

Robuste kameragestützte Präzisionslandung von automatisierten fliegenden Systemen

Endres Kathe, Amilcar do Carmo Lucas, Enrico Neumann
und Pedro Miguel Isaac Delso

IAV GmbH
Carnotstraße 1, 10587 Berlin

Zusammenfassung Eine besonders kritische Flugphase bei der Automatisierung von unbemannten automatisch fliegenden Systemen stellt das Landen dar. Je nach Größe des nutzbaren Flugkorridors und des Landeplatzes können hier Genauigkeiten im Zentimeterbereich an die Wiederholbarkeit der Flugbahn und Landeposition gefordert werden. Auch elektromagnetische Störungen können die Nutzung herkömmlicher Systeme wie GNSS besonders im Landebereich erschweren. Das von der IAV entwickelte optische Landesystem stellt eine ganzheitliche Entwicklung dar, die über die Landeplattform, die Einbindung und hardwareseitige Steuerung der Kamera, die Bildverarbeitung und Positionsberechnung bis hin zu der Integration in das Flugsteuerungssystem reicht. Durch den spezifischen Aufbau unseres Systems wird eine hohe Robustheit gegenüber Änderungen der Umgebungsbedingungen erreicht. Zusätzlich wird die Integrität des Systems durch die Schätzung der (Positions-)Genauigkeiten und der Rückmeldung des Zustandes des Gesamtsystems sichergestellt. Die Entwicklung und Tests des Systems erfolgten sowohl in der Simulation als auch unter verschiedenen realen Flugbedingungen.

Keywords Drohne, UAV, Automatisierung, Präzises Landen, Fiducial Markers, Robotik, Computer Vision, Robuste Schätzverfahren

DOI: 10.58895/ksp/1000124383-31 erschienen in:

Forum Bildverarbeitung 2020

DOI: 10.5445/KSP/1000124383 | <https://www.ksp.kit.edu/site/books/m/10.58895/ksp/1000124383/>

1 Einführung

Drohnenysteme finden vermehrt Einzug in den kommerziellen Sektor. Damit verbunden steigt auch der Wunsch nach einem höheren Automatisierungsgrad. Eine besonders kritische Flugphase bei der Automatisierung stellt dabei das Landen dar. Je nach Größe des nutzbaren Flugkorridors und des Landeplatzes können hier Genauigkeiten im Zentimeterbereich an die Wiederholbarkeit der Flugbahn und Landeposition gefordert werden. Zusätzlich kann es gehäuft im Landebereich zu Abschattungen [1] und/oder Multipath [2] von GNSS-Signalen, wie etwa durch hohe Wände, kommen. Aber auch andere Störgrößen sind oftmals auf den letzten Metern des Fluges anzutreffen, wie z. B. magnetische oder elektrische Einflüsse durch Stromleitungen. Dies behindert oft das Nutzen herkömmlicher Positions- und Orientierungssysteme wie z. B. GPS-Empfänger oder Magnetometer.

Kameras sind von diesen Störungen nicht betroffen. Zusätzlich stellen sie eine informationsreiche Sensorquelle dar, die hohe Redundanzen und Genauigkeiten ermöglicht. Deshalb haben sich die optischen Verfahren für Landeanflüge als besonders geeignet herausgestellt.

Im Folgenden soll das durch die IAV GmbH entwickelte System vorgestellt werden, was den Fokus auf Robustheit mittels Redundanz, Integrität und Flexibilität setzt, welche nachstehend erläutert werden.

2 Beschreibung des Landesystems

Das System besteht aus einer aktiven Komponente, die sich auf der Drohne befindet. Diese besteht aus einer Industriekamera mit global Shutter im Verbund mit einem Companion-Computer, auf welchem die Software läuft. Zusätzlich kann an der Drohne eine Beleuchtung angebracht werden, welche eine Landung bei Nacht ermöglicht. Ein Teil des Systems ist in Abbildung 2.1 dargestellt.

