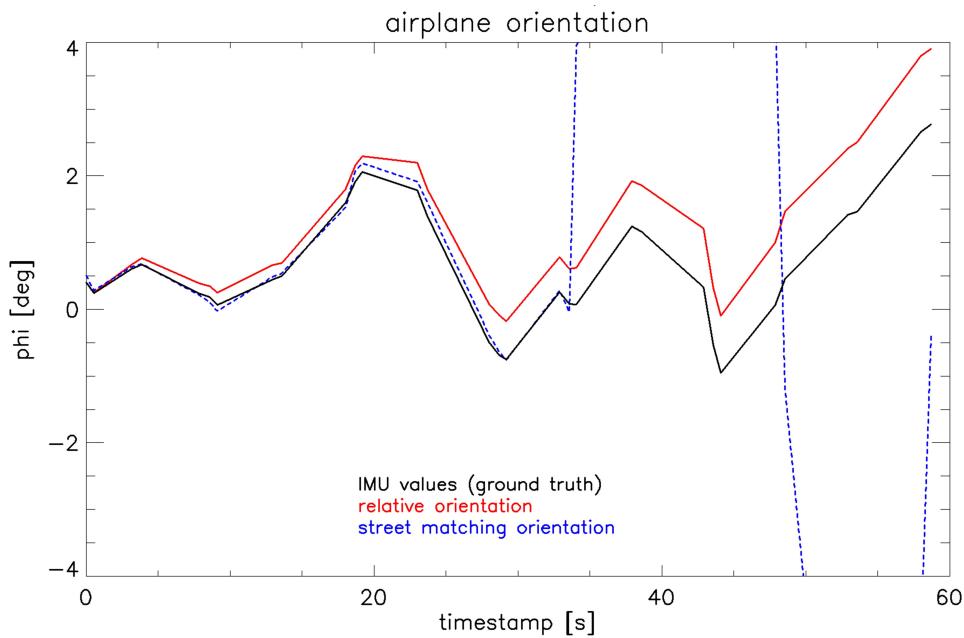


Abbildung 86:  $\phi$ -Winkel der relativen Orientierung (Überflug 11)

## 8 Zusammenfassung und Ausblick

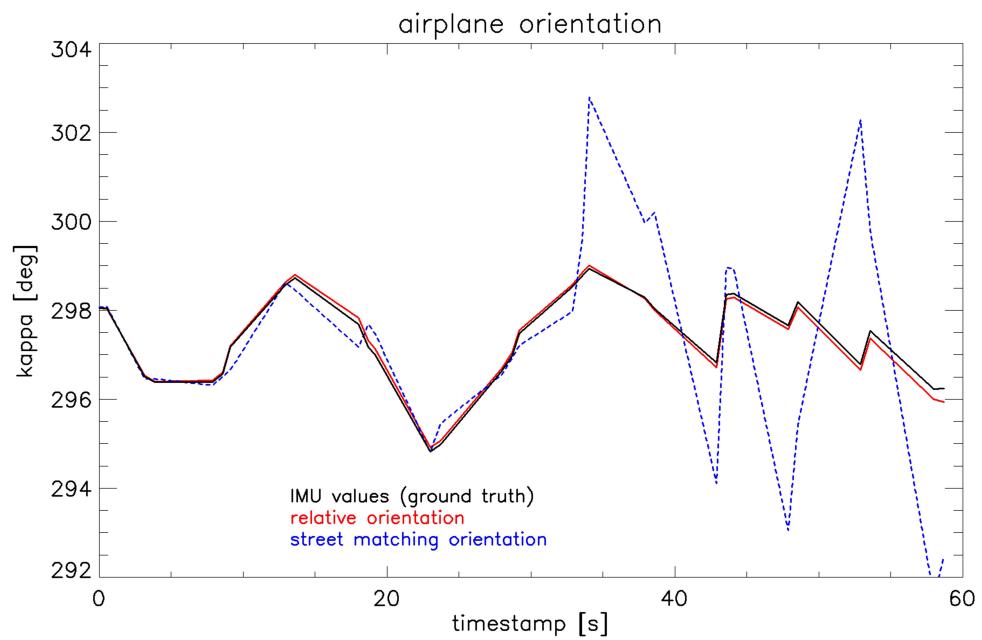


Abbildung 87:  $\kappa$ -Winkel der relativen Orientierung (Überflug 11)

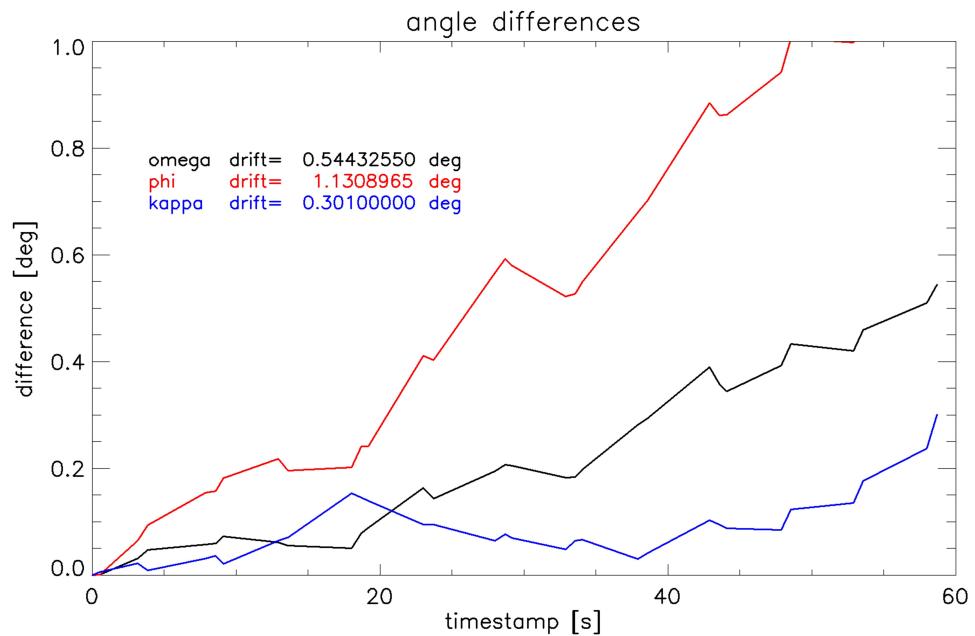


Abbildung 88: Winkelfehler der relativen Orientierung (Überflug 11)

## **Ergebnisse des Navigationsfilters**

## Überflug 1

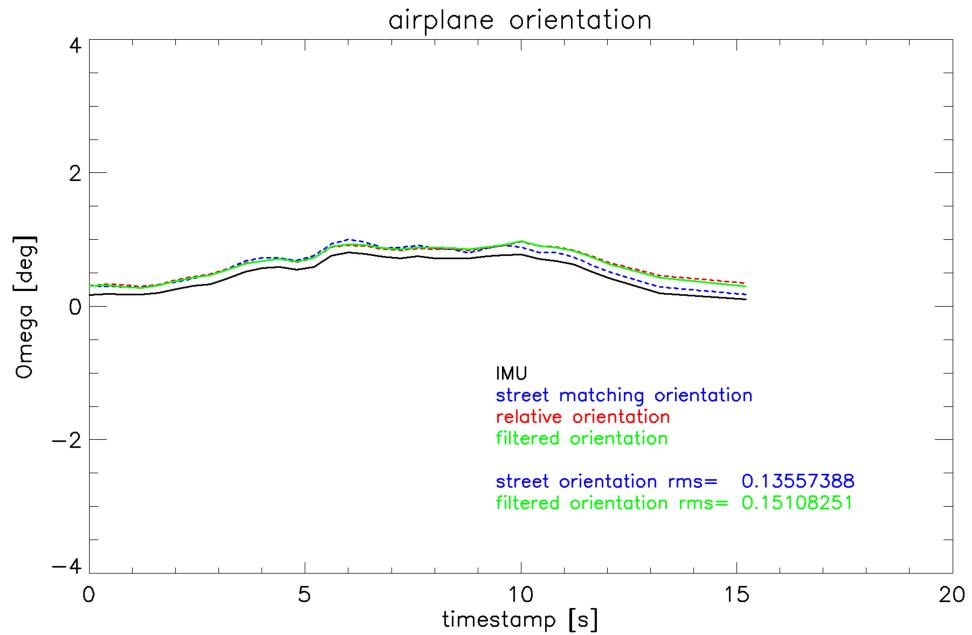


Abbildung 89:  $\omega$ -Winkel des Navigationsfilters (Überflug 1)

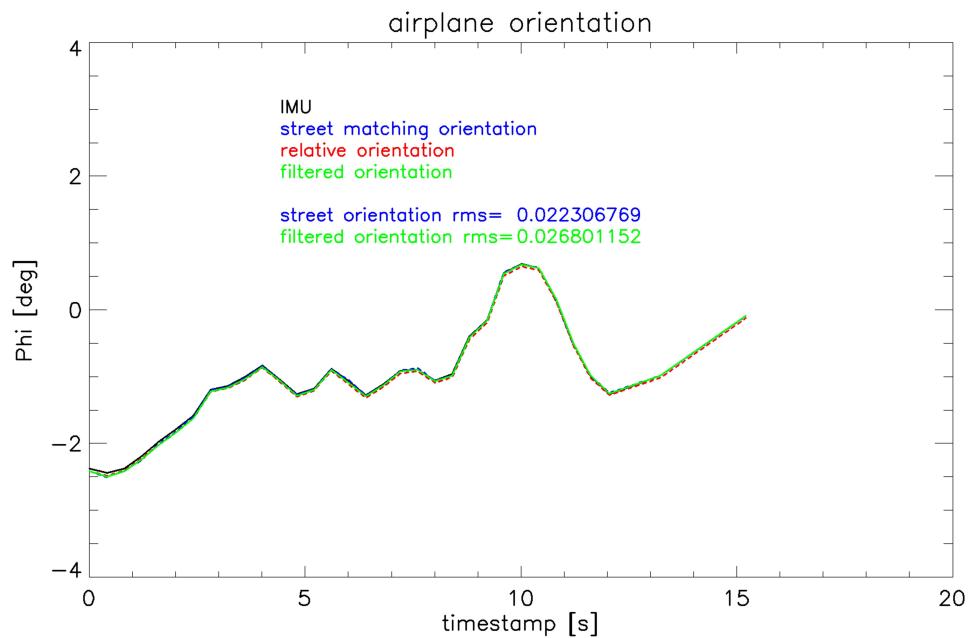


Abbildung 90:  $\phi$ -Winkel des Navigationsfilters (Überflug 1)

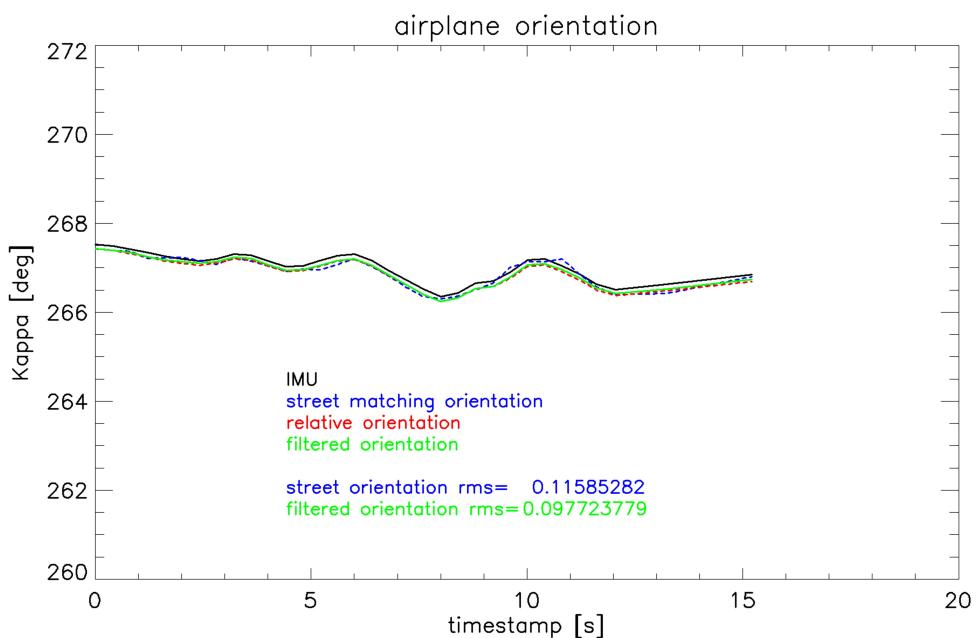


Abbildung 91:  $\kappa$ -Winkel des Navigationsfilters (Überflug 1)

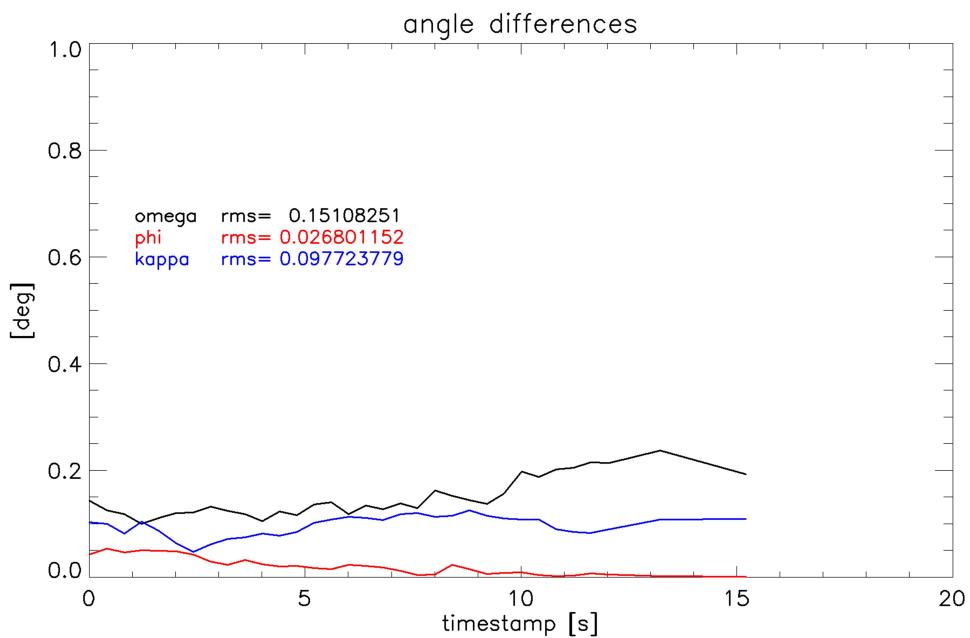


Abbildung 92: Winkelfehler des Navigationsfilters (Überflug 1)

## Überflug 2

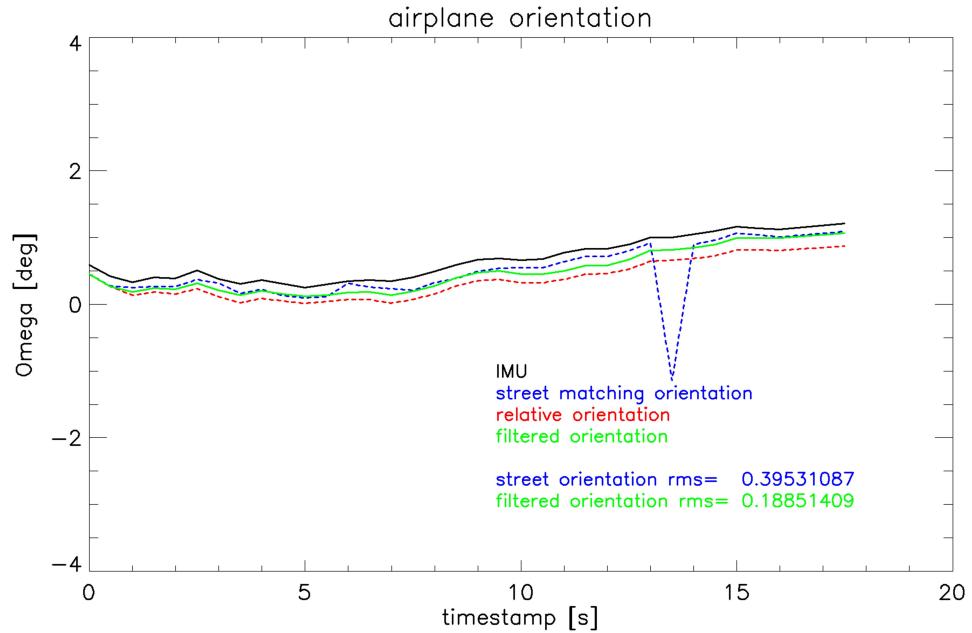


Abbildung 93:  $\omega$ -Winkel des Navigationsfilters (Überflug 2)

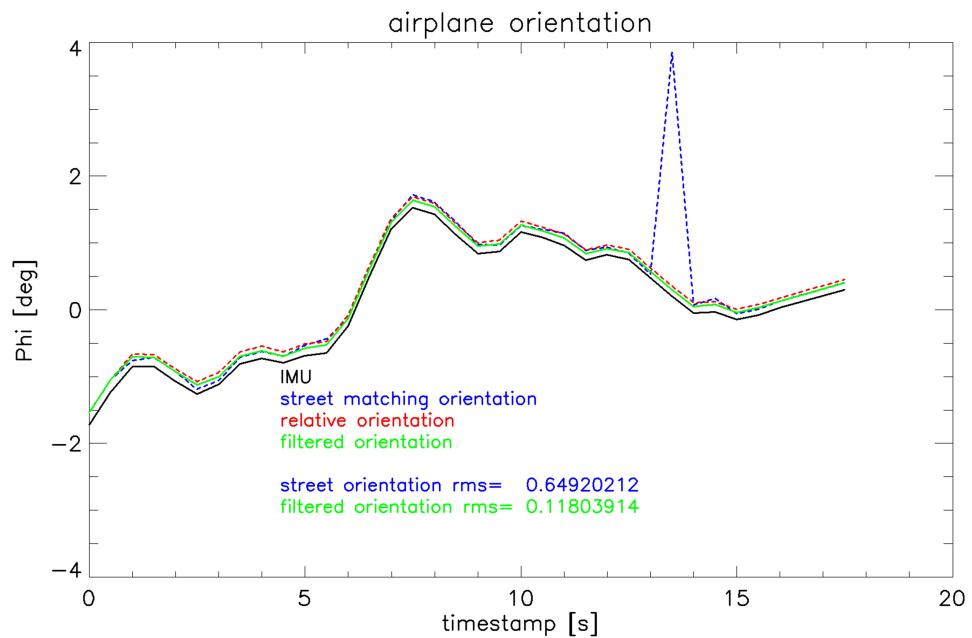


Abbildung 94:  $\phi$ -Winkel des Navigationsfilters (Überflug 2)

