

# BILDBASIERTE ZIELANFLUGSREGELUNG EINES UNBEMANNTEN QUADKOPTERS

M. Popp, J. Seibold, G.F. Trommer  
Institute of Systems Optimization (ITE),  
Karlsruhe Institute of Technology (KIT), Fritz-Haber-Weg 1, 76131 Karlsruhe, Germany

## Zusammenfassung

In dieser Arbeit wird ein Verfahren vorgestellt, das einen automatischen Gebäudeanflug unabhängig vom Vorhandensein gültiger GPS-Messungen ermöglicht. Dazu wird zunächst ein Ziel, in diesem Fall ein Gebäudefenster, das bei der Missionsplanung selektiert wurde, mittels Bildverarbeitungsalgorithmen identifiziert und im Verlauf des Anfluges verfolgt. Ein hierfür entwickeltes Flugführungsmodul, bestehend aus einer bildbasierten Regelung und einer Geschwindigkeitsregelung manövriert das Fluggerät auf eine vordefinierte Zielposition, ohne dabei das Ziel aus dem Sichtbereich zu verlieren. Um eine präzise Regelung auch bei GPS-Ausfall zu gewährleisten, wird die inertielle Navigation durch Messungen von Positionsänderungen gestützt, welche mit Hilfe von Posenschätzungen bestimmt werden.

## 1. EINFÜHRUNG

Unter den unbemannten Fluggeräten gewinnen schwebefähige MAVs (Micro Air Vehicles) immer mehr an Bedeutung. Auf Grund ihrer schnellen Einsatzfähigkeit und hohen Wendigkeit sind diese Flugsysteme prädestiniert für kurze, schnell durchführbare Einsätze in urbanem Gebiet. So lassen sie sich beispielsweise zur Unterstützung von Rettungs- und Einsatzkräften in Aufklärungsmissionen einsetzen.

Die in diesen autonomen Fluggeräten eingesetzten inertialen Navigationssysteme verwenden GPS-Informationen, um ein Wegdriften der Navigationslösung zu verhindern und diese langfristig genau zu stützen. Insbesondere in der Nähe von Gebäuden stehen jedoch GPS-Informationen auf Grund von Abschattung und Mehrwegeausbreitungen nicht immer zur Verfügung oder weisen eine nur sehr geringe Güte auf. Um in diesem Fall weiterhin über eine langzeitgenaue Navigationslösung zu verfügen, müssen alternative, in diesen Situationen noch verfügbare Informationen genutzt werden.

Auf Grund des geringen Gewichts und der kleinen Abmessung moderner Kameras werden für diesen Zweck häufig Bildinformationen verwendet. So lassen sich Navigationsfilter bei GPS-Ausfällen mit Hilfe georeferenzierter Luftaufnahmen stützen. In [1] wird zum Einen der optische Fluss aus Bildern einer auf den Boden gerichteten Kamera zur Geschwindigkeitsstützung verwendet. Außerdem werden diese Bilder mit georeferenzierten Luftbildern aus einer Datenbank verglichen. Aus korrekten Zuordnungen werden Informationen über die aktuelle Position gewonnen.

Anstelle von zweidimensionalen Bildern werden in [2] Stereobilder zur Messung der Position und Geschwindigkeit eines Quadkopters genutzt.

Eine Möglichkeit zur Orientierung innerhalb von Gebäuden besteht in der Ausnutzung perspektivischer Bildinformationen. So lassen sich beispielsweise Fluchtpunkte im Bild detektieren und zur Orientierung verwenden [3]. Zusätzlich werden Informationen über die

Umgebungsstruktur durch Umgebungsklassifizierung gewonnen [4]. Ein simultaneous localization and mapping Algorithmus (SLAM) wird in [5] zur Lokalisierung und Umgebungskartierung in Innenräumen eingesetzt.

Neben der Generierung von Positions- und Geschwindigkeitsinformationen lassen sich Bildverarbeitungssysteme auch dazu nutzen, autonome Fluggeräte zu stabilisieren und zu regeln. Sogenannte visual servoing Algorithmen werden hauptsächlich in der Robotik zur Bewegungssteuerung verwendet. Aufgrund der Vielzahl an Techniken werden in dieser Einführung lediglich die zwei meist genutzten Techniken, die bildbasierte visuelle Regelung (2-D visual servoing) und die positionsbasierte visuelle Regelung (3-D visual servoing) beschrieben. Während die Regelung beim 2-D visual servoing im Bildbereich stattfindet, wird bei der positionsbasierten Regelung die Bildinformation genutzt, um die Kamerapose zu bestimmen. Die Regelung findet dann im dreidimensionalen Weltkoordinatensystem statt. Dies führt zu einer optimalen Bewegungstrajektorie im dreidimensionalen Raum. Das bildbasierte Verfahren verfolgt dahingegen eine optimale Trajektorie im Bildbereich. Die Gefahr, Bildmerkmale durch ungeschickte Kamerabewegungen aus dem Bild zu verlieren, wird dadurch verringert und macht die bildbasierte Regelung robuster gegen Bildverarbeitungsfehler [6], [7].

Visuelle Regelsysteme wurden bereits erfolgreich als Flugregelung für MAVs getestet. So wird in [8] eine bildbasierte Regelung für einen autonomen Landeanflug auf einem Flugzeugträger vorgestellt. Eine Regelung unter Verwendung optischer Flussmessungen für einen Schwebeflug und eine anschließende Landung wurde in [9] erfolgreich getestet. In [10] erfolgt die Stabilisierung des Schwebefluges eines MAVs mit Hilfe einer Kombination aus einer Inertial Measurement Unit (IMU) und einem visuellen System.

Die Verwendung detektierter Gebäudemerkmal zur Unterstützung der Navigation und Flugregelung wurde in [11] und [12] vorgestellt. Gebäudefenster werden von einem Nutzer manuell ausgewählt und im Verlauf des

Fluges verfolgt. Dabei wurde das Fluggerät autonom bezüglich des Fensters ausgerichtet. Der Anflug hingegen wurde mit Hilfe eines GPS-Positionsreglers realisiert. Dies hat den Nachteil einer Abhängigkeit von gültigen GPS-Messungen, die insbesondere in Gebäudenähe häufig nicht vorhanden sind.