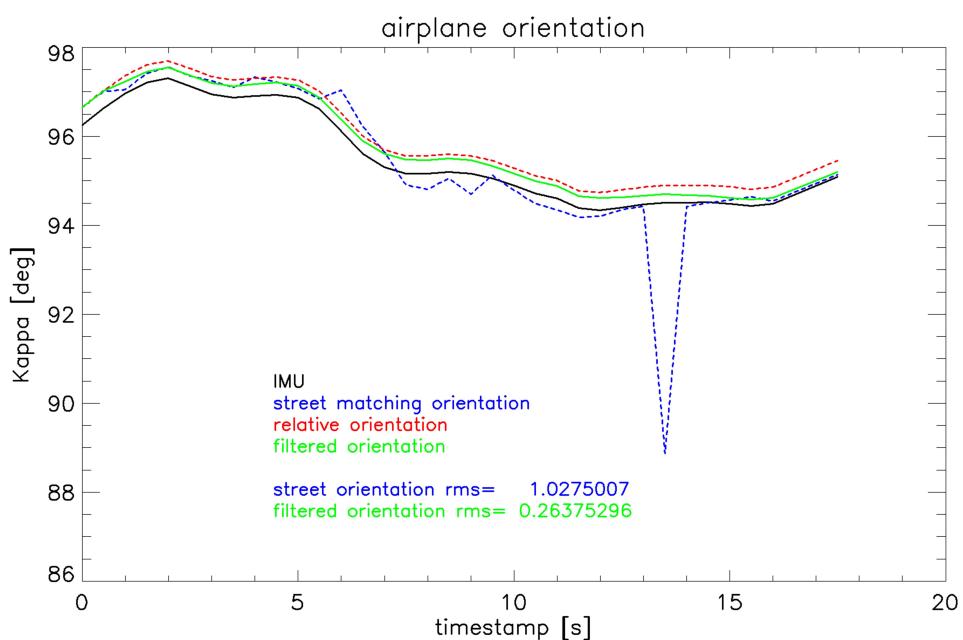


Abbildung 95:  $\kappa$ -Winkel des Navigationsfilters (Überflug 2)

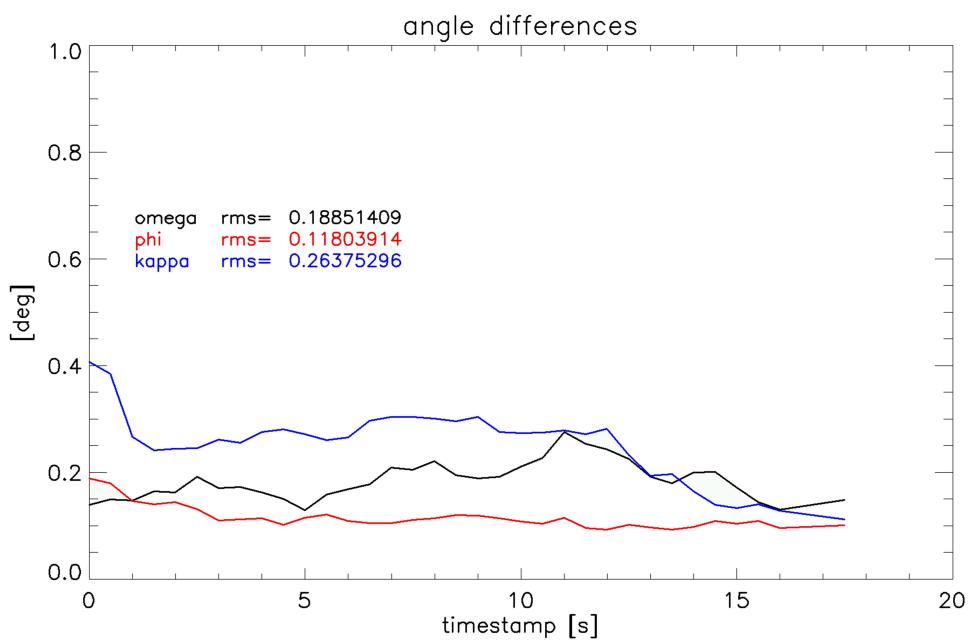


Abbildung 96: Winkelfehler des Navigationsfilters (Überflug 2)

### Überflug 3

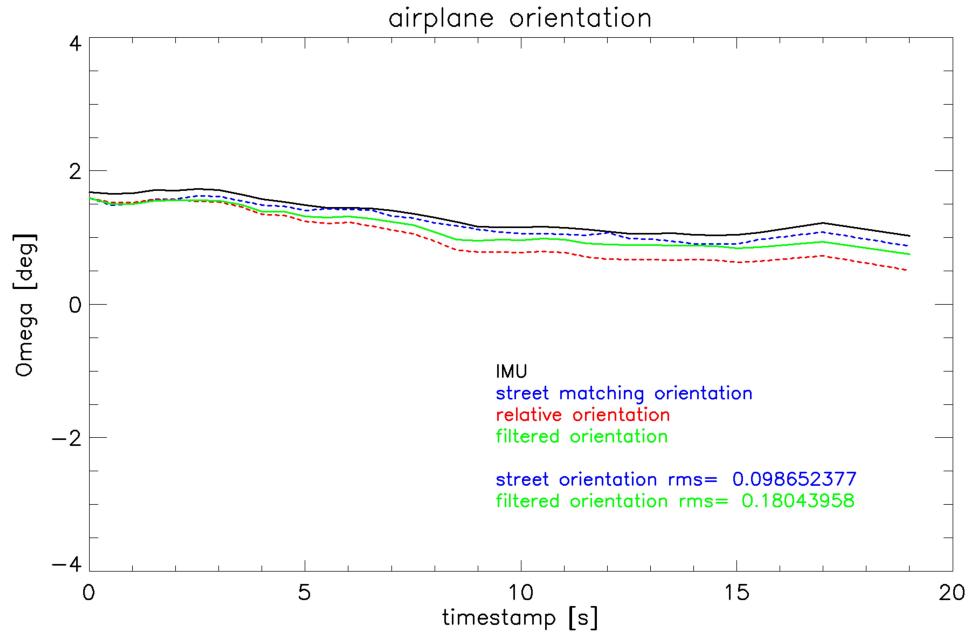


Abbildung 97:  $\omega$ -Winkel des Navigationsfilters (Überflug 3)

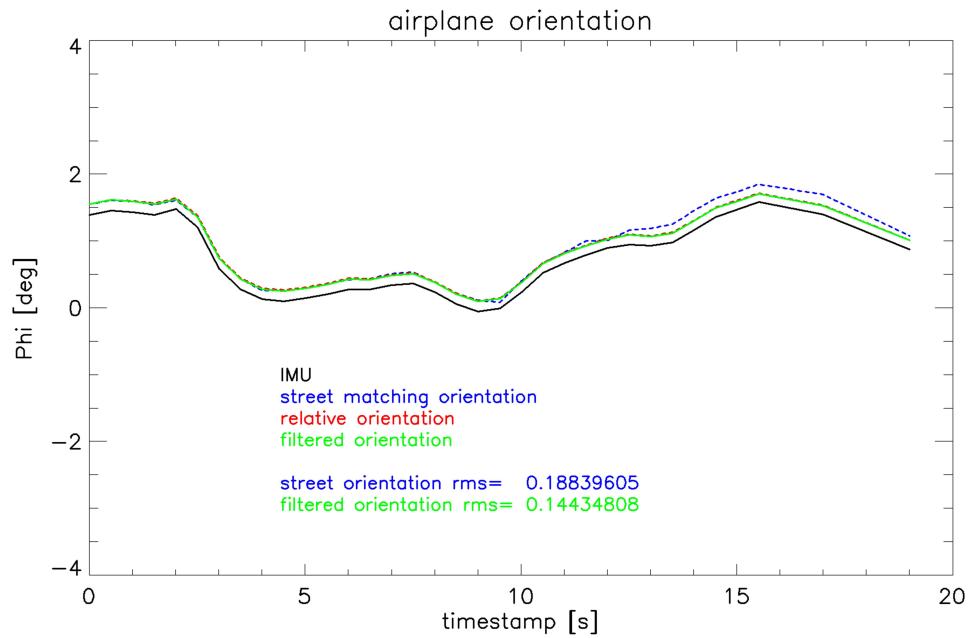


Abbildung 98:  $\phi$ -Winkel des Navigationsfilters (Überflug 3)

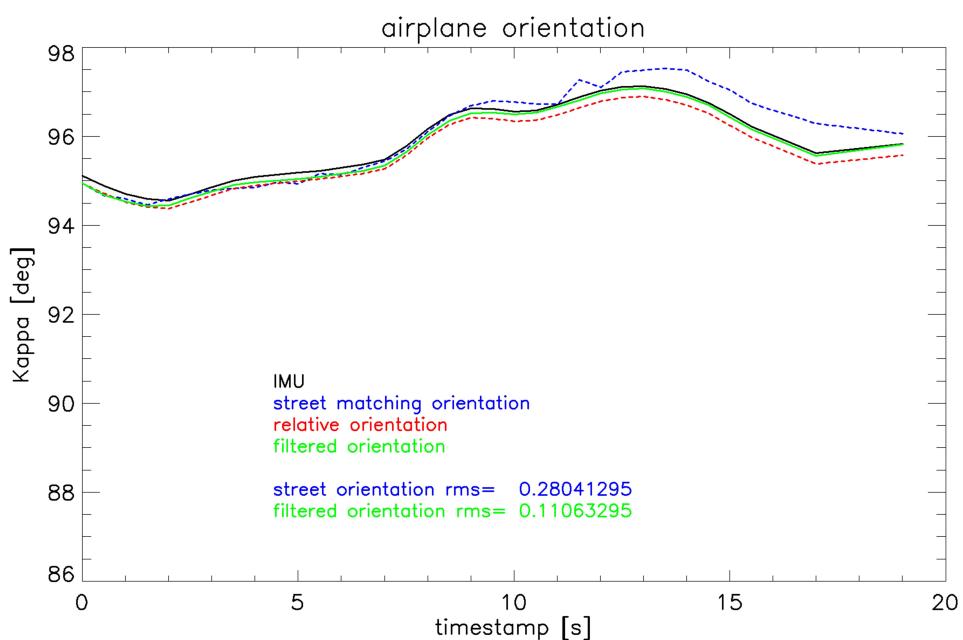


Abbildung 99:  $\kappa$ -Winkel des Navigationsfilters (Überflug 3)

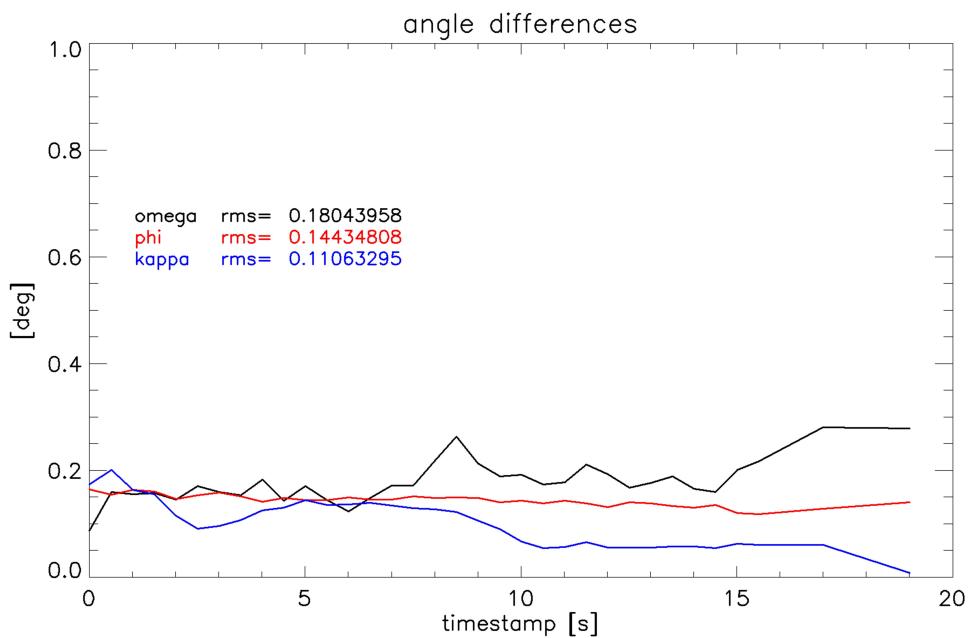


Abbildung 100: Winkelfehler des Navigationsfilters (Überflug 3)

### Überflug 4

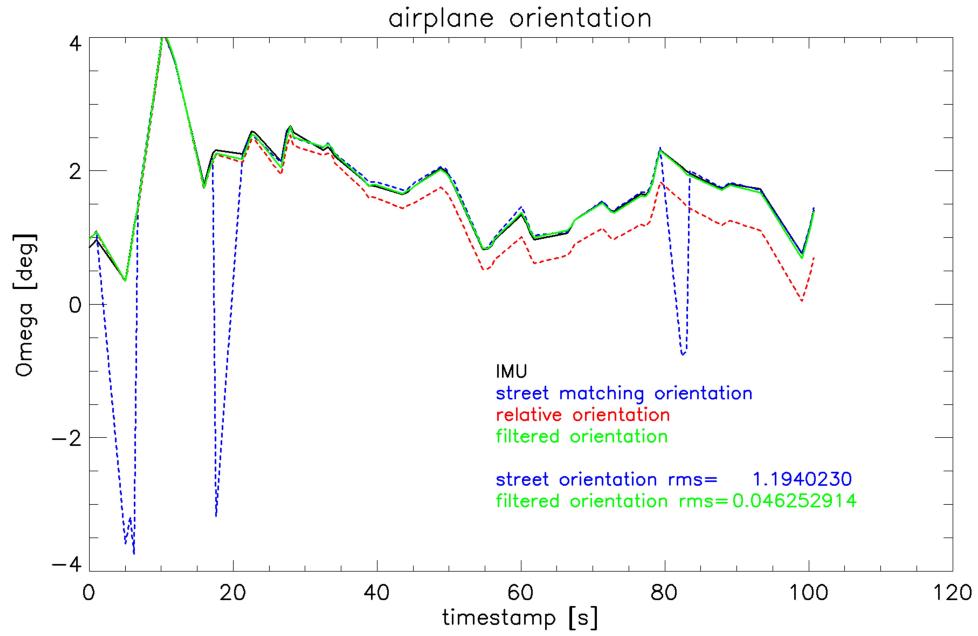


Abbildung 101:  $\omega$ -Winkel des Navigationsfilters (Überflug 4)

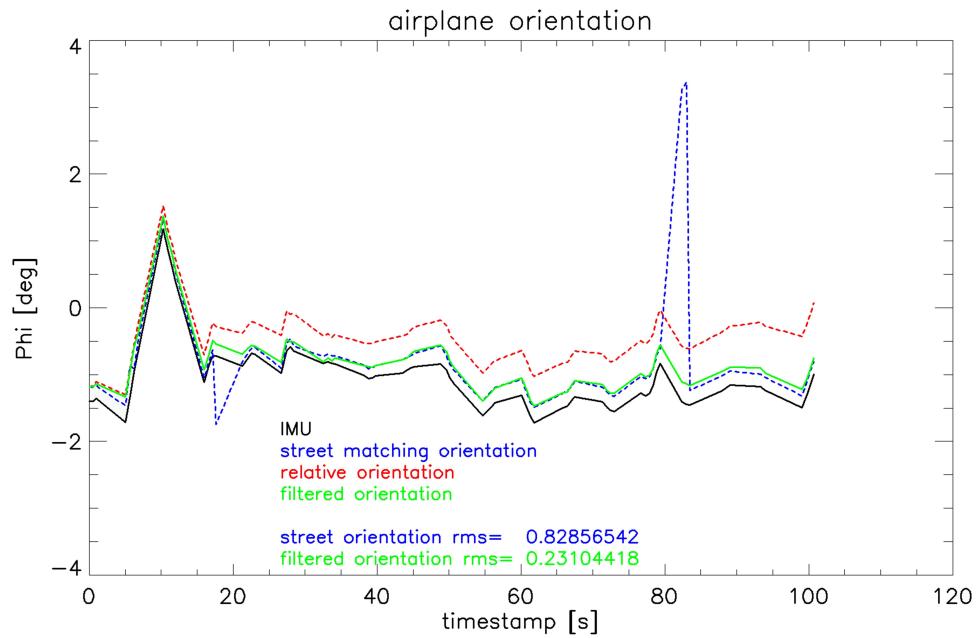


Abbildung 102:  $\phi$ -Winkel des Navigationsfilters (Überflug 4)

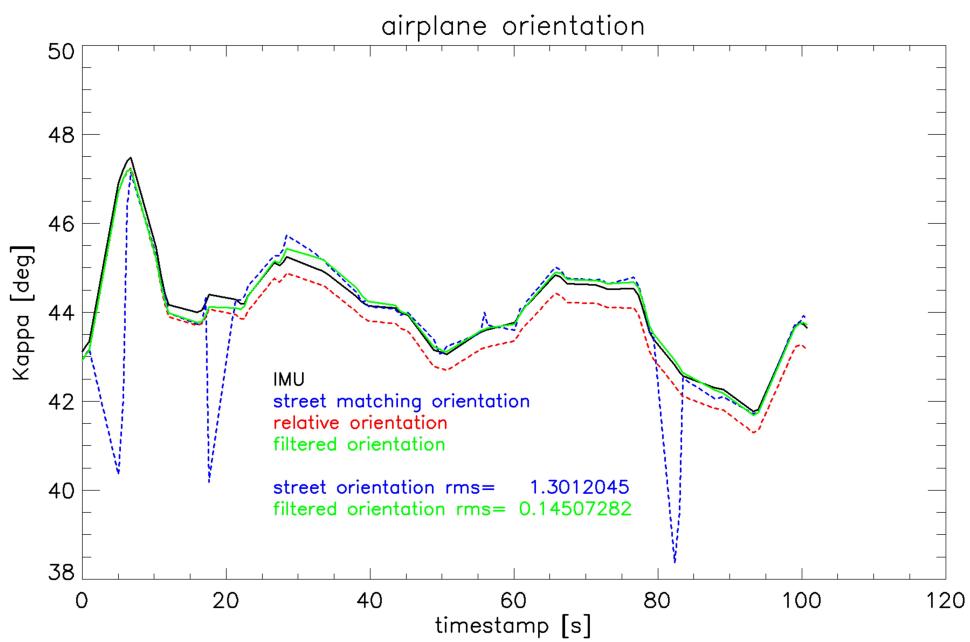


Abbildung 103:  $\kappa$ -Winkel des Navigationsfilters (Überflug 4)

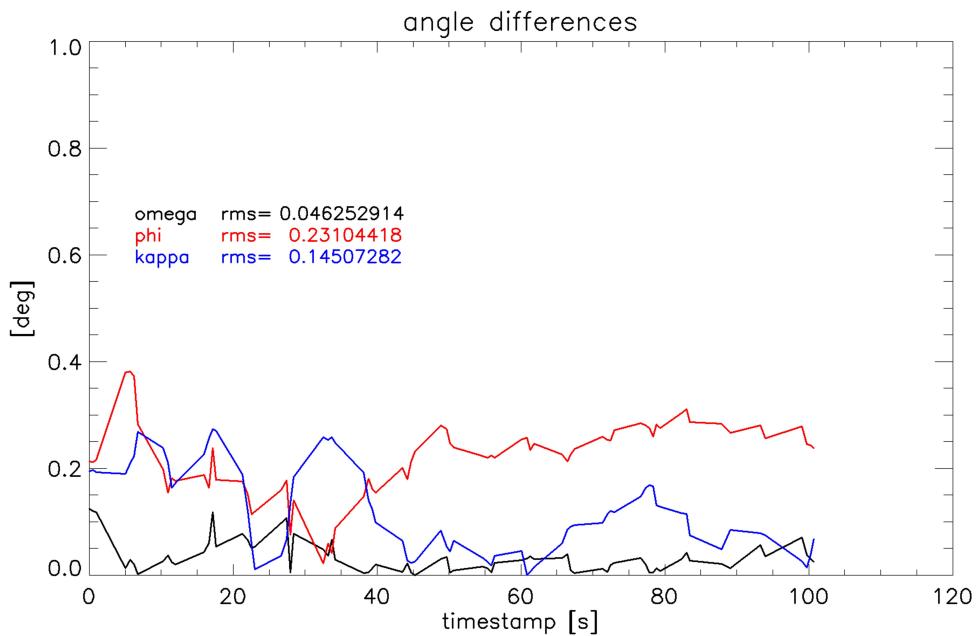


Abbildung 104: Winkelfehler des Navigationsfilters (Überflug 4)

### Überflug 5

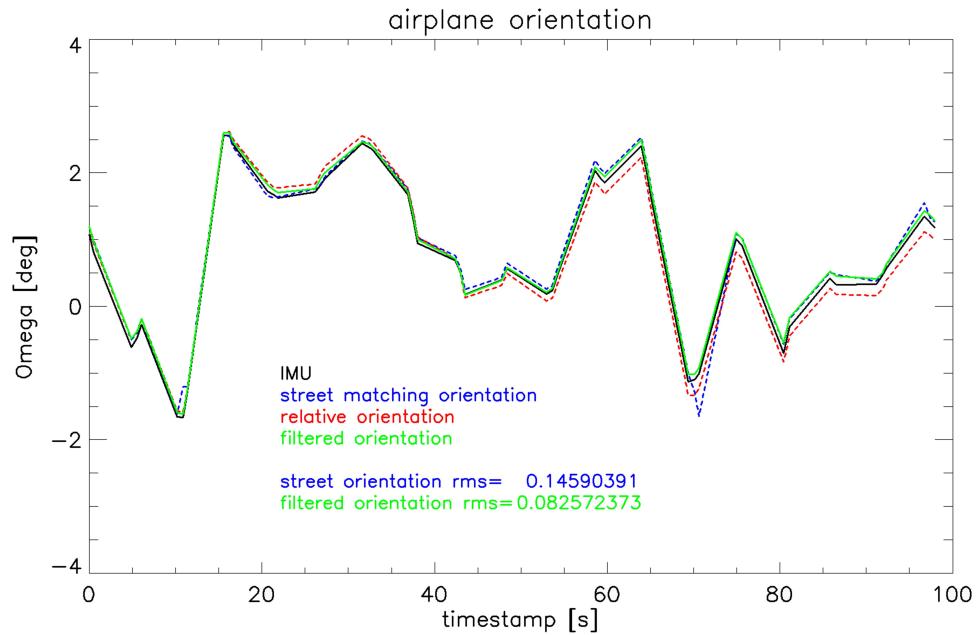


Abbildung 105:  $\omega$ -Winkel des Navigationsfilters (Überflug 5)

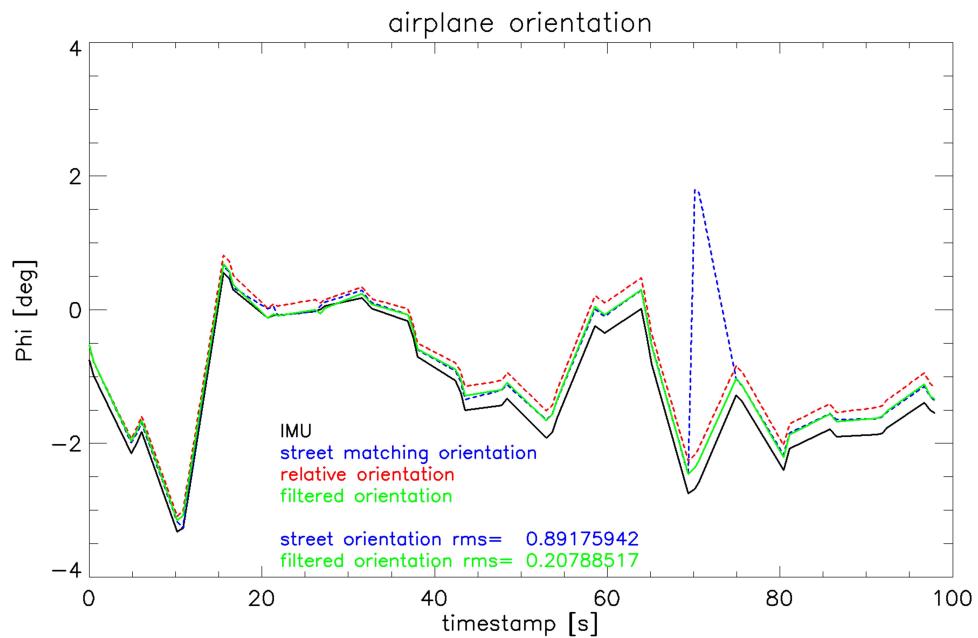


Abbildung 106:  $\phi$ -Winkel des Navigationsfilters (Überflug 5)

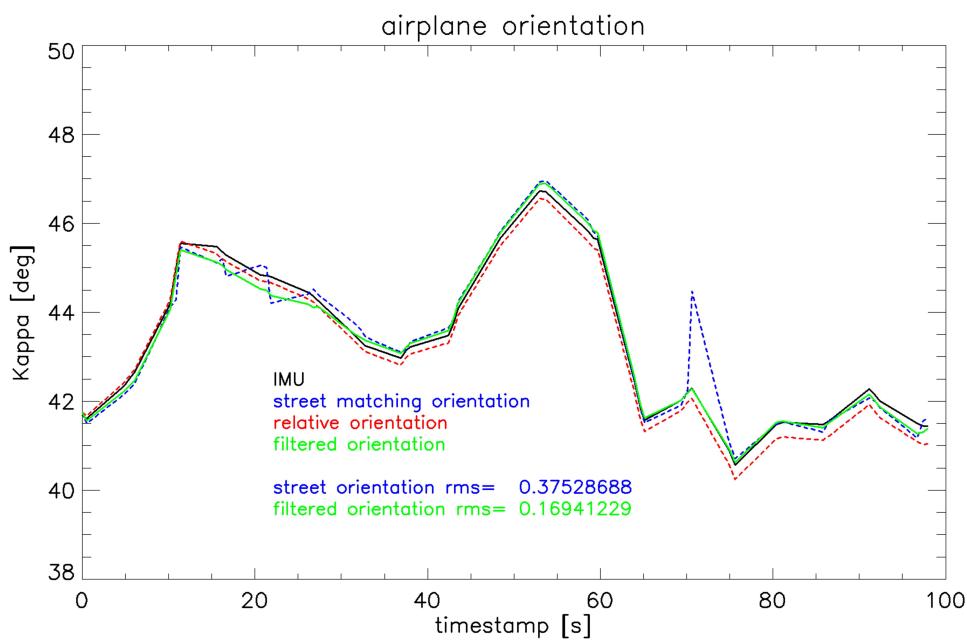


Abbildung 107:  $\kappa$ -Winkel des Navigationsfilters (Überflug 5)

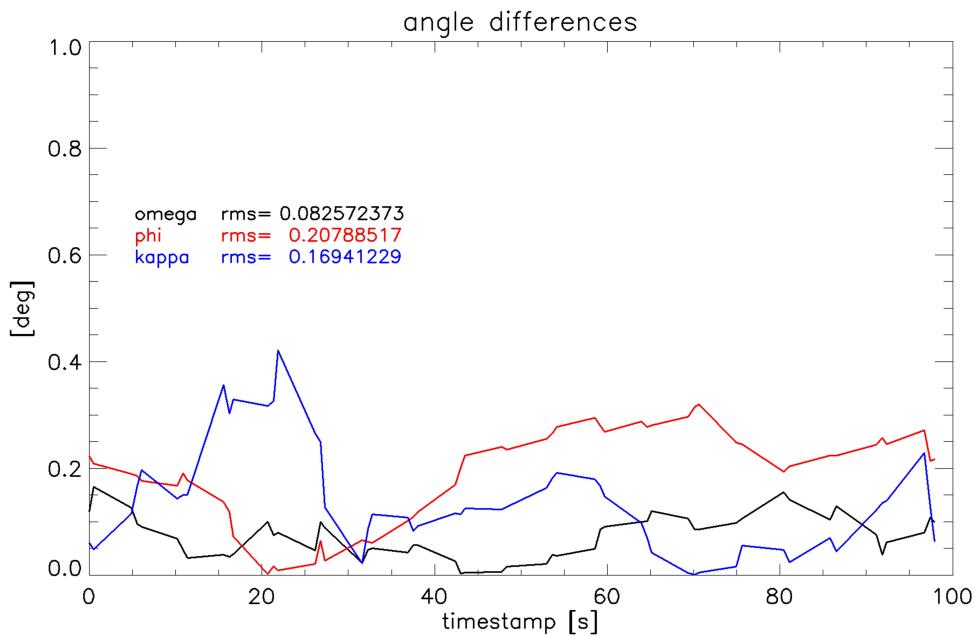


Abbildung 108: Winkelfehler des Navigationsfilters (Überflug 5)

## 8 Zusammenfassung und Ausblick

### Überflug 6

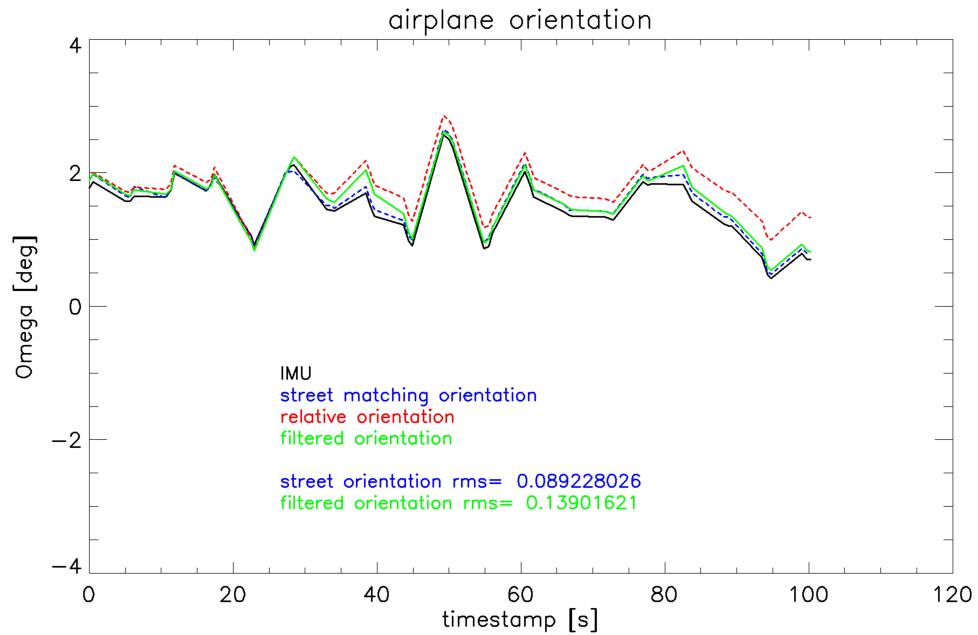


Abbildung 109:  $\omega$ -Winkel des Navigationsfilters (Überflug 6)

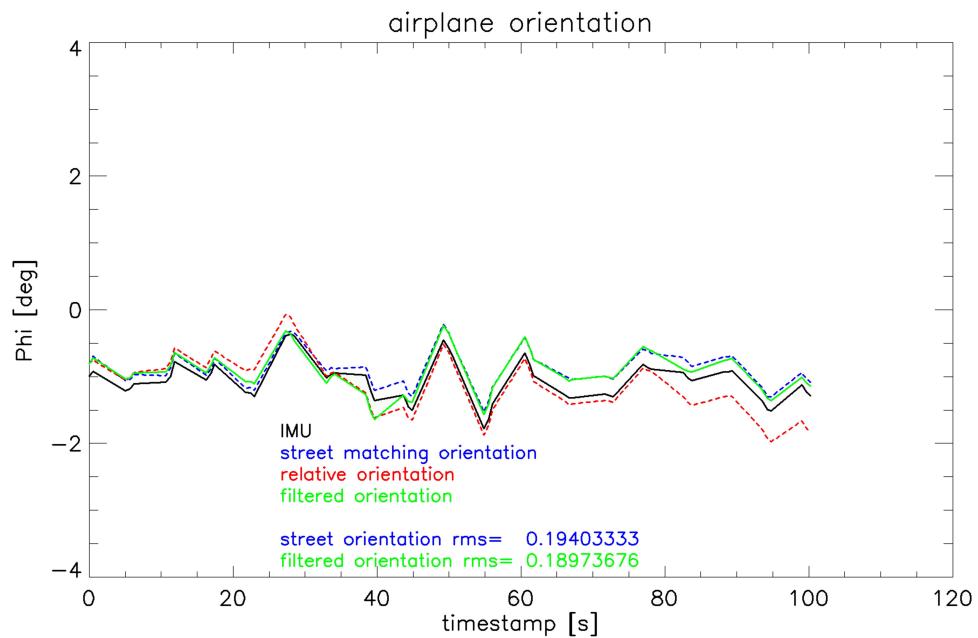


Abbildung 110:  $\phi$ -Winkel des Navigationsfilters (Überflug 6)

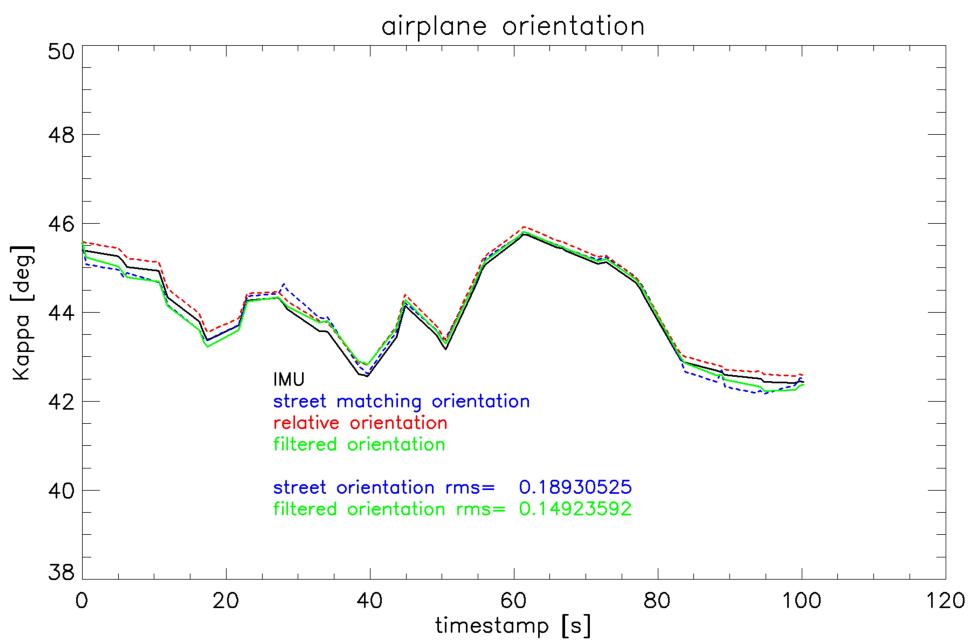


Abbildung 111:  $\kappa$ -Winkel des Navigationsfilters (Überflug 6)