In dieser Arbeit wird ein Verfahren zu einem autonomen Gebäudeanflug vorgestellt, das während des Anfluges nicht auf gültige GPS-Messungen angewiesen ist. Zu Beginn des Anfluges werden sämtliche im Bild befindlichen Fenster detektiert. Die Auswahl des anzusteuern Fensters erfolgt durch den Nutzer. Im weiteren Verlauf wird das ausgewählte Zielfenster durch die Bildverarbeitung verfolgt. Der visual servoing Algorithmus wird zur Berechnung der Sollgeschwindigkeiten und Soll Drehraten verwendet, die benötigt werden, um das Fluggerät auf seine vordefinierte Sollposition zu manövrieren. Gleichzeitig erfolgt eine Schätzung der aktuellen Pose des Fluggerätes bezüglich des Zielfensters. Diese relativen Positionsinformationen kommen dann zur Stützung des Navigationsfilters zum Einsatz.

Nach einer kurzen Beschreibung der für diese Arbeit verwendeten Hardware in Kapitel 2 wird der Ablauf des autonomen Zielfluges in Kapitel 3 ausführlicher erläutert. Die Präsentation der mit diesem Verfahren erzielten Ergebnisse erfolgt in Kapitel 4. Der Abschluss der Arbeit erfolgt durch eine Zusammenfassung in Kapitel 5.

## 2. HARDWARE

Die folgenden Abschnitte beschreiben das für diese Arbeit verwendete Fluggerät, die Sensorik und die Bildverarbeitungshardware.

### 2.1. Fluggerät

Die für diese Arbeit verwendeten Fluggeräte wurden am Institut für Theoretische Elektrotechnik und Systemoptimierung (ITE) entwickelt. Neben unterschiedlichen Quadkoptern ist am ITE auch ein ach rotoriger Octocopter in Verwendung, der für die in dieser Arbeit präsentierten Testflüge genutzt wurde. Die Spannweite dieses Fluggerätes beträgt 105 cm. Damit ist es in der Lage, eine Nutzlast von bis zu 1,5 kg bei einer Flugzeit von 15 bis 20 Minuten zu tragen.

### 2.2. Navigationssystem

Alle am ITE genutzten Fluggeräte werden mit Hilfe eines selbstentwickelten Navigationsboards navigiert. Dieses kombiniert eine IMU mit diversen anderen Sensoren, wie GPS, einem Barometer-Altimeter und einem Magnetfeldsensor. Die Fusion der Sensorinformationen erfolgt direkt auf dem Navigationsboard unter Verwendung eines Error State Space Kalman Filters, welches in [13] näher beschrieben wird.

#### 2.2.1. Flugführung und Flugregelung

Neben den Navigationsinformationen werden auf dem Navigationsboard die Flugregelung und die Flugführung

ausgeführt. Zur Unterstützung des Nutzers ist die Flugführung unter anderem in der Lage, über einen Position Hold Modus eine definierte Position zu halten oder im Wegpunktnavigationsmodus eine Liste vordefinierter Wegpunkte abzufliegen. Die Flugregelung hat die Aufgabe die Höhe und die drei Lagewinkel (Roll, Nick und Gier) des Fluggerätes zu regeln.

### 2.2.2. Bildverarbeitungshardware

Aufgrund des erhöhten Rechenbedarfs wird die Bildverarbeitung nicht auf dem Navigationsboard ausgeführt, sondern auf der separaten Hardwareplattform CoolXpressRunner GS45. Das geringe Gewicht und der niedrige Leistungsbedarf von maximal 33 Watt machen diesen Embedded PC zu einem geeigneten Bildverarbeitungsrechner für kleine Fluggeräte. Er verfügt über einen Intel Core 2 Duo Prozessor mit einer Taktrate von 2,26 GHz, einem GMA 4500 Grafikprozessor und 1 GB DDR3-1066 RAM, sowie einer Solid State Festplatte [14]. Als Betriebssystem wurde eine Standard Linux Distribution installiert.

Die Kommunikation zwischen dem Navigationsboard und dem Embedded PC findet über eine serielle Schnittstelle statt. Während die meisten Sensoren über das Navigationsboard mit dem XpressRunner kommunizieren, ist die Kamera über USB direkt an den Embedded PC angeschlossen.

Als Kamera wird die CMOS Kamera UI-1240SE-C von IDS Imaging Development Systems eingesetzt. Diese zeichnet sich unter anderem durch einen global shutter, eine maximale Auflösung von 1028 x 1024 Pixeln und eine Bilderfassung von maximal 87 fps aus [15]. Das verwendete Objektiv DF6HA-1B von Fujinon mit einer fokalen Länge von 6 mm weist eine Blendenöffnung zwischen f1,2 und f16 auf [16]. Zur Reduzierung von Spiegelungen auf den anzufliegenden Fenstern und zur Kontrasterhöhung wird zudem ein zirkular polarisierter Filter verwendet.

## 3. AUTONOMER ZIELANFLUG

Im folgenden Kapitel werden das Konzept und die Umsetzung des bildbasierten Zielflugsystems dargestellt. Wie in Bild 1 zu sehen ist, laufen die Prozesse auf dem Navigationsboard und dem XpressRunner parallel zueinander ab. Die Flugregelung und Flugführung werden dabei sowohl auf dem Navigationsboard als auch auf dem XpressRunner ausgeführt. Im Falle einer Fehlfunktion des XpressRunners lässt sich das Fluggerät mit Hilfe des Navigationsboards weiter betreiben, wenngleich der bildbasierte Zielflug abgebrochen werden muss.

Das auf dem XpressRunner ausgeführte Zielflugsystem besteht im Wesentlichen aus drei Elementen. Als erstes erfolgt die Bildverarbeitung, die im Abschnitt 3.1 näher ausgeführt wird. Die Stützung der Navigation mit Hilfe von Messungen der Positionsänderung wird in Abschnitt 3.2 beschrieben. Eine Vorstellung des Flugführungsmoduls, bestehend aus einer zweidimensionalen bildbasierten Regelung und einer Geschwindigkeitsregelung, erfolgt in Abschnitt 3.3.