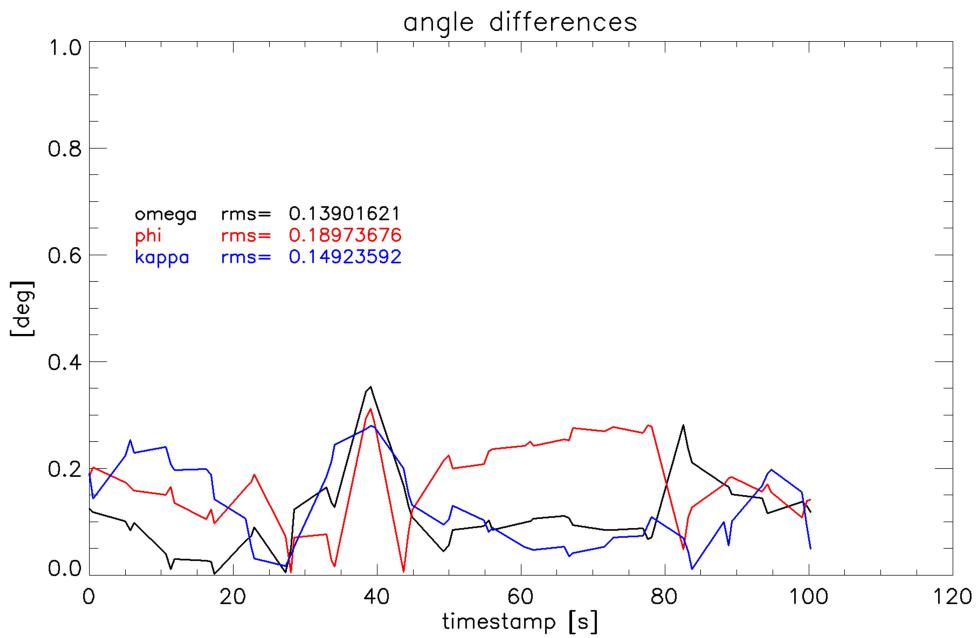


Abbildung 112: Winkelfehler des Navigationsfilters (Überflug 6)

### Überflug 7

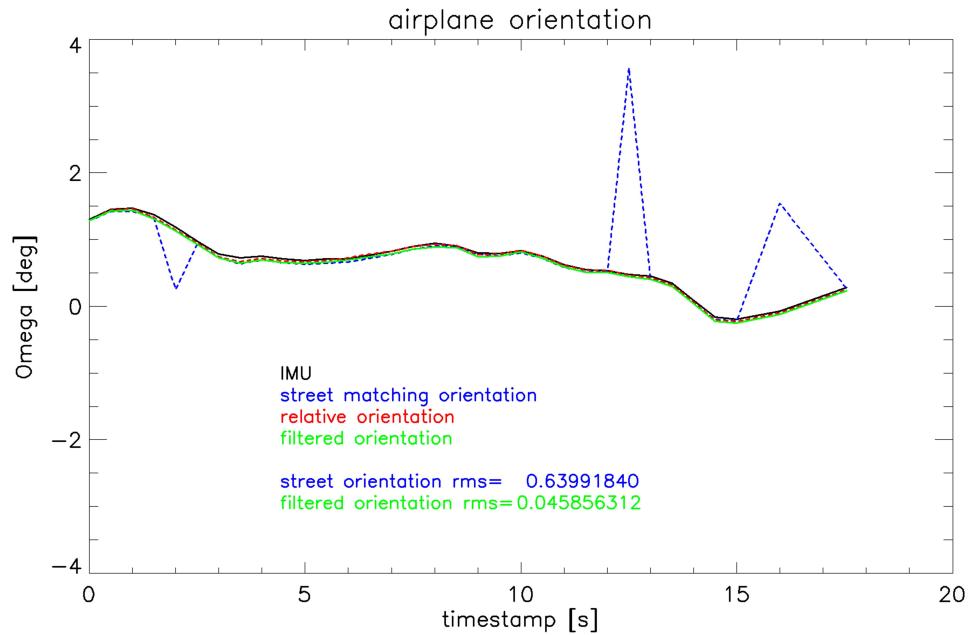


Abbildung 113:  $\omega$ -Winkel des Navigationsfilters (Überflug 7)

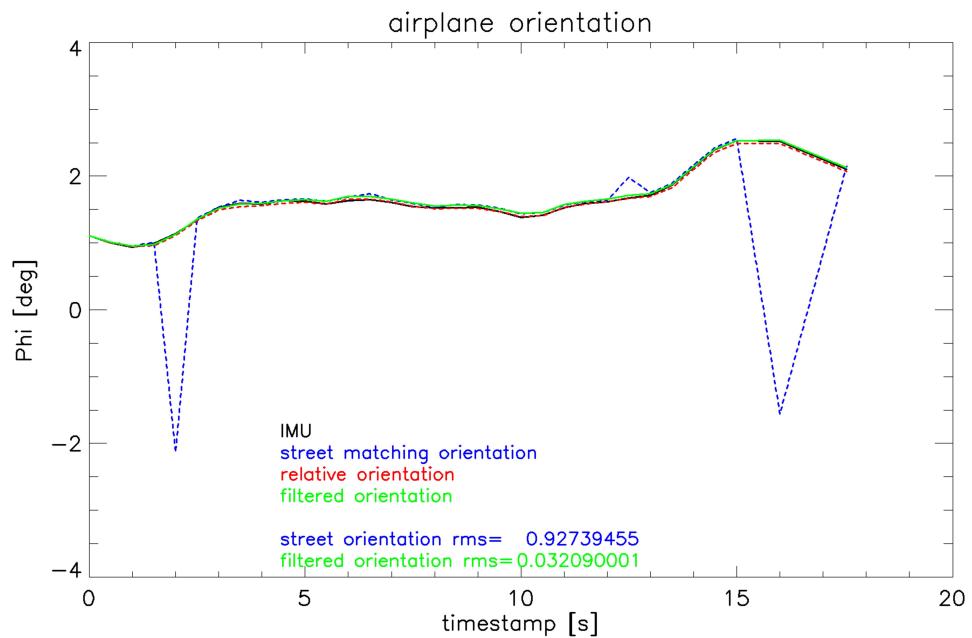


Abbildung 114:  $\phi$ -Winkel des Navigationsfilters (Überflug 7)

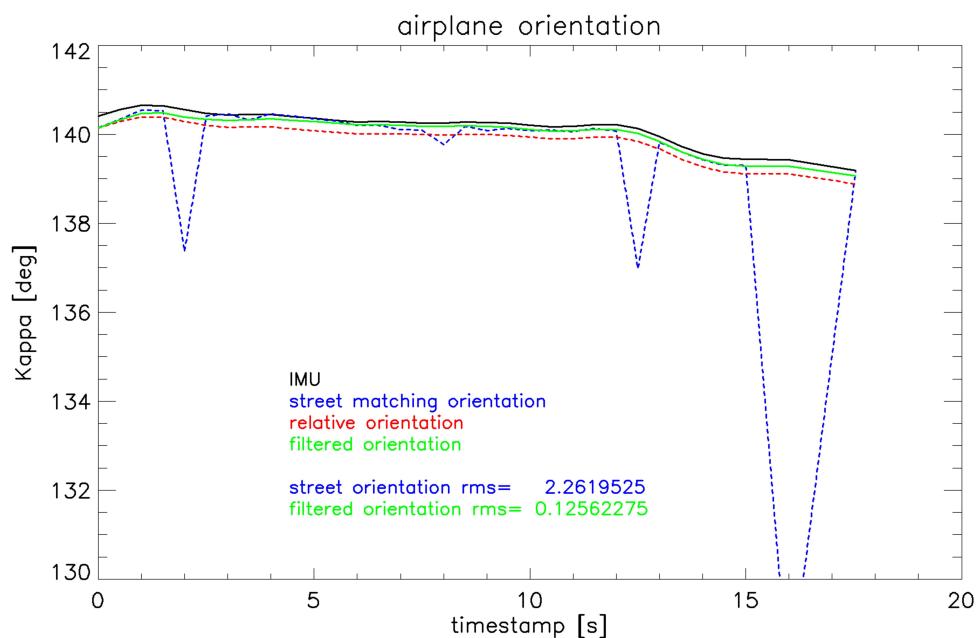


Abbildung 115:  $\kappa$ -Winkel des Navigationsfilters (Überflug 7)

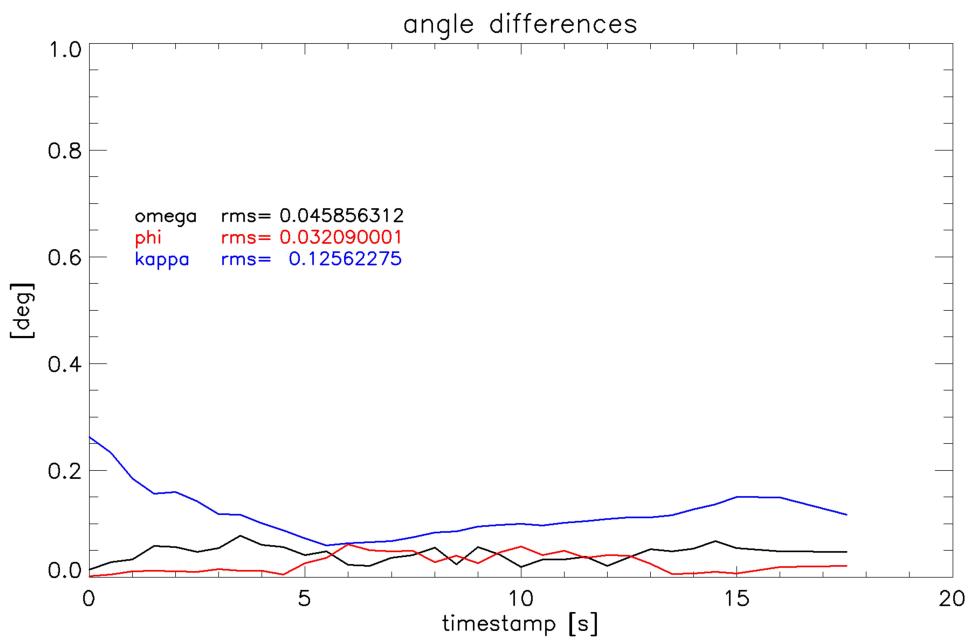


Abbildung 116: Winkelfehler des Navigationsfilters (Überflug 7)

## Überflug 8

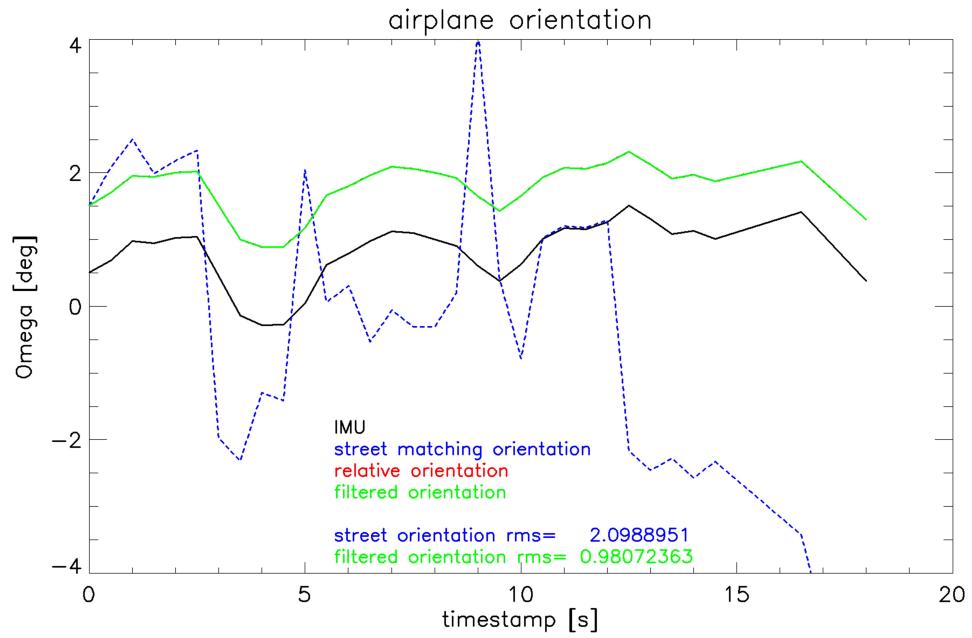


Abbildung 117:  $\omega$ -Winkel des Navigationsfilters (Überflug 8)

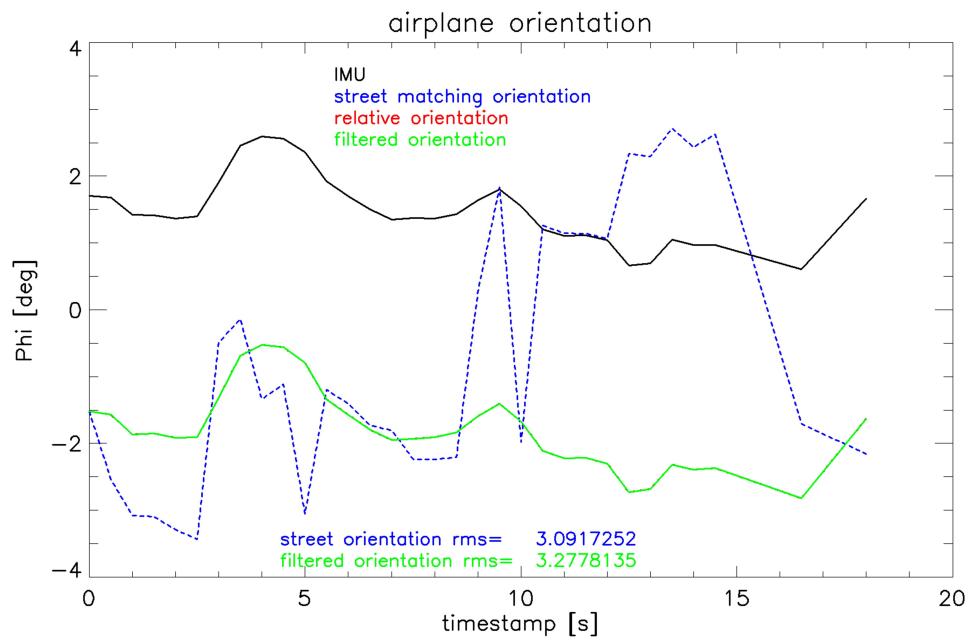


Abbildung 118:  $\phi$ -Winkel des Navigationsfilters (Überflug 8)

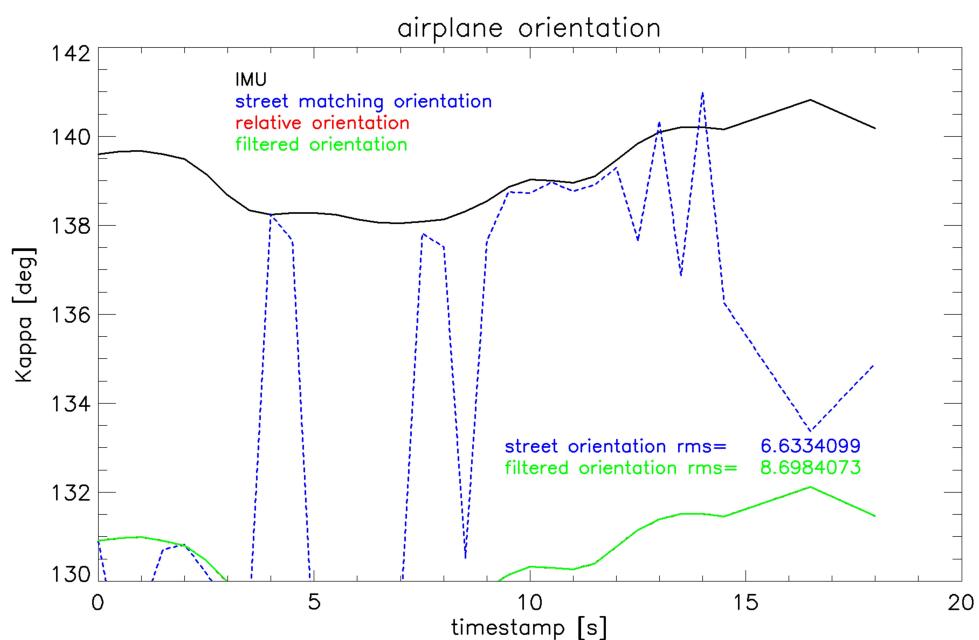


Abbildung 119:  $\kappa$ -Winkel des Navigationsfilters (Überflug 8)

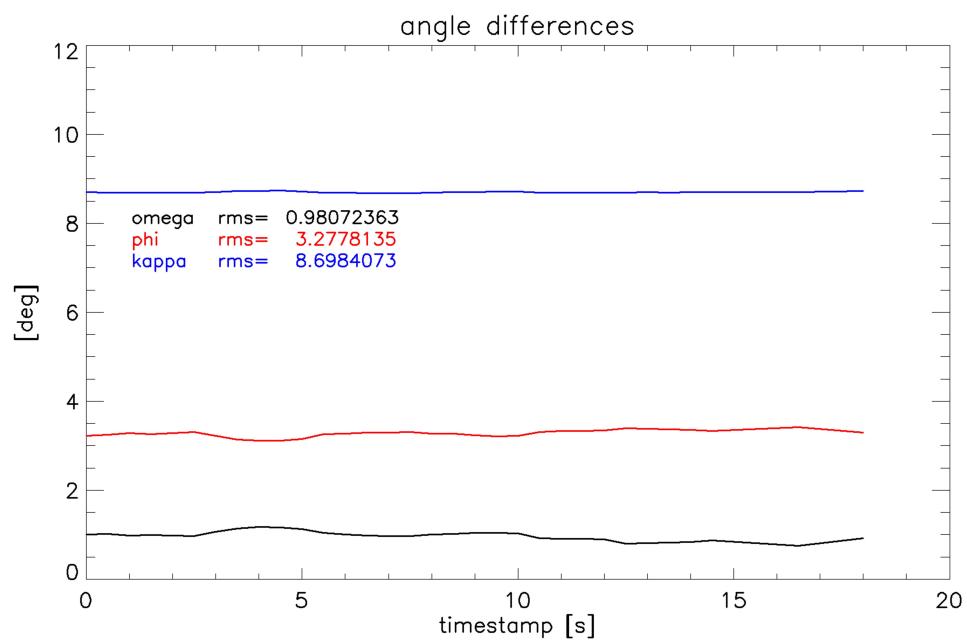


Abbildung 120: Winkelfehler des Navigationsfilters (Überflug 8)