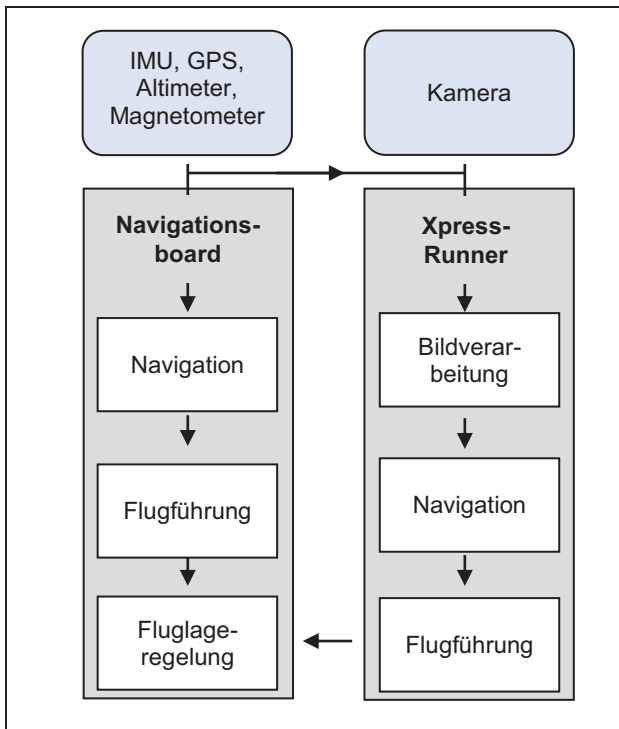


BILD 1: Darstellung der gesamten Flugregelschleife.

### 3.1. Zieldetektion und Verfolgung

Aus einem aufgenommenen Kamerabild wird zunächst ein Satz Binärbilder mit unterschiedlichen Schwellwerten generiert. Aus jedem dieser Binärbilder erfolgt dann im nächsten Schritt eine Konturextraktion mit Hilfe einem Verfahren nach Suzuki [17]. Von all diesen detektierten Konturen beschreibt jedoch lediglich eine Kontur das gesuchte Zielfenster. Da in einem Bild meist mehrere Fenster dargestellt werden, geschieht die Auswahl des Zielfensters manuell durch den Nutzer. Dieser definiert hierzu einen Point of Interest irgendwo innerhalb des Zielfensters. Um die gesuchte Kontur schließlich zu ermitteln, werden alle detektierten Konturen auf verschiedene Eigenschaften geprüft:

- 1) Zunächst muss die Kontur den vom Nutzer definierten Point of Interest umschließen.
- 2) Um sicherzugehen, dass die Kontur eine viereckige Form beschreibt, muss sie sich mit Hilfe von lediglich vier Punkten beschreiben lassen.
- 3) Die Falschalarmrate wird verringert, indem überprüft wird, ob die Kontur eine Fläche mit einer vordefinierten Mindestgröße umschließt.
- 4) Schlussendlich wird geprüft, ob die Ecken der Kontur näherungsweise rechteckig sind.

Sollte mehr als eine Kontur diese Eigenschaften aufweisen, wird die Kontur, die den Point of Interest am engsten umschließt, gewählt.

Die Beschreibung des Fensters geschieht durch die vier Fensterecken, welche mit Hilfe eines optischen Fluss-Verfahrens nach Lucas Kanade [18] im weiteren Verlauf

verfolgt werden. Dabei wird nach jedem Schritt die Plausibilität der Ergebnisse geprüft, um ein Wegdriften der Merkmalspunkte zu vermeiden. Dazu erfolgt eine Überprüfung des Verhältnisses von Fensterhöhe zu Fensterbreite und der Winkel der Fensterecken. Wird ein Drift der Merkmale festgestellt muss das Fenster erneut detektiert werden. Als Point of Interest kommt dabei der Mittelpunkt des zuletzt als gültig klassifizierten Fensters zur Verwendung.

### 3.2. Navigation

Nach einer erfolgreichen Detektion des Zielfensters wird die Pose des Fluggerätes geschätzt. Aus den dadurch bestimmten Relativpositionen zweier Zeitschritte lässt sich, wie in Bild 2 dargestellt, eine Positionsänderung  $\Delta pos$  folgendermaßen bestimmen:

$$(1) \quad \Delta pos_k^{b,k-1} = \vec{d}_{k-1}^{b,k-1} - C_n^{b,k-1} C_{b,k}^n C_c^{b,k} \vec{d}_k^c$$

Die Matrix  $C_c^{b,k}$  transformiert dabei von Kamerakoordinaten in das körperfeste Koordinatensystem. Der Zusammenhang zwischen dem Navigationskoordinatensystem und dem körperfesten Koordinatensystem, gültig zum vorherigen Zeitpunkt  $k-1$ , wird von der Transformationsmatrix  $C_n^{b,k-1}$  hergestellt. Die Entfernungen zwischen Fenster und Fluggerät werden mit  $\vec{d}_k$  bezeichnet.

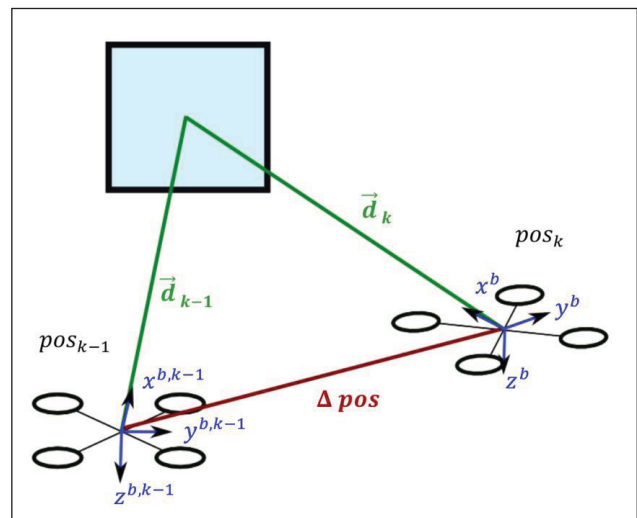


BILD 2: Berechnung der Positionsänderung aus Relativpositionen.

Der jeweilige Abstand zwischen Fluggerät und Fenster berechnet sich unter Zuhilfenahme der Posenschätzung nach Lu et al. [19].

Zur Verarbeitung der Positionsänderungen wird in dieser Arbeit ein sogenanntes Stochastic Cloning Error State Space Kalman Filter verwendet. Die Theorie des Stochastic Clonings wird detailliert in [20] und [21] beschrieben. Das Prinzip des Filterkonzeptes besteht darin, den Zustandsvektor des Kalman Filters um eine Kopie des vorherigen Zustandes zu erweitern, um Kreuzkorrelationen zwischen den Zuständen zu berücksichtigen.