### Überflug 9

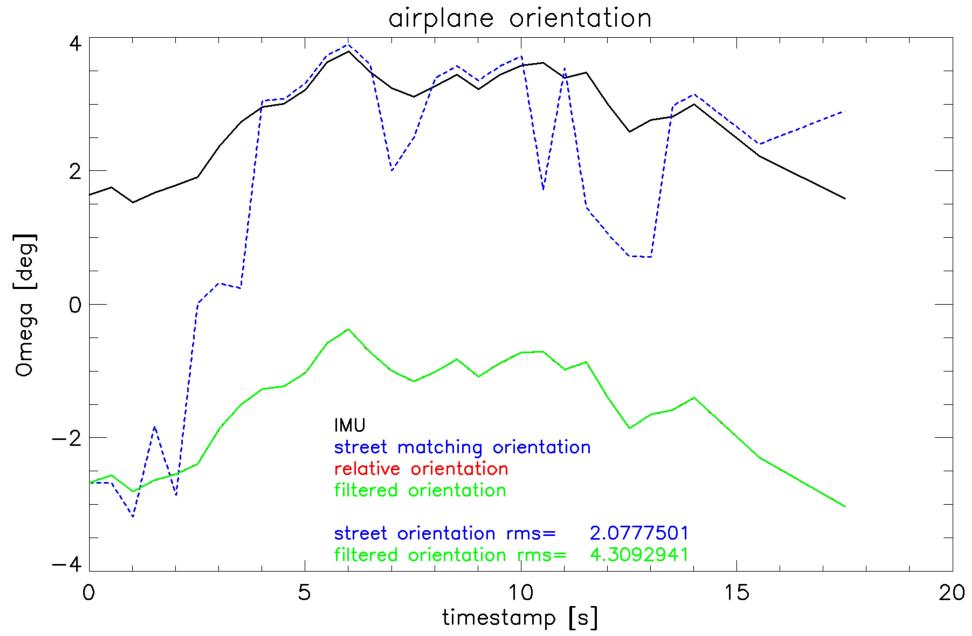


Abbildung 121:  $\omega$ -Winkel des Navigationsfilters (Überflug 9)

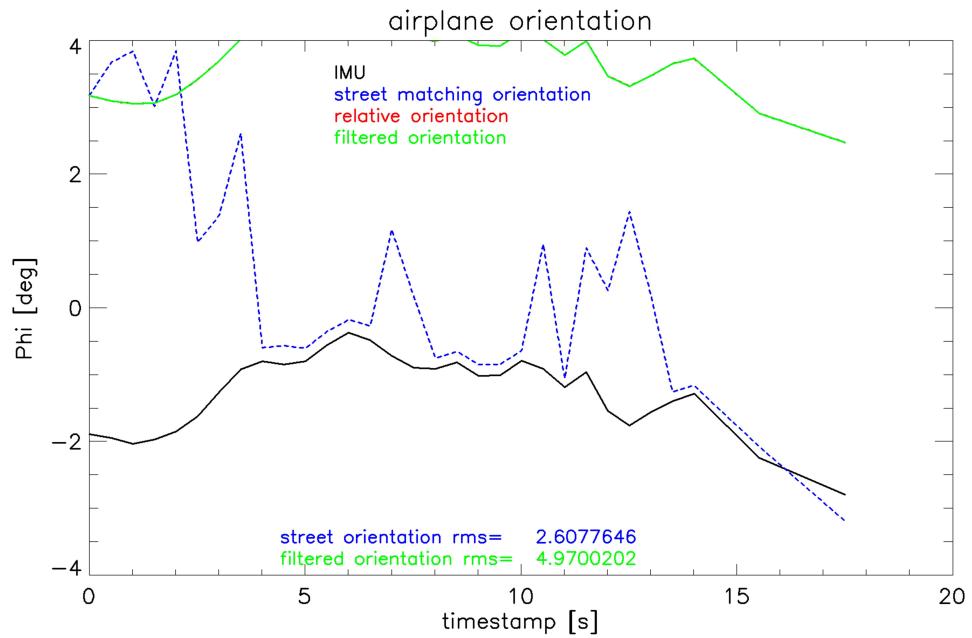


Abbildung 122:  $\phi$ -Winkel des Navigationsfilters (Überflug 9)

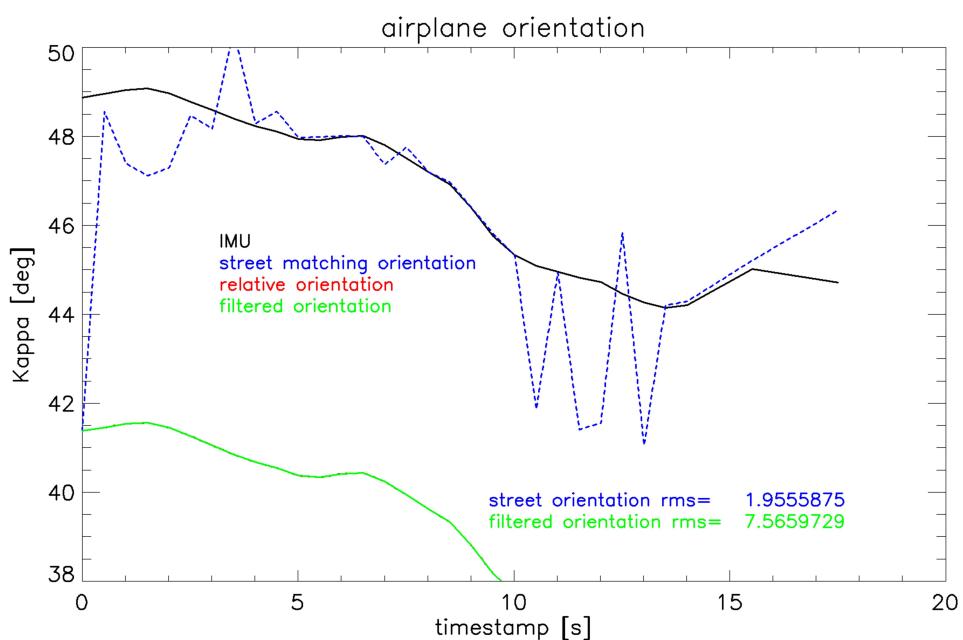


Abbildung 123:  $\kappa$ -Winkel des Navigationsfilters (Überflug 9)

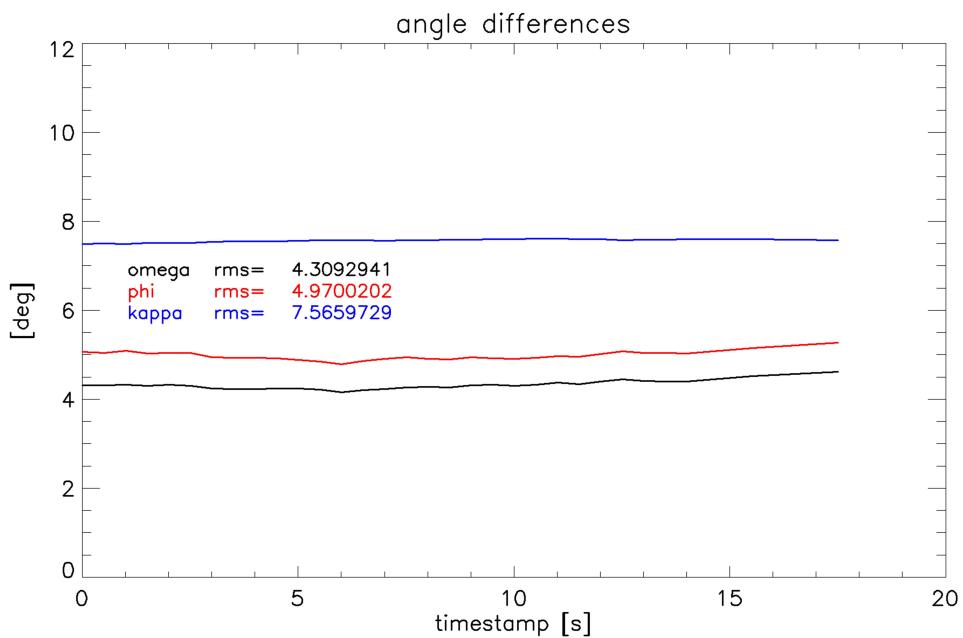


Abbildung 124: Winkelfehler des Navigationsfilters (Überflug 9)

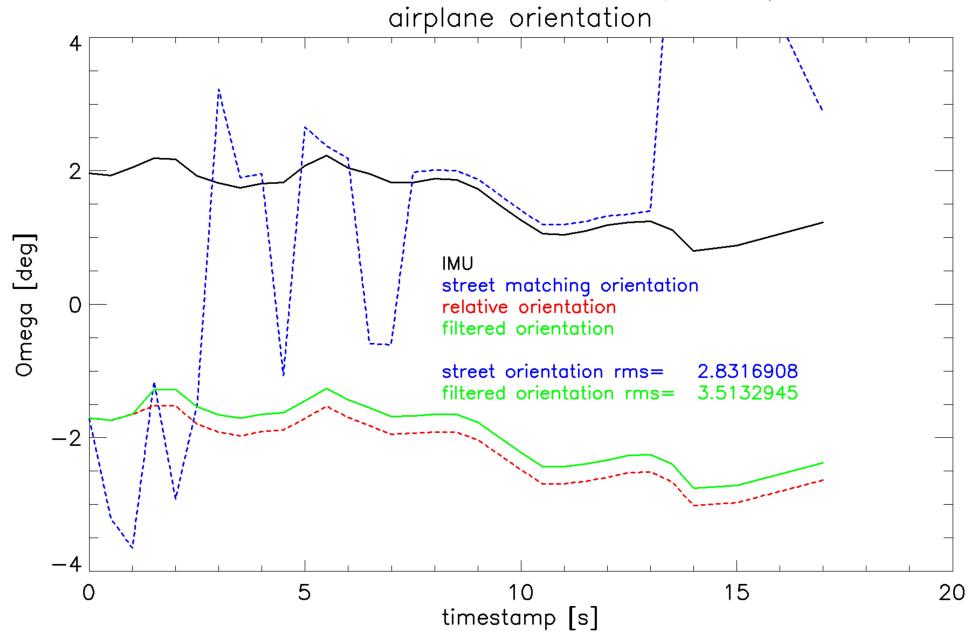


Abbildung 125:  $\omega$ -Winkel des Navigationsfilters (Überflug 10)

### Überflug 10

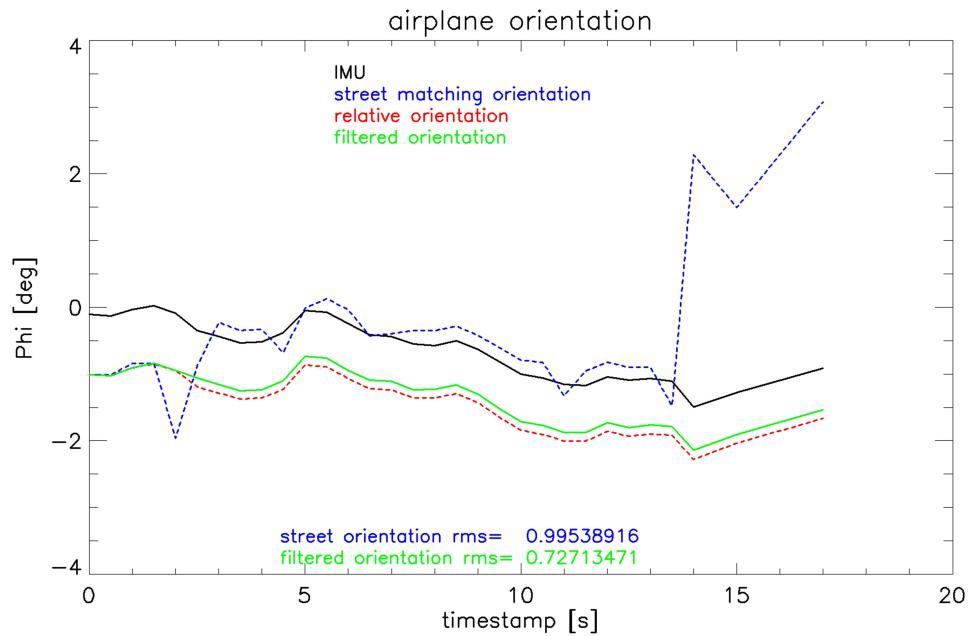


Abbildung 126:  $\phi$ -Winkel des Navigationsfilters (Überflug 10)

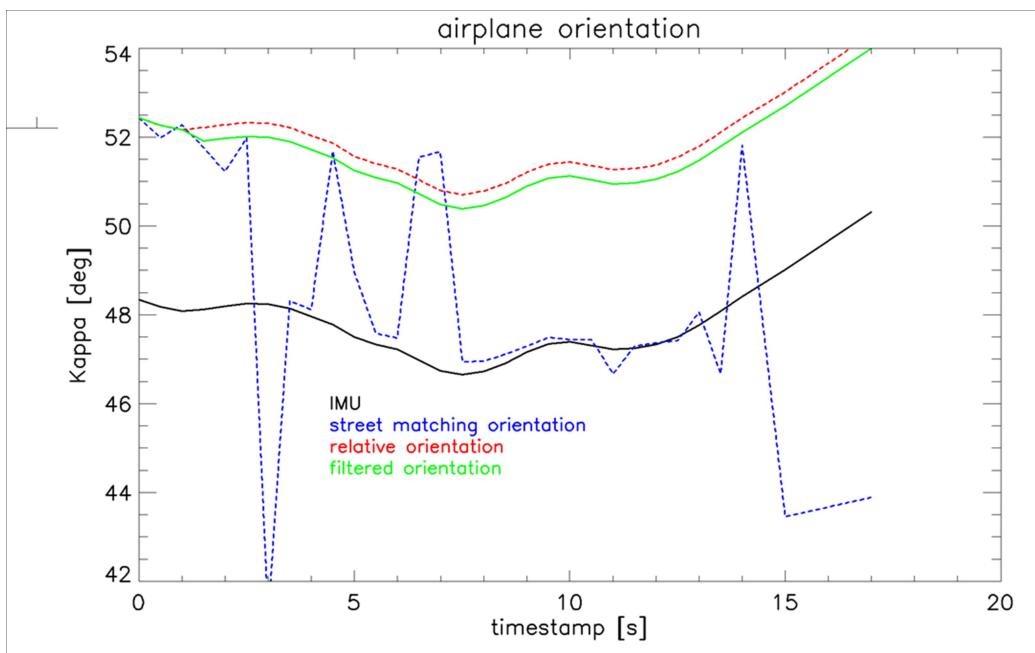


Abbildung 127:  $\kappa$ -Winkel des Navigationsfilters (Überflug 10)

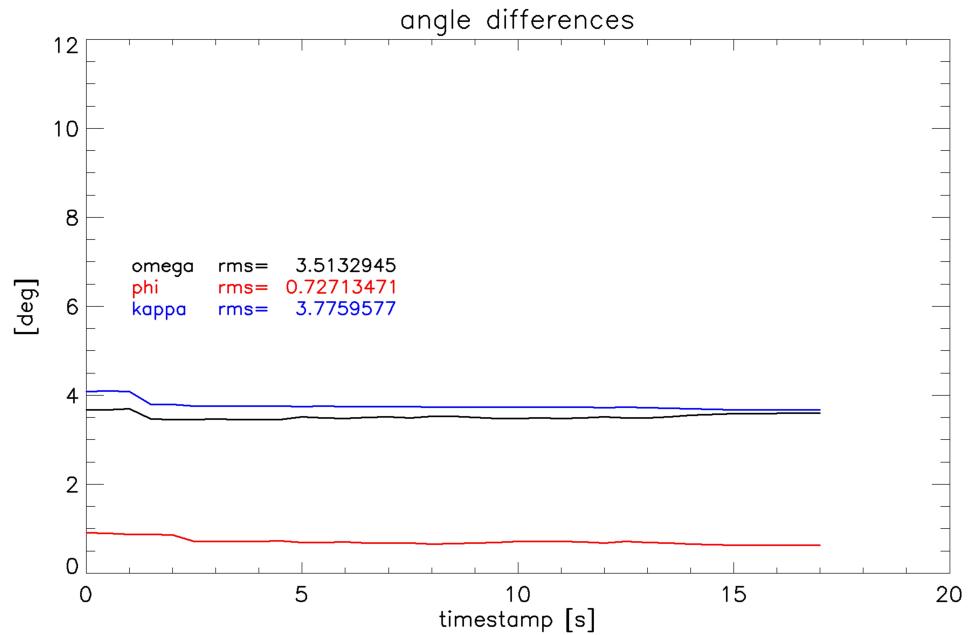


Abbildung 128: Winkelfehler des Navigationsfilters (Überflug 10)

## 8 Zusammenfassung und Ausblick

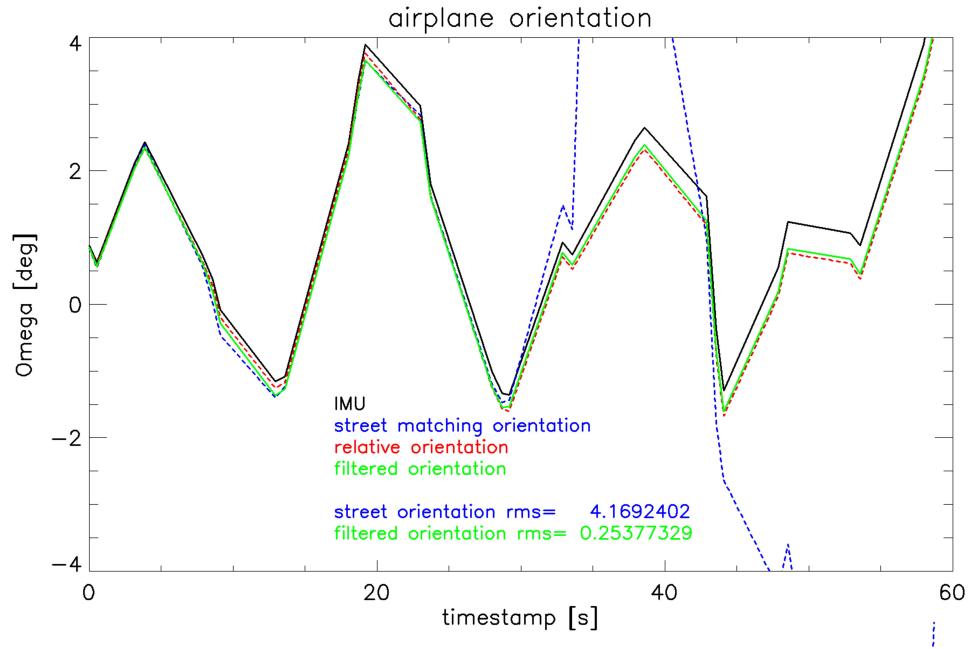


Abbildung 129:  $\omega$ -Winkel des Navigationsfilters (Überflug 11)