Der erweiterte Zustand ergibt sich dann zu:

$$(2) \quad \delta x_{SC} = \begin{bmatrix} \delta \vec{x} \\ \delta \vec{x}_c \end{bmatrix}$$

In dieser Arbeit lautet der nicht geklonte Zustand  $\delta \vec{x}$ :

$$(3) \quad \delta \vec{x} = (\delta \vec{p}, \delta \vec{v}, \delta \vec{\sigma}, \delta \vec{b}_a, \delta \vec{b}_\omega, \delta \vec{b}_{baro})^T$$

Damit beinhaltet der Zustand Positionsfehler  $\delta \vec{p}$ , Geschwindigkeitsfehler  $\delta \vec{v}$ , Lagefehler  $\delta \vec{\sigma}$ , sowie Schätzfehler über die Sensorbiase des Beschleunigungsmessers  $\delta \vec{b}_a$ , des Drehratensensors  $\delta \vec{b}_\omega$  und des Barometer-Altimeters  $\delta \vec{b}_{baro}$  [13].

Neben dem Positionsfehler werden die Fehler des Roll-Winkels  $\delta \phi_c$  und des Gier-Winkels  $\delta \psi_c$  geklont:

$$(4) \quad \delta \vec{x}_c = (\delta \vec{p}_c, \delta \phi_c, \delta \psi_c)^T$$

Während  $\delta \vec{x}$  wie bei einem herkömmlichen Kalman Filter propagiert wird, darf der geklonte Teil  $\delta \vec{x}_c$  keine Änderung erfahren. Die Transitionsmatrix  $\Phi_{SC}$  des Filters lautet daher:

$$(5) \quad \Phi_{SC} = \begin{pmatrix} \Phi & 0_{16 \times 5} \\ 0_{5 \times 16} & I_{5 \times 5} \end{pmatrix}$$

Neben der Transitionsmatrix muss auch die Kovarianzmatrix  $P_{SC}$  entsprechend erweitert werden:

$$(6) \quad P_{SC} = \begin{pmatrix} P & \tilde{P}_c \\ \tilde{P}_c & P_c \end{pmatrix}$$

P beschreibt hierbei die Kovarianzmatrix der ungeklonten Zustände, während  $P_c$  die Kovarianzmatrix des geklonten Teils darstellt. In  $\tilde{P}_c$  sind die Kreuzkorrelationen zwischen den geklonten und ungeklonten Zustände zum Zeitpunkt des Klonens eingetragen. Während der Prädiktion bleiben die Einträge von  $P_c$  konstant, die Kreuzkorrelationen zwischen den geklonten und ungeklonten Zuständen werden jedoch prädiziert. Somit werden die Kreuzkorrelationen zwischen den Zuständen berücksichtigt. Die Einträge der Transitionsmatrix  $\Phi$  und der Kovarianzmatrix P für den ungeklonten Zustand finden sich in [13].

Die Herleitung des Mess- und Prädiktionsschrittes des Filters ist in [20] ausführlich dargestellt.

### 3.3. Flugführungsmodul

Das für diese Arbeit entwickelte Flugführungsmodul besteht, wie in Bild 3 dargestellt, aus zwei Kernelementen, der bildbasierten Regelung und einer Geschwindigkeitsregelung.

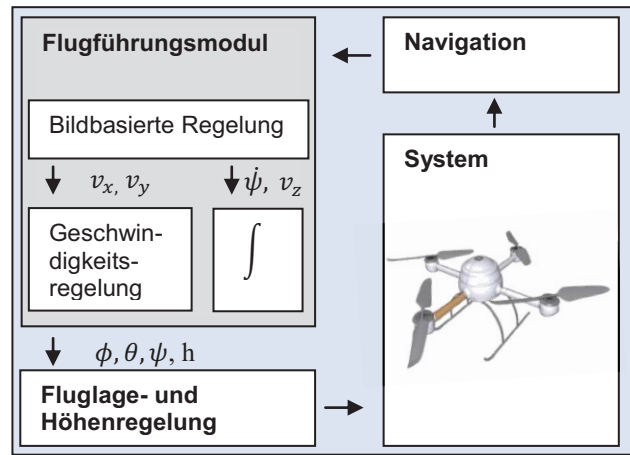


BILD 3: Komponenten des Flugführungsmoduls.

#### 3.3.1. Bildbasierte Regelung

Nach einer erfolgreichen Merkmalsdetektion wird neben der Merkmalsverfolgung auch die bildbasierte Regelung gestartet.

Während der Initialisierung werden die Positionen  $s^*$  der Merkmale im Bildbereich bestimmt, wie sie von der gewünschten Zielposition aus zu sehen sind. Die Bestimmung der Bildkoordinaten dieser Zielmerkmale erfolgt durch eine homogene Transformation und einer anschließenden Projektion von Welt- in Bildkoordinaten. Dafür muss jedoch die gewünschte Pose des Fluggerätes bezüglich des Fensters vom Nutzer vorgegeben werden. Desweiteren muss die Fenstergeometrie bekannt sein. In Bild 4 ist der Zusammenhang zwischen den aktuellen Merkmalspositionen  $s$  und den gewünschten Positionen im Bildbereich dargestellt.

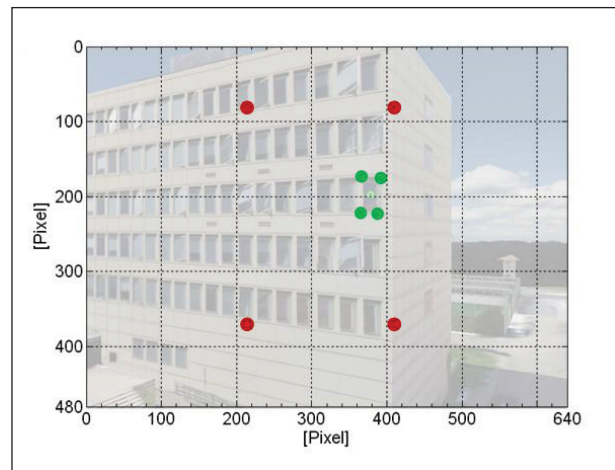


BILD 4: Darstellung der aktuellen Bildmerkmale (grün) und der gewünschten Merkmalspositionen im Bildbereich (rot).

Die Herleitung des hier verwendeten Regelgesetzes findet sich in [22], eine ausführliche Beschreibung in [6].

Das Regelgesetz gibt den Zusammenhang zwischen dem Fehler  $e = (s - s^*)$  und dem Vektor  $v_c$  an:

$$(7) \quad v_c = -\lambda L_e^+ e$$

Dieser Vektor enthält dabei neben der Geschwindigkeit auch die Drehraten, welche die Kamera (und damit das Fluggerät) auszuführen hat, um den Fehler  $e$  zu minimieren.  $L_e^+$  ist die Moore-Penrose Pseudo Inverse von  $L_e$ :

$$(8) \quad L_e = \begin{bmatrix} \frac{-1}{z} & 0 & \frac{x}{z} & xy & -(1+x^2) & y \\ 0 & \frac{-1}{z} & \frac{y}{z} & (1+y^2) & -xy & -x \end{bmatrix}$$

Die Merkmalskoordinaten im Bildbereich werden mit  $x$  und  $y$  beschrieben. Die aktuelle Tiefe jedes Merkmales  $Z$  muss bei jedem Iterationsschritt bekannt sein. Diese Information wird auch schon zur Bestimmung der Positionsänderung herangezogen und steht damit bei der Berechnung der bildbasierten Regelung zur Verfügung.

### 3.3.2. Geschwindigkeitsregelung

Die Geschwindigkeitsregelung hat zur Aufgabe, die horizontale Sollgeschwindigkeit in Solllagewinkel zu transformieren, die als Stellgrößen für die Fluglageregelung dienen. Die Sollhöhe und die Yaw-Lage bestimmen sich durch einfache Integration.

Der Zusammenhang zwischen den Lagewinkeln und den Sollgeschwindigkeiten  $\tilde{v}_{eb}^n$  lautet:

$$(9) \quad \phi_{set} = K_p(\tilde{v}_{eb,x}^n - v_{eb,x}^n) \cdot \sin(\psi) - K_p(\tilde{v}_{eb,y}^n - v_{eb,y}^n) \cdot \cos(\psi)$$

$$(10) \quad \theta_{set} = K_p(\tilde{v}_{eb,x}^n - v_{eb,x}^n) \cdot \cos(\psi) + K_p(\tilde{v}_{eb,y}^n - v_{eb,y}^n) \cdot \sin(\psi)$$

Die Schätzung für die Geschwindigkeit  $v_{eb}^n$  stammt aus dem Navigationsfilter.

## 4. ERGEBNISSE

Um die Ergebnisse mit Referenzwerten vergleichen zu können, wurden neben realen Flugdaten auch simulierte Flugdaten verwendet. Die Echtzeitfähigkeit des Systems wird mit Hilfe von Laufzeittests demonstriert.

### 4.1. Simulationsergebnisse

Für die folgenden Experimente wurde eine am Institut entwickelte Simulationsumgebung verwendet. Diese enthält ein physikalisches Modell des Fluggerätes mit Modellen aller Sensoren zur Erzeugung verrauschter Sensordaten, sowie ein 3D-Modell des Campus. Dieses 3D-Modell wird zusammen mit einer OpenGL-Umgebung eingesetzt um künstliche Kamerabilder passend zur jeweiligen Flugsituation zu erstellen. Die Simulationsumgebung ist damit in der Lage, neben Referenzwerten auch realistische Sensorsignale zu

generieren, die zur Erzeugung der folgenden Ergebnisse verwendet wurden.

#### 4.1.1. Navigation

Mit Hilfe der beschriebenen Simulationsumgebung wurde ein Anflug auf ein Fenster simuliert. Die Simulation wurde parallel mit zwei verschiedenen Navigationsfiltern, einem Stochastic Cloning Error State Space Kalman Filter mit Bildstützung und einem Error State Space Kalman Filter ohne Bildstützung durchgeführt. Neben den IMU Messungen verwenden beide Filter Messungen des Barometer-Altimeters und des Magnetometers.

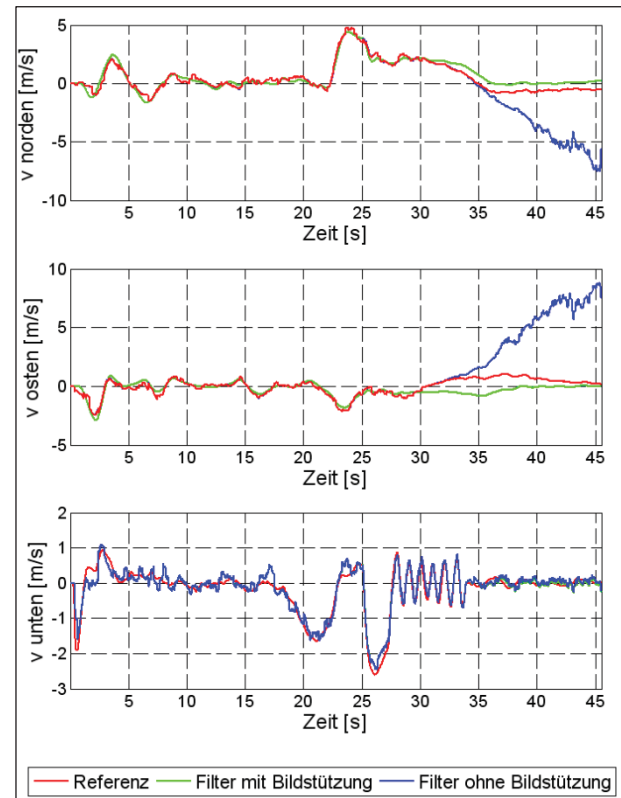


BILD 5: Vergleich eines bilgestützten Errors State Space Kalman Filters mit einem Error State Space Kalman Filter ohne Bildstützung.

Während der ersten 20 Sekunden der Simulation stand das Fluggerät still auf dem Boden und die Navigationsfilter schätzten die Sensorfehler. Anschließend startete das Fluggerät in Richtung des Fensters. Auf Grund der Annäherung an das Gebäude wurde ab der 28. Sekunde ein totaler GPS-Signalausfall simuliert. Vor diesem Signalausfall sind die Ergebnisse beider Filter, wie in Bild 5 dargestellt, nahezu identisch. Ab dem Zeitpunkt des Ausfalls steigen die horizontalen Geschwindigkeitsschätzfehler des Filters ohne Bildstützung sehr stark an. Die Schätzfehler des bildgestützten Navigationsfilters sind nach dem GPS-Signalausfall leicht erhöht. Da bei diesem Filter jedoch die inertielle Navigation weiterhin gestützt wird, driftet die horizontale Geschwindigkeitsschätzung nicht ab. Die vertikale Geschwindigkeitsschätzung ist für beide Filter

nahezu identisch, da diese zusätzlich durch die Messungen des Barometer-Altimeters gestützt wird.