### Überflug 11

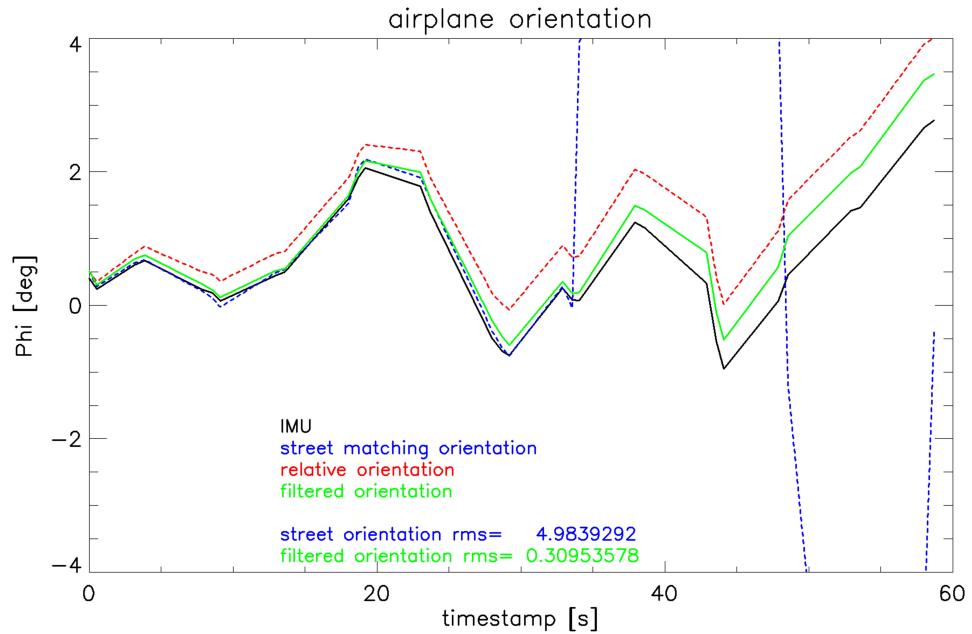


Abbildung 130:  $\phi$ -Winkel des Navigationsfilters (Überflug 11)

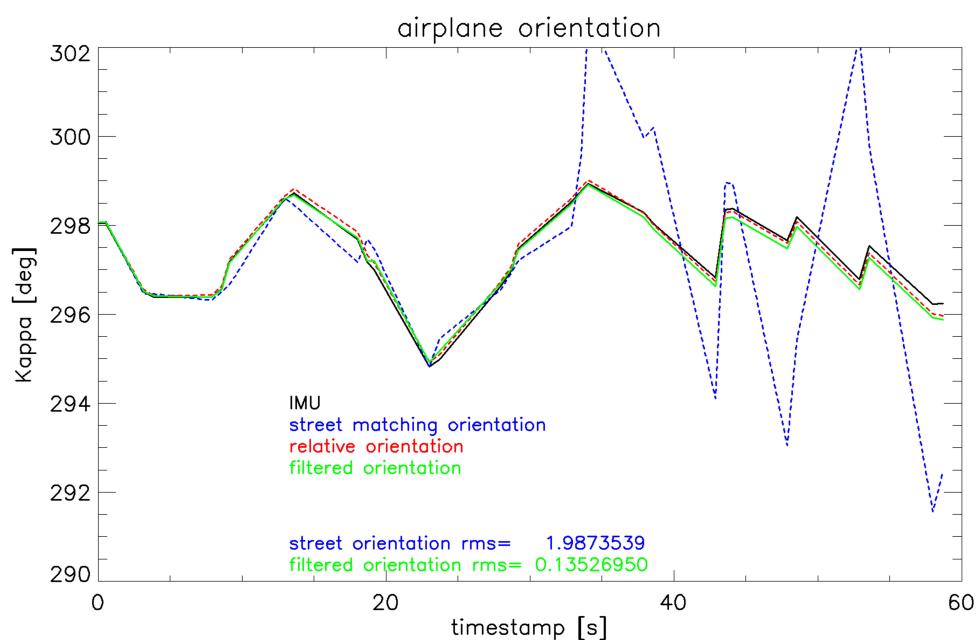


Abbildung 131:  $\kappa$ -Winkel des Navigationsfilters (Überflug 11)

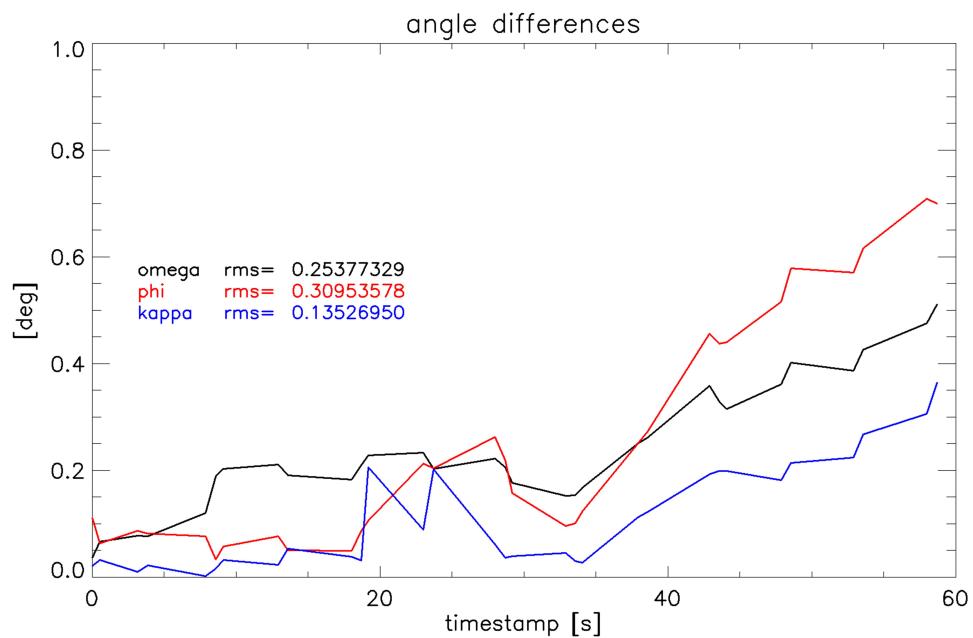


Abbildung 132: Winkelfehler des Navigationsfilters (Überflug 11)

## **RANSAC-korrigierte Ergebnisse des Navigationsfilters**

## Überflug 8

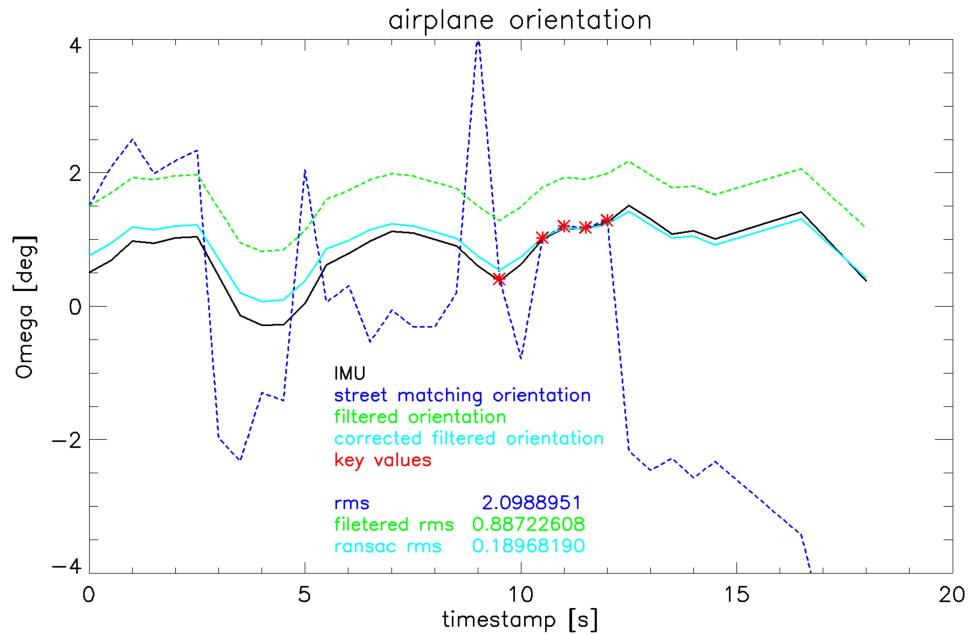


Abbildung 133: RANSAC-korrigierte  $\omega$ -Winkel (Überflug 8)

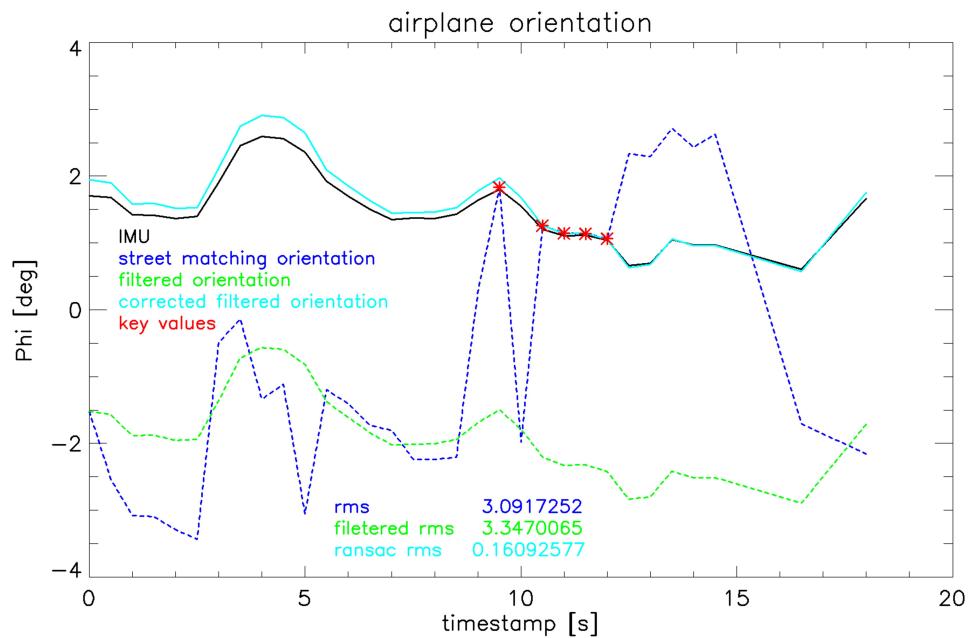


Abbildung 134: RANSAC-korrigierte  $\phi$ -Winkel (Überflug 8)

## 8 Zusammenfassung und Ausblick

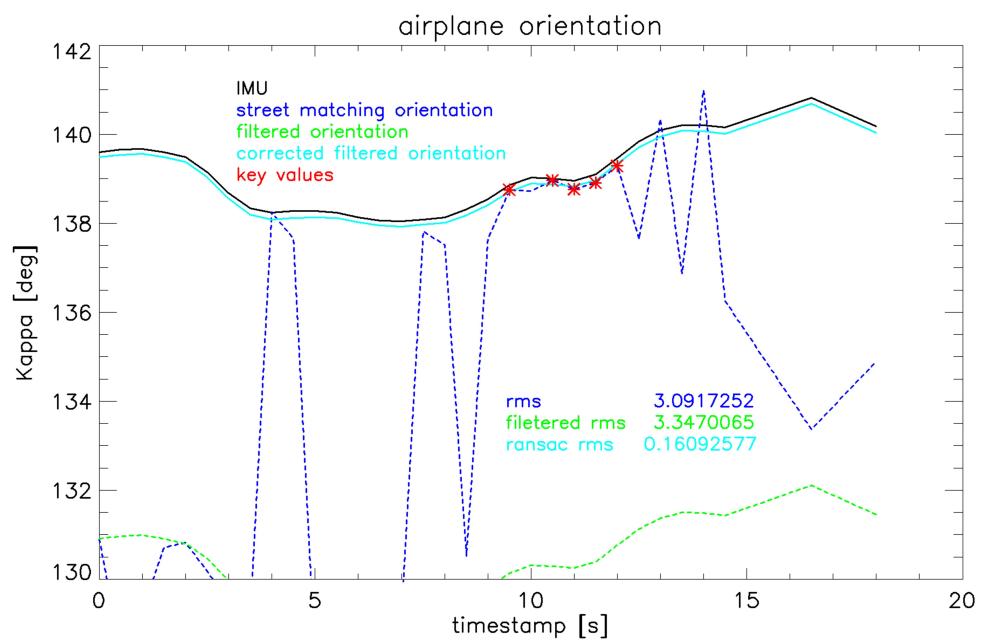


Abbildung 135: RANSAC-korrigierte  $\kappa$ -Winkel (Überflug 8)

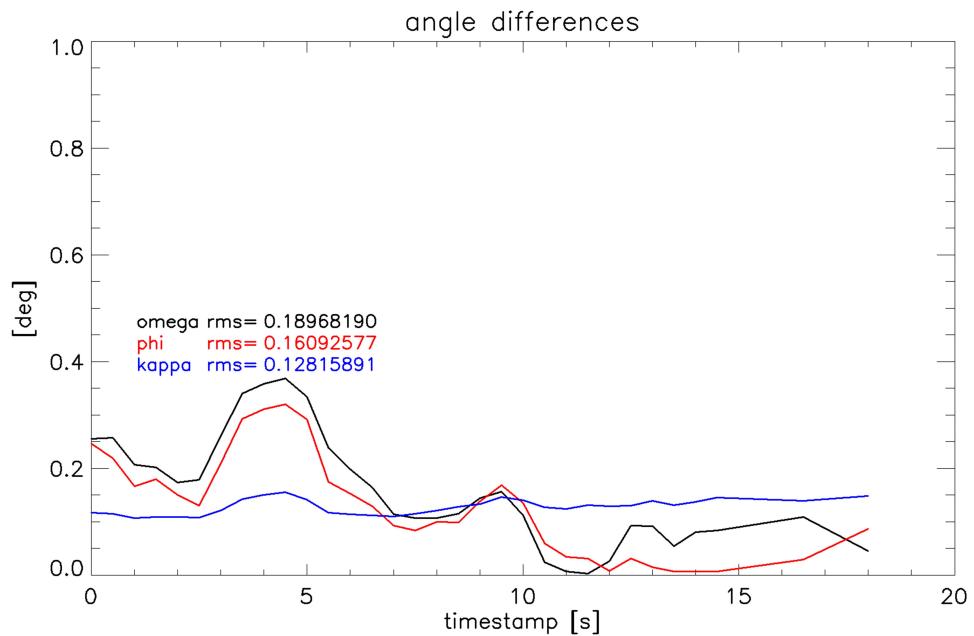


Abbildung 136: RANSAC-korrigierte Winkelfehler (Überflug 8)

## Überflug 9

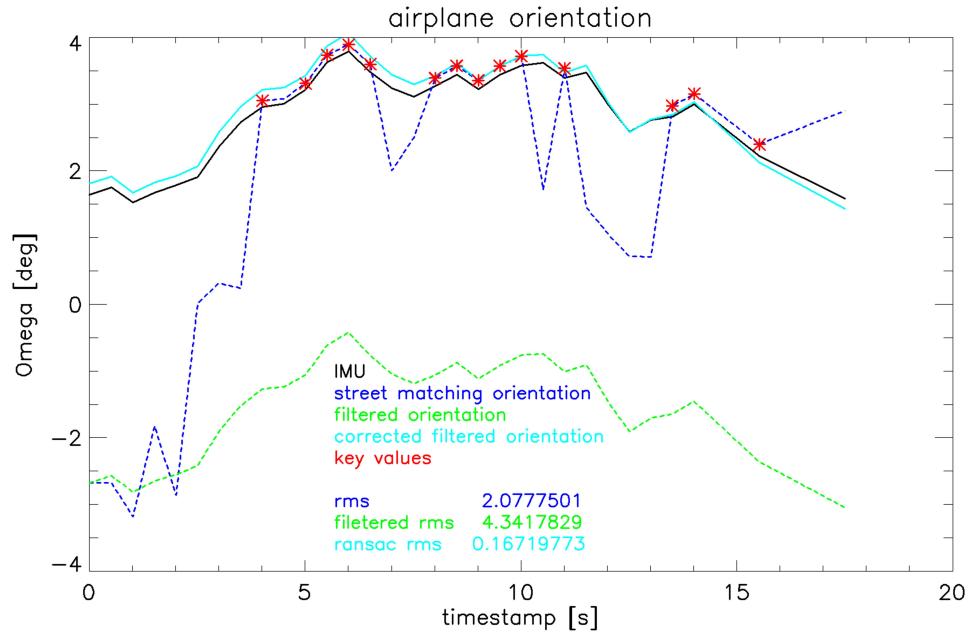


Abbildung 137: RANSAC-korrigierte  $\omega$ -Winkel (Überflug 9)