#### 4.1.2. Flugführung

Auch zur Verifikation der Flugführung wurde ein simulierter Gebäudeanflug genutzt.



BILD 6: Simuliertes Kamerabild zu Beginn des Anfluges (links) und nach Erreichen der Zielposition (rechts).

In Bild 6 sind zwei simulierte Kamerabilder dargestellt. Einmal vom Zeitpunkt der Fensterdetektion und einmal nach Erreichen der Zielposition. Die Entfernung des MAVs zu diesem Zielfenster ist in Bild 7 in blau dargestellt. Die vordefinierte Zielposition ist dabei rot markiert. In beiden Abbildungen ist erkennbar, dass das Fluggerät seine Sollposition mit einer nur geringen Abweichung erreicht. Maximal beträgt diese in z-Richtung des Fluggerätes 0,25 m.

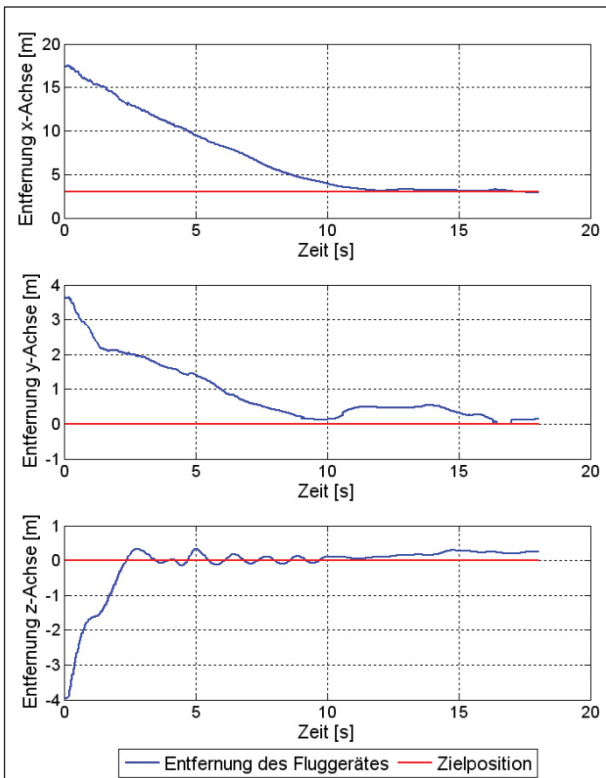


BILD 7: Darstellung der Entfernung des Fluggerätes zum Zielfenster (blau) im Vergleich zur vorgegebenen Entfernung (rot).

#### 4.2. Ergebnisse unter Verwendung von realen Daten

Die in diesem Abschnitt präsentierten Ergebnisse entstanden unter Verwendung der bereits beschriebenen Hardware. Da die in Bild 1 dargestellte Rückführung vom Navigationsboard auf den XpressRunner noch nicht implementiert ist, wurde das Fluggerät manuell gesteuert. Die autonome Flugführung kam hierbei nicht zum Einsatz.

##### 4.2.1. Ergebnisse der Bildverarbeitung

Wie in Bild 8 zu sehen ist, ist das in dieser Arbeit beschriebene Detektionsverfahren in der Lage, nahezu alle dargestellten Fenster fehlerfrei zu detektieren.

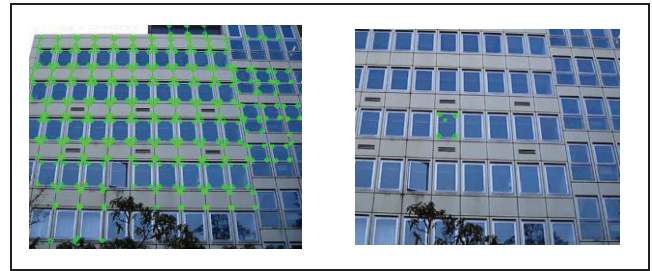


BILD 8: Detektion aller sichtbaren Fenster (links) und des ausgewählten Zielfensters (rechts).

Nachdem der Point of Interest definiert wurde, erfolgt auf der rechten Bildseite die Selektion des gewünschten Zielfensters.

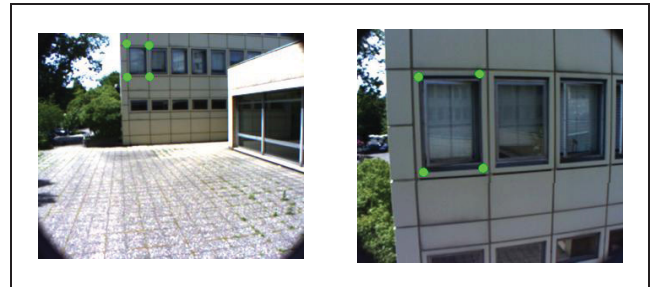


BILD 9: Detektiertes Fenster zu Beginn des Anfluges (links) und bei Erreichen der Endposition (rechts).

Zur Demonstration der Objektverfolgung sind in Bild 9 zwei Kamerabilder zu unterschiedlichen Zeiten dargestellt. Das Kamerabild auf der linken Seite stammt vom Zeitpunkt der Detektion, während die Aufnahme auf der rechten Seite 20 Sekunden später aufgenommen wurde. Auch unter Verwendung realer Kamerabilder werden die Fensterreihen sicher verfolgt und driften mitunter auf Grund der bereits beschriebenen Plausibilitätsprüfung nicht ab.

### 4.2.2. Ergebnisse des Navigationsfilters

Im Gegensatz zu den Ergebnissen aus der Simulationsumgebung stehen im folgenden Abschnitt keine Referenzlösungen zur Verfügung. Eine Bewertung der Ergebnisse kann daher nur qualitativ erfolgen.

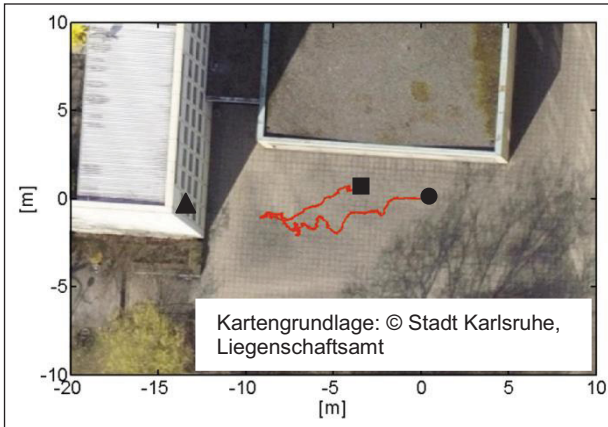


BILD 10: Georeferenziertes Luftbild mit geschätzter Flugtrajektorie, Startpunkt (●) und Endpunkt(■) des Fluges, sowie der Position des Zielfensters (▲) [23].