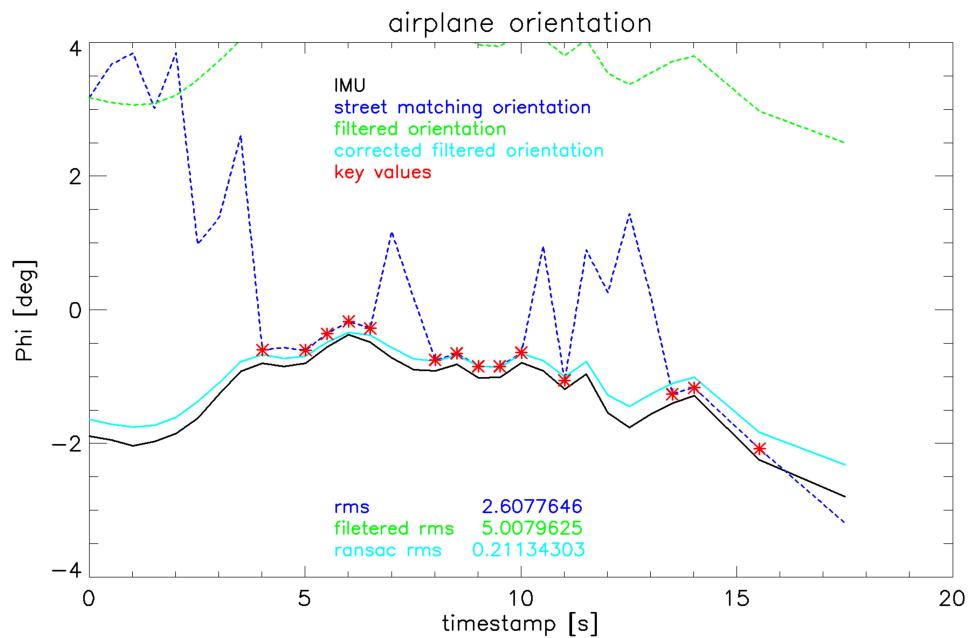


Abbildung 138: RANSAC-korrigierte  $\phi$ -Winkel (Überflug 9)

## 8 Zusammenfassung und Ausblick

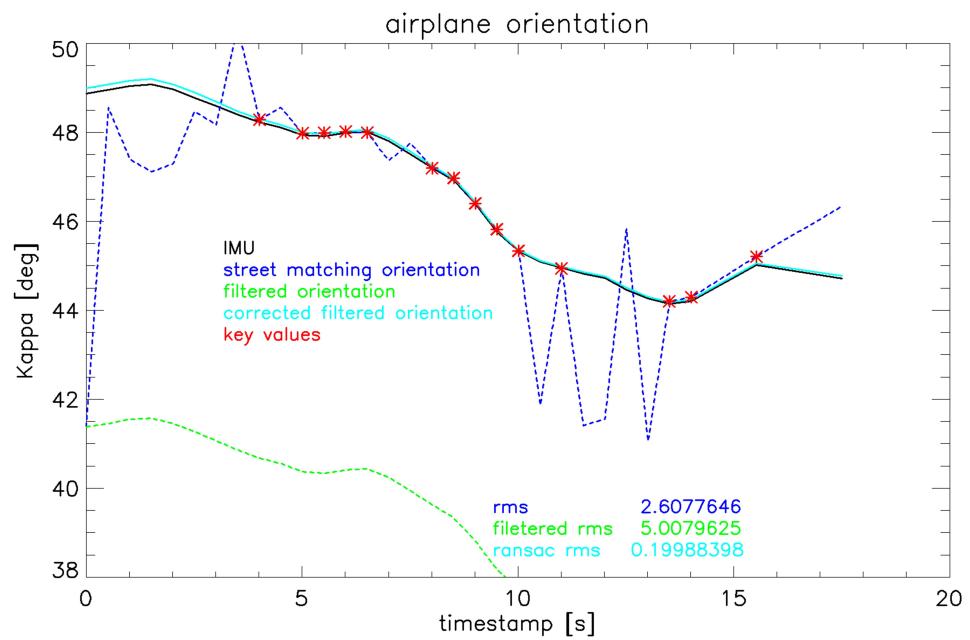


Abbildung 139: RANSAC-korrigierte  $\kappa$ -Winkel (Überflug 9)

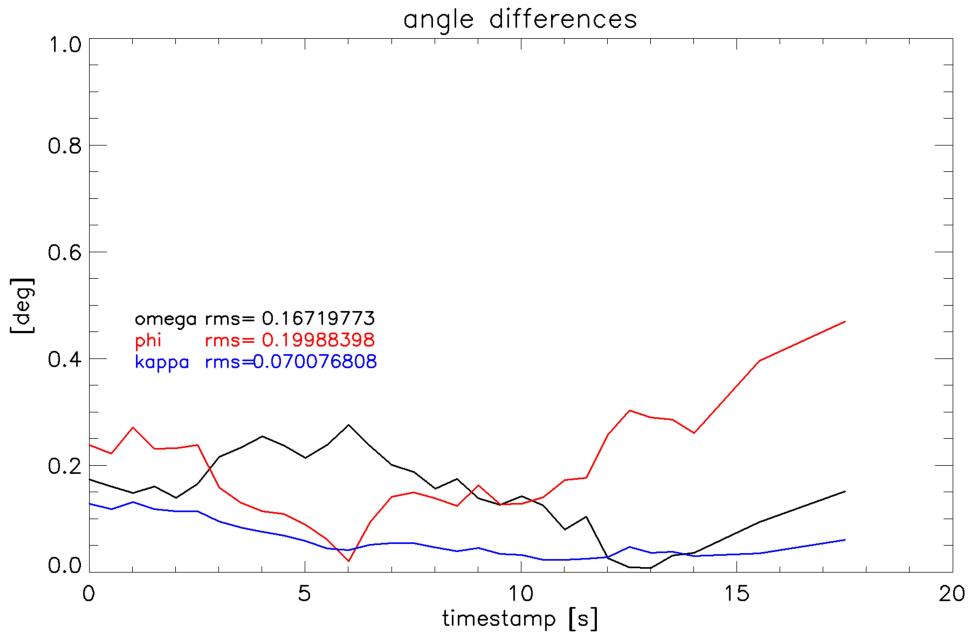


Abbildung 140: RANSAC-korrigierte Winkelfehler (Überflug 9)

## Überflug 10

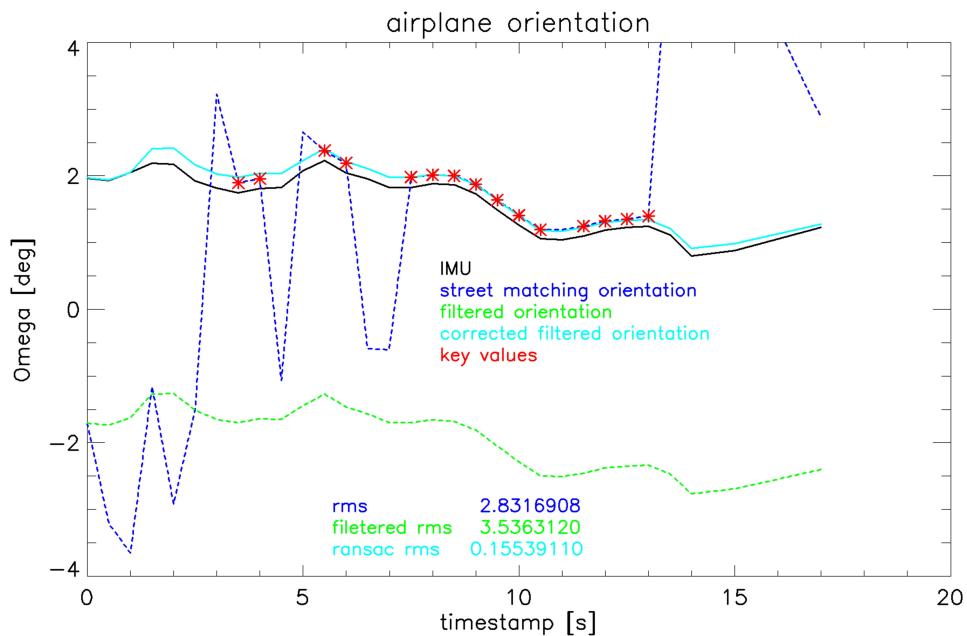


Abbildung 141: RANSAC-korrigierte  $\omega$ -Winkel (Überflug 10)

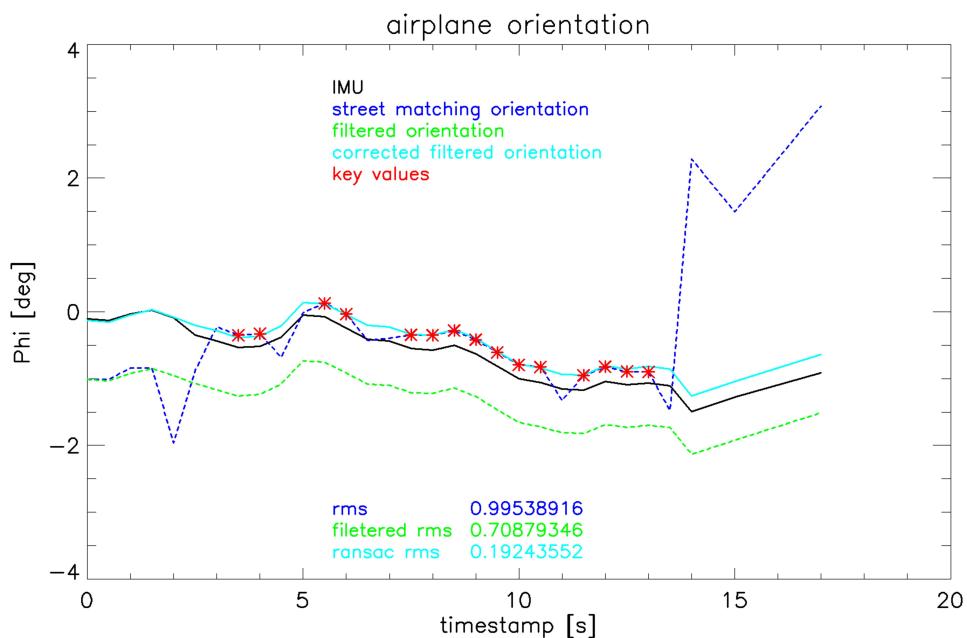


Abbildung 142: RANSAC-korrigierte  $\phi$ -Winkel (Überflug 10)

## 8 Zusammenfassung und Ausblick

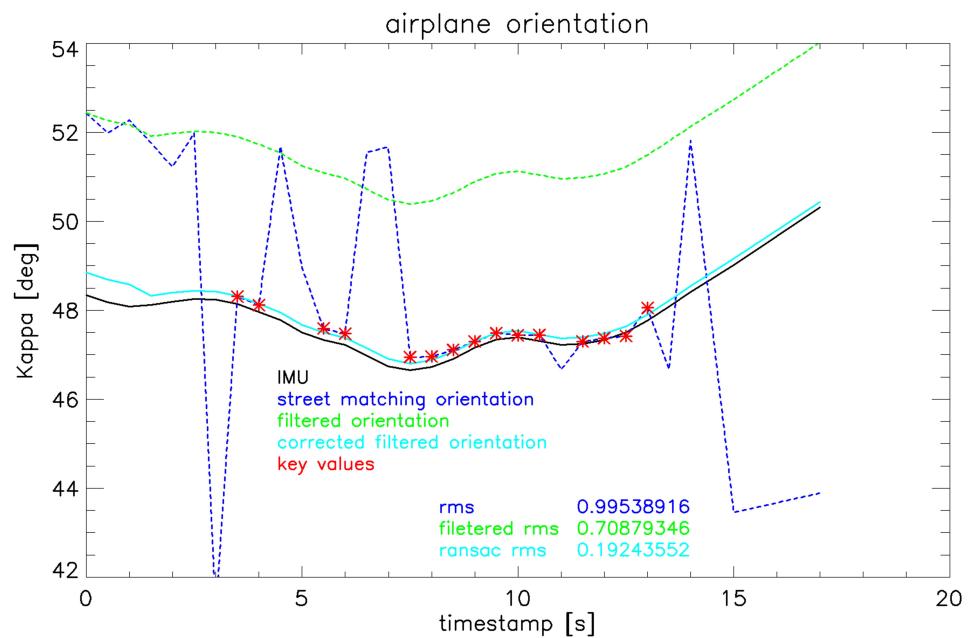


Abbildung 143: RANSAC-korrigierte  $\kappa$ -Winkel (Überflug 10)

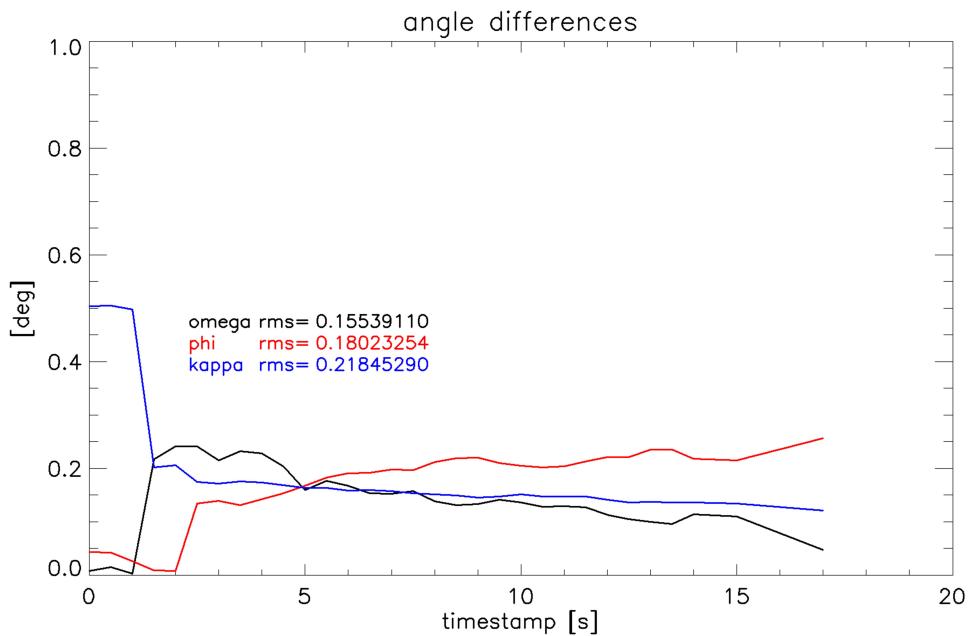


Abbildung 144: RANSAC-korrigierte Winkelfehler (Überflug 10)

# Literaturverzeichnis

- [1] *RAS-Q - Richtlinie für die Anlage von Strassen, Teil: Querschnitte.* Verlag der Forschungsgesellschaft für Straßen- und Verkehrswesen, 1996
- [2] *UTM - Abbildung und Koordinaten.* Bayerisches Landesvermessungsamt, 2002
- [3] Beliebteste Autofarbe. <http://www.autosieger.de/article3948.html>. Version: 10 2004
- [4] Neue beliebteste Autofarben: Silber ist nicht mehr der Champion. [http://www.rp-online.de/auto/news/Silber-ist-nicht-mehr-der-Champion\\_aid\\_521371.html](http://www.rp-online.de/auto/news/Silber-ist-nicht-mehr-der-Champion_aid_521371.html). Version: 01 2008
- [5] arc tangent function of two variables. <http://linux.die.net/man/3/atan2>. Version: 09 2009
- [6] NAVTEQ Maps and Traffic. [www.navteq.com](http://www.navteq.com). Version: 10 2009
- [7] Shuttle Radar Topography Mission. <http://www2.jpl.nasa.gov/srtm/>. Version: 06 2009
- [8] AEROcontrol & AEROoffice - Ingenieur-Gesellschaft für Interfaces mbH. <http://www.igi.eu/aerocontrol.html>. Version: 06 2010
- [9] Color Conversion Algorithms. [http://www.cs.rit.edu/~ncs/color/t\\_convert.html](http://www.cs.rit.edu/~ncs/color/t_convert.html). Version: 04 2010
- [10] GeoEye Products Imagery Source. <http://www.geoeye.com/CorpSite/products/imagery-sources/Default.aspx#ikonos>. Version: 08 2010
- [11] ATKIS - Amtliches Topographisch-Kartographisches Informationssystem. [www.atkis.de](http://www.atkis.de). Version: 02 2011
- [12] BAR-SHALOM, Y. ; LI, X.-R. ; KIRUBARAJAN, T. : *Estimation with Applications to Tracking and Navigation.* Artech House, Norwood, MA, 2001
- [13] BAY, H. ; ANDREAS, E. ; TUYTEL AARS, T. ; VAN GOOL, L. : SURF: Speeded Up Robust Features. In: *Computer Vision and Image Understanding* 110 (2008), S. 346 – 359
- [14] BHASKARAN, S. ; RIEDEL, J. E. ; SYNNOTT, S. P.: Autonomous Optical Navigation for Interplanetary Missions / Navigation and Flight Mechanics Section, Jet Propulsion Laboratory, California Institute of Technology. 1996. – Forschungsbericht

## Literaturverzeichnis

- [15] BÖHRINGER, F. : *Gleisselektive Ortung von Schienenfahrzeugen mit bordautonomer Sensorik*, Universität Karlsruhe, Diss., 2008
- [16] BORGEFORS, G. : Hierachical Chamfer Matching: A Parametric Edge Matching Algorithm. In: *IEEE Transactions on Pattern Analysis and Intelligence* 10 (1988), S. 849–865
- [17] BROWN, D. C.: Close-Range Camera Calibration. In: *Photogrammetric Engineering* 8 (1971), S. 855–866
- [18] BÄUMKER, M. : Kalibration und Umrechnung von INS- und photogrammetrischen Winkeln für beliebige gegenseitige Anordnungen. In: *Proceedings 14. Internationale Geodätische Woche 2007, Obergurgl, Ötztal*, 2007
- [19] BURGES, C. J.: A Tutorial on Support Vector Machines for Pattern Recognition. In: *Data Mining and Knowledge Discovery* 2 (1998), S. 121–167
- [20] CLODE, S. P. ; ROTTENSTEINER, F. ; KOOTSOOKOS, P. : Improving City Model Determination by Using Road Detection from LIDAR Data. In: *International Archives of Photogrammetry and Remote Sensing* XXXVI (2005), S. 159–164
- [21] CRAMER, M. : Performance of GPS/Inertial Solutions in Photogrammetry. In: *Photogrammetrische Woche 2001* (2001), S. 49–62
- [22] CRISTIANINI, N. ; SHawe-Taylor, J. : *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000
- [23] D., L. ; S., H. : Refining Correctness of Vehicle Detection and Tracking in Aerial Image Sequences by Means of Velocity and Tajectory Evaluation. In: *International Archives of Photogrammetry and Remote Sensing* XXXVIII (2009), S. 181–186
- [24] DALAL, N. ; TRIGGS, B. : Histograms of Oriented Gradients for Human Detection. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005
- [25] DESAI, S. ; HAN, D. ; BHASKARAN, S. ; KENNEDY, B. ; MCELRATH, T. ; NULL, G. W. ; RIEDEL, J. E. ; RYNE, M. ; SYNNOTT, S. P. ; WANG, T. C. ; WERNER, R. A. ; ZAMANI, E. B.: Autonomous Optical Navigation (AutoNav) Technology Validation Report / Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California. 2004. – Forschungsbericht
- [26] ERNST, I. ; HETSCHER, M. ; THIESSENHUSEN, K. ; RUHÉ, M. ; BÖRNER, A. ; ZUEV, S. : New approaches for real time traffic data acquisition with airborne systems. In: *International Archives of Photogrammetry and Remote Sensing* XXXVI (2005), S. 69–73