Die Ergebnisse des Navigationsfilters entstanden während eines kurzen Fluges, bei dem sich das Fluggerät zunächst auf das Zielfenster zubewegt und anschließend davon entfernt. Dabei blieb die Kamera jedoch immer auf das Fenster gerichtet.

Um eine Vergleichbarkeit der Ergebnisse zu erzielen, wurde parallel zum bildgestütztem Navigationsfilter ein herkömmliches Error State Space Kalman Filter ausgeführt. Beide Filter starteten zum Zeitpunkt der Fensterdetektion. Aufgrund der Abschattung durch die im Bild 10 abgebildeten Gebäude wurde bei diesem Experiment keine GPS-Stützung verwendet. Daher erfolgte eine Initialisierung beider Filter mit der gleichen Startposition. Beide Navigationsfilter verwendeten Messungen der IMU, des Barometer-Altimeters und des Magnetometers.

Die vom Bildgestützten Navigationsfilter geschätzte Flugtrajektorie ist in Bild 10 dargestellt. Sowohl der Verlauf als auch die Start- und Endpunkte der Trajektorie beschreiben den Flug plausibel.

Auch die Geschwindigkeitsschätzung des bildgestützten Navigationsfilters aus Bild 11 stimmt mit dem Flugverlauf überein. Während der ersten 15 Sekunden bewegt sich das Fluggerät mit positiver Geschwindigkeit in Body-x-Richtung auf das Fenster zu. Anschließend befindet es sich für 10 Sekunden nahezu stationär vor dem Fenster bevor es sich mit negativer Geschwindigkeit in Body-x-Richtung vom Fenster entfernt.

Zum Vergleich sind in blau die Schätzungen des herkömmlichen Navigationsfilters ohne Bildstützung abgebildet. Aufgrund der fehlenden GPS-Stützung driftet die Geschwindigkeitsstzung sowohl in x- als auch in y-Richtung stark ab und ist damit, im Gegensatz zur bildgestützten Schätzung, zur Flugregelung nicht mehr verwendbar.

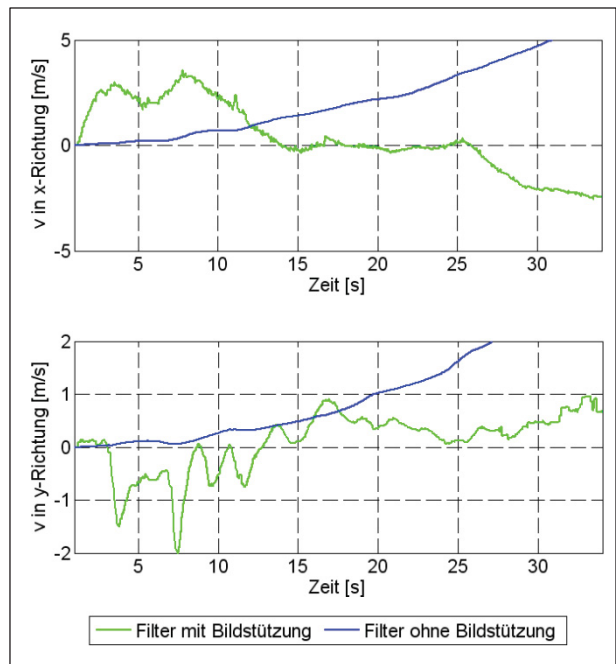


BILD 11: Vergleich der Geschwindigkeitsschätzungen zweier Navigationsfilter. Error State Space Kalman Filter gestützt durch Bildverarbeitung (grün) und ohne Stützung durch Bildverarbeitung (blau).

### 4.3. Laufzeittests

Um eine präzise Fluglageregelung zu ermöglichen, darf die Berechnung der gesamten Regelschleife nicht mehr als vier Millisekunden überschreiten. Dies beschränkt die für die Bildverarbeitung zur Verfügung stehende Berechnungszeit auf dem XpressRunner erheblich.

Wie in Tabelle 1 zu sehen ist, überschreitet der Bildverarbeitungspart diese Zeitbeschränkung. Aus diesem Grund wird dieser in einem gesonderten Thread parallel zu den sonstigen Berechnungen ausgeführt. Im Gegensatz zur Bildverarbeitung ist der Zeitbedarf zur Berechnung der Flugführung mit 0,19 ms sehr gering und wird daher nicht weiter untersucht.

Tabelle 2 gibt die Anzahl der parallel zum Bildverarbeitungsthread ausgeführten Regelschleifen an. Mit einer Kameraupdaterate von 30 Hz kann damit während der Merkmalsverfolgung jedes Kamerabild verwendet werden. Damit ist die Echtzeitfähigkeit der Bildverarbeitung demonstriert.

	Maximal	Durchschnitt
Bildverarbeitung	78,32 ms	19,02 ms
• Detektion	78,32 ms	47,71 ms
• Verfolgung	15,46 ms	7,66 ms
Flugführung (inkl. visual servoing)	0,19 ms	0,02 ms

TAB 1: Laufzeiten der einzelnen Programmkomponenten

	Maximal	Durchschnitt
Detektion	21 Schleifen	5 Schleifen
Verfolgung	5 Schleifen	2 Schleifen

TAB 2. Anzahl der parallel zur Bildverarbeitung ausgeführten Regelschleifen.

## 5. ZUSAMMENFASSUNG

In dieser Arbeit wird ein Verfahren für einen autonomen Zielflug an ein Gebäudefenster vorgestellt. Eine Kamera wird zur Detektion und Verfolgung des Zielfensters verwendet. Mit Hilfe eines bildbasierten Regelverfahrens wird das Fluggerät auf eine vordefinierte Zielposition vor dem Fenster manövriert, ohne die Fenstermerkmale aus dem Sichtbereich zu verlieren. Die nachfolgenden Geschwindigkeits- und Lageregelungen dienen hierbei zur Umsetzung der Stellgrößen aus dem bildbasierten Regler.

Um eine präzise Flugregelung zu ermöglichen ist das Wissen über die Position und Lage des Fluggerätes zu jedem Zeitpunkt notwendig. Um über diese Informationen auch zu Zeiten eines GPS-Ausfalles zu verfügen, werden Positionsänderungen über Posenschätzungen bestimmt und mit Hilfe eines Stochastic Cloning Error State Space Kalman Filters verarbeitet.

Es hat sich sowohl in der Simulation als auch bei realen Testflügen gezeigt, dass die verwendeten Bildverarbeitungsmethoden das Zielfenster sicher detektieren und die Fenstermerkmale während eines Anfluges sicher verfolgt werden. Des Weiteren konnte bei diesen Tests zu Zeiten eines GPS-Ausfalles eine signifikante Verbesserung der Navigationslösung des vorgestellten bildgestützten Navigationsfilters gegenüber den Ergebnissen eines Filters ohne Bildstützung demonstriert werden. Die Verifikation des vorgestellten Flugführungsmoduls erfolgte mit Hilfe einer Simulationsumgebung.

Die Echtzeitfähigkeit des Gesamtsystems wurde mit Hilfe von Laufzeittest belegt.

## 6. LITERATURVERZEICHNIS

- [1] G. Conte und P. Doherty, „An Integrated UAV Navigation System Based on Aerial Image Matching,“ *IEEE Aerospace Conference*, pp. 1-10, March 2008.
- [2] L. R. G. Carrillo, A. E. D. López, R. Lozano und C. Pégard, „Combining Stereo Vision and Inertial Navigation System for a Quad-Rotor UAV,“ *Journal of Intelligent & Robotic Systems*, Bd. 65, Nr. 1-4, p. 373, 2011.
- [3] J. Huang und B. Cowan, „Simple 3D Reconstruction of Single Indoor Image with Perspective Cues,“ *Canadian Conference on Computer and Robot Vision*, pp. 140-147, 2009.
- [4] C. Bills, J. Chen und A. Saxena, „Autonomous MAV Flight in Indoor Environment using Single Image Perspective Cues,“ *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5776 - 5783, 2011.
- [5] M. Blösch, S. Weiss, D. Scaramuzza und R. Siegwart, „Vision Based MAV Navigation in Unknown and Unstructured Environments,“ *IEEE International Conference on Robotics and Automation*, pp. 21 - 28, 2010.
- [6] F. Chaumette und S. Hutchinson, „Visual Servo Control Part I: Basic Approaches,“ *Robotics & Automation Magazine, IEEE*, Bd. 13, Nr. 4, p. 82, 2006.
- [7] S. Hutchinson, G. Hager und P. Corke, „A Tutorial on Visual Servo Control,“ *IEEE Transactions on Robotics and Automation*, Bd. 12, Nr. 5, pp. 651 - 670, 1996.
- [8] L. Coutard, F. Chaumette und J.-M. Pfimlin, „Automatic landing on aircraft carrier by visual servoing,“ *International Conference on Intelligent Robots and Systems*, pp. 2843 - 2848, 2011.
- [9] B. Herisse, F.-X. Russotto, T. Hamel und R. Mahony, „Hovering flight and vertical landing control of a VTOL Unmanned Aerial Vehicle using optical flow,“ *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 801-806, 2008.
- [10] H. Romero, S. Salazar und R. Lozano, „Visual servoing applied to real-time stabilization of a multi-rotor UAV,“ *Robotica*, pp. 1-10, 2012.
- [11] L. Mejias, P. Campoy, S. Saripalli und G. Sukhatme, „A Visual Servoing Approach for Tracking Features in Urban Areas Using an Autonomous Helicopter,“ *IEEE International Conference on Robotics and Automation*, pp. 2503 - 2508, 2006.
- [12] L. Mejias, S. Saripalli, P. Campoy und G. S. Sukhatme, „Visual Servoing of an Autonomous Helicopter in Urban Areas using Feature Tracking,“ *Visual Servoing of an Autonomous Helicopter in Urban Areas using Feature Tracking*, Bde. 3-4, Nr. 23, pp. 185-201, 2006.
- [13] O. Meister, Entwurf und Realisierung einer Aufklärungsplattform auf Basis eines unbemannten Minihelikopters mit autonomen Flugfähigkeiten, Karlsruhe Institute of Technology, Institute of Systems Optimization: Logos, 2010.
- [14] „lippertembedded,“ [Online]. Available: <http://www.lippertembedded.de/en/lipperts-cool-xpressrunner-gs45.html>. [Zugriff am Mai 2012].
- [15] „ids-imaging,“ [Online]. Available: [http://www.ids-imaging.de/pdfmodule/products\\_pdf.php?cam\\_id=127](http://www.ids-imaging.de/pdfmodule/products_pdf.php?cam_id=127). [Zugriff am August 2012].
- [16] „Fujinon,“ [Online]. Available: <http://www.fujinonbroadcast.com/ProductDocs/DF6H-A-1B.pdf>. [Zugriff am August 2012].
- [17] S. Suzuki und K. Abe, „Topological Structural Analysis of Digitized Binary Images by Border Following,“ *Computer Vision, Graphics, and Image Processing*, Bd. 30, Nr. 1, p. 32, 1985.
- [18] J.-Y. Bouguet, „Pyramidal Implementation of the Lucas Kanade Feature Tracker,“ *Technical report, Tech. Rep. included in the OpenCV library*, 2000.
- [19] C.-P. Lu, G. D. Hager und E. Mjølness, „Fast and Globally Convergent Pose Estimation from Video Images,“ *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Bd. 22, Nr. 6, pp. 610 - 622, 2000.



- [20] S. Roulletis und J. Burdick, „Stochastic Cloning: A generalized framework for processing relative state measurements,“ *IEEE International Conference on Robotics and Automation*, Bd. 2, Nr. 2, pp. 1788 - 1795 , 2002.
- [21] A. Mourikis, S. Roulletis und J. Burdick, „SC-KF Mobile Robot Localization: A Stochastic Cloning Kalman Filter for Processing Relative-State Measurements,“ *IEEE Transactions on Robotics*, Bd. 23, Nr. 4, pp. 717 - 730 , 2007.
- [22] B. Espiau, F. Chaumette und P. Rives, „A new Approach to Visual Servoing in Robotics,“ *IEEE Transactions on Robotics and Automation*, Bd. 8, Nr. 3, pp. 313 - 326 , 1992.
- [23] Stadt Karlsruhe, Liegenschaftsamt, „Luftbild,“ Karlsruhe, 2011.