- [27] ERNST, I. ; HETSCHER, M. ; ZUEV, S. ; RUHÉ, M. : Lumos - ein neuartiges System zur luftgestützten Verkehrsdatenerfassung. In: *IMA 2004 Informationssysteme für mobile Anwendungen*, 2004
- [28] ERNST, I. ; SUJEW, S. ; THIESSENHUSEN, K.-U. ; HETSCHER, M. ; RASSMANN, S. ; RUHÉ, M. : LUMOS - Airborne Traffic Monitoring System. In: *IEEE 6th International Conference On Intelligent Transportation Systems*, 2003
- [29] EUGSTER, H. : Georegistrierung von mit Mini- und Mikrodrohnen erfassten Video-sequenzen mit Hilfe von digitalen 3D-Landschaftsmodellen. In: *3D-NordOst 2008 Berlin Tagungsband* (2008), S. 111–120
- [30] EUGSTER, H. : *Echtzeit-Georegistrierung von Videodaten mit Hilfe von Navigationssensoren geringer Qualität und digitalen 3D-Landschaftsmodellen*, Humboldt-Universität zu Berlin, Diss., 2011
- [31] EUGSTER, H. ; NEBIKER, S. : Real-Time Georegistration of Video Streams from Mini or Micro UAS Using Digital 3D City Models. In: *6th International Symposium on Mobile Mapping Technology*, 2009
- [32] FISCHLER, M. A. ; TENENBAUM, J. M. ; WOLF, H. C.: Detection of Roads and Linear Structures in Low-Resolution Aerial Imagery Using a Multisource Knowledge Integration Technique. In: *Computer Graphics and Image Processing* 15 (1981), S. 201–223
- [33] FISCHLER, M. A. ; BOLLES, R. C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. In: *Communications of ACM* 24 (1981), S. 381–395
- [34] FREUD, Y. ; SCHAPIRE, R. E.: A Short Introduction to Boosting. In: *Journal of Japanese Society for Artificial Intelligenz* 14 (5) (1999), S. 771–780
- [35] GRABNER, H. ; NGUYEN, T. T. ; GRUBER, B. ; BISCHOF, H. : On-line boosting-based car detection from aerial images. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 63 (2008), S. 382–396
- [36] GREINER, T. : *RFID in der Verkehrstelematik*. Grin Verlag, 2009
- [37] GRENZDÖRFFER, G. ; ZUEV, S. : Bestimmung des photogrammetrischen Genauigkeitspotentials des Online-Systems ANTAR zur luftgestützten Verkehrsdatenerfassung / Universität Rostock, Institut für Management ländlicher Räume Deutsches Zentrum für Luft- und Raumfahrt (DLR), Optische Informationssysteme. 2007. – Forschungsbericht
- [38] GRIESSBACH, D. ; BAUMBACH, D. ; ZUEV, S. : Vision aided inertial navigation. In: *EuroCOW*, 2010

## Literaturverzeichnis

- [39] HAAG, M. ; NAGEL, H.-H. : Combination of Edge Element and Optical Flow Estimates for 3D-Model-Based Vehicle Tracking in Traffic Image Sequences. In: *International Journal of Computer Vision* 35 (1999), S. 295–319
- [40] HARTUNG, J. ; ELPELT, B. ; KLÖSENER, K.-H. ; OLDENBOURG (Hrsg.): *Statistik*. Oldenbourg, 1999
- [41] HAUSBURG, M. : *Validierung von Objekthypothesen in ARGOS-Luftbildern*, Technische Universität Berlin, Diplomarbeit, 2010
- [42] HELBING, D. : *Verkehrs dynamik. Neue dynamische Modellierungskonzepte*. Springer Verlag Berlin Heidelberg, 1997
- [43] HINZ, S. : Automatische Fahrzeugerkennung in optischen Bilddaten: Ein Überblick mit Konzeption eines neuen Verfahrens / Chair for Photogrammetry and Remote Sensing Technische Universität München. 2002. – Forschungsbericht
- [44] HINZ, S. : *Automatische Extraktion urbaner Straßennetze aus Luftbildern*, Technische Universität München, Diss., 2003
- [45] HINZ, S. ; KURZ, F. ; WEIHING, D. ; SUCHAND, S. ; MEYER, F. ; BAMLER, R. : Evaluation of Traffic Monitoring based on Spatio-Temporal Co-Registration of SAR Data and Optical Image Sequences. In: *Photogrammetrie Fernerkundung Geoinformation* 5 (2007), S. 309–325
- [46] HUANG, J. ; KONG, B. ; LI, B. ; ZHENG, F. : A New Method of Unstructured Road Detection Based on HSV Color Space and Road Features. In: *Proceedings of the 2007 International Conference on Information Acquisition* (2007), S. 596–601
- [47] HUANG, M. ; AINE, C. ; SUPEK, S. ; BEST, E. ; RANKEN, D. ; FLYNN, E. : Multi-start downhill simplex method for spatio-temporal source localization in magnetoencephalography. In: *Electroencephalography and clinical Neurophysiology* 108 (1998), S. 32–44
- [48] HUTTON, J. : Computation of Phi Omage Kappa from Roll Pitch Heading / Applanix - Applied Analytics Corporation. 1997. – Forschungsbericht
- [49] JACOBSEN, K. : Calibration aspects in direct georeferencing of frame imagery. In: *ISPRS Commission I / Pecora 15 conference proceedings* 34 (2002), S. 82–89
- [50] JUNGHANS, M. : Diskretes Kalman-Filter (KF) / TU Dresden. 2007. – Forschungsbericht
- [51] KALMAN, R. E.: A New Approach to Linear Filtering and Prediction Problems. In: *Transactions of the ASME - Journal of Basic Engineering* 86 (1960), S. 35–45
- [52] KANNALA, J. ; S., B. : A Generic Camera Model and Calibration Method for Conventional, Wide-Angle, and Fish-Eye Lenses. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (2006), S. 1335–1340

- [53] KÜHNE, R. ; RUHÉ, M. ; BONERT, M. : Anwendung der luftgestützten Verkehrsdatenerfassung bei Großveranstaltungen. In: *Verkehrswissenschaftliche Tage*, 2005
- [54] KLEEMAN, L. : Understanding and Applying Kalman Filtering / Department of Electrical and Computer Systems Engineering. 2002. – Forschungsbericht
- [55] KLEIN, L. A.: *Sensor technologies and data requirements for ITS*. Artech House, 2001
- [56] KOCH, A. : *Konzept und Implementierung eines Kalman-Filters zur Navigation eines VTOL-UAVs mit Hilfe optischer Sensoren*, Technical University Braunschweig, Diplomarbeit, 2005
- [57] KOZEMPEL, K. : *Untersuchung verschiedener Matchingverfahren und Weiterentwicklung eines dieser Verfahren unter Berücksichtigung der spezifischen Struktur der zu untersuchenden Objekte*, Brandenburgische Technische Universität Cottbus, Diplomarbeit, 2007
- [58] KOZEMPEL, K. ; REULKE, R. : Camera Orientation Based on Matching Road Networks. In: *Image and Vision Computing New Zealand*, 2009
- [59] KOZEMPEL, K. ; REULKE, R. : Fast Vehicle Detection and Tracking in Aerial Image Bursts. In: *International Archives of Photogrammetry and Remote Sensing XXXVIII* (2009), S. 175–180
- [60] KRAETZSCHMAR, G. K. ; ENDERLE, S. : Self-localization using sporadic features. In: *Robotics and Autonomous Systems* 40 (2002), Nr. 2-3, 111 - 119. [http://dx.doi.org/10.1016/S0921-8890\(02\)00236-1](http://dx.doi.org/10.1016/S0921-8890(02)00236-1). – DOI DOI: 10.1016/S0921-8890(02)00236-1. – ISSN 0921-8890
- [61] KRAUSS, K. ; KG, H. . C. G. . C. (Hrsg.): *Photogrammetrie Band 1, Geometrische Informationen aus Photographien und Laserscanneraufnahmen*. De Gruyter Lehrbuch, 2004
- [62] KRUCK, E. : Rotations of space and coordinate transformations. In: *Proceedings ISPRS WG I/5 workshop on theory, technology and realities of inertial/GPS sensor orientation*, 2003
- [63] KURZ, F. ; MÜLLER, R. ; STEPHANI, M. ; REINARTZ, P. ; SCHROEDER, M. : Calibration of a Wide-Angle Digital Camera System for Near Real Time Scenarios. In: *ISPRS Hannover Workshop 2007, High Resolution Earth Imaging for Geospatial Information*, 2007
- [64] LEI, Z. ; LI, D. ; FANG, T. : Vehicle Detection in High-Resolution Satellite Imagery using SIFT Features and Support Vector Machine. In: *ISPRS Journal of Photogrammetry and Remote Sensing* (2008)
- [65] LEICH, A. : *Ein Beitrag zur Realisierung der videobasierten weiträumigen Verkehrsbeobachtung*, Technische Universität Dresden, Diss., 2006

## Literaturverzeichnis

- [66] LEITLOFF, J. ; HINZ, S. ; STILLA, U. : Vehicle Detection in Very High Resolution Satellite Images of City Areas. In: *IEEE Transactions on Geoscience and Remote Sensing* 48 (2010), S. 2795–2806
- [67] LENHART, D. ; HINZ, S. : Automatic Vehicle Tracking in Low Frame Rate Aerial Image Sequences. (2008). <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=?doi=10.1.1.109.2412>
- [68] LOWE, D. G.: Distinctive Image Features from Scale-Invariant Keypoints. In: *International Journal of Computer Vision* 60 (2004), S. 91–110
- [69] LUBER, A. ; REULKE, R. : A unified calibration approach for generic cameras. In: *International Archives of Photogrammetry and Remote Sensing and Spatial Information Sciences*, 2010
- [70] LUHMANN, T. ; KG, J. P. H. G. . C. (Hrsg.): *Nahbereichsphotogrammetrie*. 2. Herbert Wichmann Verlag, 2003
- [71] MARKOWETZ, F. : Klassifikation mit Support Vector Machines / Max-Planck-Institut für Molekulare Genetik, Berlin Center for Genome Based Bioinformatics. 2003. – Forschungsbericht
- [72] MIKOLAJCZYK, K. ; SCHMID, C. : A performance evaluation of local descriptors / Department of Engineering Science, University of Oxford, UK. 2005. – Forschungsbericht
- [73] MÜLLER, R. ; LEHNER, M. ; MÜLLER, R. ; REINARTZ, P. ; SCHROEDER, M. ; B.VOLLMER: A Program for Direct Georeferencing of Airborne and Spaceborne Line Scanner Images. In: *Integrating Remote Sensing at the Global, Regional and Local Scale*, 2002
- [74] MOIK, J. G.: *Digital Processing of Remotely Sensed Images*. NASA, 1980
- [75] MOON, H. ; CHELLAPPA, R. ; ROSENFIELD, A. : Perfomance analysis of a simple vehicle detection algorithm. In: *Image and Vision Computing* 20 (2002), S. 1–13
- [76] MOSTAFA, D. M. ; HUTTON, J. ; LITHOPOULOS, E. : Ground Accuracy from Directly Georeferenced Imagery / Applanix Corporation. 2000. – Forschungsbericht
- [77] MOSTAFA, M. ; HUTON, J. ; REID, B. ; HILL, R. : GPS/IMU Products - the Applanix approach. In: *Photogrammetric Week*, 2001
- [78] NEJADASL, F. K. ; GORTE, B. G. ; HOGENDOORN, S. P.: Optical flow based vehicle tracking strengthened by statistical decisions. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 61 (2006), S. 159–169
- [79] NELDER, J. A. ; MEAD, R. : A Simplex Method for Function Minimization. In: *Computer Journal* 7 (1965), S. 308–313

- [80] NEUBECK, A. ; VAN GOOL, L. : Efficient Non-Maximum Suppression. In: *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*. Washington, DC, USA : IEEE Computer Society, 2006. – ISBN 0–7695–2521–0, S. 850–855
- [81] PILU, M. : Uncalibrated Stereo Correspondence by Singular Value Decomposition. In: *Computer Vision and Pattern Recognition*, 1997
- [82] PRECHTEL, N. ; BRINGMANN, O. : Near-Real-Time Road Extraction from Satellite Images Using Vector Reference Data. In: *International Archives of Photogrammetry and Remote Sensing* XXXII (1998), S. 229–234
- [83] RAJAGOPALAN, A. ; BURLINA, P. ; CHELLAPPA, R. : Higher Order Statistical Learning for Vehicle Detection in Images. In: *International Conference on Computer Vision* (1999)
- [84] REINARTZ, P. ; KRAUSS, T. ; PÖTZSCH, M. ; RUNGE, H. ; ZUEV, S. : Traffic Monitoring with Serial Image from Airborne Cameras. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 61 (2006), S. 149–158
- [85] ROSENBAUM, D. ; KURZ, F. ; THOMAS, U. ; SURI, S. ; REINARTZ, P. : Towards automatic near real-time traffic monitoring with an airborne wide angle camera system. In: *European Conference of Transport Research Institutes*, 2008
- [86] SANDAU, R. : *Digital Airborne Camera: Introduction and Technology*. Springer, 2009
- [87] SAVII, G. G.: Camera Calibration Using Compound Genetic-Simplex Algorithm. In: *Journal of Optoelectronics and Advanced Materials* 6 (2004), S. 1255–1261
- [88] SCHNABEL, W. ; LOHSE, D. : Grundlagen der Straßenverkehrstechnik und der Verkehrsplanung / Forschungsgesellschaft für Straßen- und Verkehrswesen e.V. 1997. – Forschungsbericht
- [89] SCHWARZ, K.-P. : Aircraft Position and Attitude Determination by GPS and INS. In: *International Archives of Photogrammetry and Remote Sensing* XXXI (1996), S. 67–73
- [90] SCOTT, G. L. ; LONGUET-HIGGINS, H. C.: An algorithm for associating the features of two images. In: *Proceedings of the Royal society of London. Series B. Biological sciences*, 1991
- [91] SHAWE-TAYLOR, J. ; CRISTIANINI, N. : *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004
- [92] SKALOUD, J. : Problems in Direct-Georeferencing by INS/DGPS in the Airborne Environment. In: *ISPRS Workshop on 'Direct versus Indirect Methods of Sensor Orientation'*, 1999

## Literaturverzeichnis

- [93] SPANGENBERG, R. : *Objektverfolgung mit Partikelfiltern*, Humboldt-Universität zu Berlin, Diplomarbeit, 2007
- [94] STANKUTE, S. : *Entwicklung und Implementierung von Algorithmen für ein automatisiertes Verfahren zur Zusammenführung von Parametern aus Geodatenbanken unter besonderer Beachtung von Unschärfen in der Georeferenzierung*, Universität Potsdam, Mathematisch-Naturwissenschaftliche Fakultät, Institut für Geographie, Abteilung Geoinformatik, Diplomarbeit, 2007
- [95] STRELOW, D. ; SINGH, S. : Optimal motion estimation from visual and inertial measurements. In: *Sixth IEEE Workshop on Applications of Computer Vision*, 2002
- [96] THOMAS, U. ; KURZ, F. ; ROSENBAUM, D. ; MUELLER, R. ; REINARTZ, P. : GPU-Based Orthorectification of Digital Airborne Camera Images in Real Time. In: *ISPRS Congress Beijing 2008*, 2008
- [97] TITTERTON, D. H. ; WESTON, J. L. ; NEWGEN IMAGING SYSTEMS (P) LTD., I. Chennai (Hrsg.): *Strapdown inertial navigation technology*. 2. The Institution of Electrical Engineers, 2004
- [98] TUERMER, S. ; LEITLOFF, J. ; REINARTZ, P. ; STILLA, U. : Automatic Vehicle Detection in Aerial Image Sequences of Urban Areas Using 3D HOG Features. In: *Photogrammetric Computer Vision and Image Analysis*, 2010
- [99] VALGREN, C. ; LILIENTHAL, A. : SIFT, SURF and Seasons: Long-term Outdoor Localization Using Local Features. In: *3rd European Conference on Mobile Robots*, 2007
- [100] VIOLA, P. ; JONES, M. : Rapid Object Detection using a Boosted Cascade of Simple Features. In: *Accepted Conference on Computer Vision and Pattern Recognition*, 2001
- [101] WELCH, G. ; BISHOP, G. : An Introduction to the Kalman Filter. Chapel Hill, NC, USA : University of North Carolina at Chapel Hill, 1995. – Forschungsbericht
- [102] WU, A. D. ; JOHNSON, E. N. ; PROCTOR, A. A.: Vision-Aided Inertial Navigation for Flight Control. In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005
- [103] ZHAO, T. ; NEVATIA, R. : Car detection in low resolution aerial images. In: *Image and Vision Computing* 21 (2003), S. 693–703

# **Selbständigkeitserklärung**

Ich erkläre, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Berlin, den 28.01.2008

Karsten Kozempel