Die passive Komponente befindet sich am Landeplatz und besteht aus einer Folienkombination aus retroreflektivem weiß und mattem schwarz. Durch das matte schwarz werden Reflektionen durch



Abbildung 2.1: Aufbau des Systems auf der Drohne.

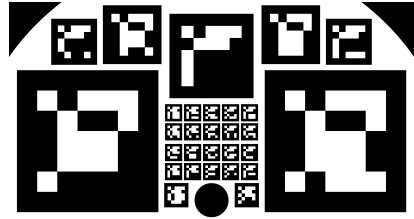


Abbildung 2.2: Aufbau der Landemarkierung.

ungünstige Einstrahlwinkel der Sonne oder anderer Lichtquellen minimiert. Das retroreflektive Weiß ermöglicht das effiziente Beleuchten durch die Drohne bei Dunkelheit. Die Folien bilden Aruco-Marker [3] nach dem Muster, wie es in Abbildung 2.2 ersichtlich ist. Diese werden in ihrer Größe und Kombination der Landefläche angepasst, um die Sichtbarkeit mehrerer Marker bei unterschiedlichen Höhen im Kamerabild zu ermöglichen.

2.1 Beschreibung der Bildverarbeitung

Das in diesem Artikel vorgestellte Verfahren kombiniert verschiedene Algorithmen der Bildverarbeitung zu einer Datenverarbeitungskette, um aus den aufgenommenen Bildern zunächst die Markierungen zu erkennen und anschließend eine Lösung der relativen Position und Orientierung zu erhalten. Zusätzlich wird die Güte der Positionslösung geschätzt.

Da das System im Außenbereich arbeitet, ist es starken Schwankungen des Umgebungslichts ausgesetzt, welche von einer dunklen Nacht bis zu einem hellen, wolkenfreien Sommertag mit gegebenenfalls ungleichmäßiger Beleuchtung reichen.

Einstellung des Kamera-Gains

Das Hardware-Gain der Kamera wird je nach Umgebungslicht mittels einer Histogrammanalyse dynamisch angepasst. Die Verschlusszeit wird auf einen empirisch ermittelten Wert fixiert und stellt einen Kompromiss zwischen Bildrauschen und Motion-Blur dar, der durch die Flugbewegung verursacht wird.

Adaptive Threshold

Das durch die Kamera erzeugte Bild wird zuerst mit einem Adaptive Threshold Algorithmus binarisiert, um es somit in das Schwarzweiße zu übersetzen. Hier wird bei jedem Pixel ein Histogramm der Nachbarschaftspixel erstellt, wodurch der individuelle Schwellwert für die Binarisierung berechnet wird [4]. Wird bei den nachfolgend erläuterten Prozessen kein Marker im Bild erkannt, wird der Parameter der Nachbarschaftsgröße stetig automatisch durch das System verändert um so unterschiedlichste Ausleuchtungen der Landemarkierung kompensieren zu können.

Konturensuche

Das binarisierte Bild wird nun nach konvexen Vierecken durchsucht. Zuerst werden Konturen mit einem border following algorithmus gefunden [5]. Anschließend werden diese mit dem Verfahren von Douglas-Peucker vereinfacht [6].

Danach kommt das Quadrilateral Sum Conjecture-Kriterium zum Einsatz um zu prüfen, ob es sich bei einer gefundenen Kontur um ein konvexes Viereck bzw. ein Quadrat unter perspektivischer Verformung handelt [7]. Hierbei muss die Summe der vier Winkel der Kontur 360° ergeben. Dieses Kriterium ermittelt auch Quadrate unter starker perspektivischer Verzerrung. Als weitere Kriterien darf die Summe des Cosinus der Winkel des Vierecks einen gewissen Wert nicht überschreiten und die Pixelfläche des gefundenen und auf die Bildfläche projizierten Viereck muss eine Mindestgröße besitzen, um so Rauschen herauszufiltern.

Identifikation der Aruco-Marker

Wurde ein Viereck im Bild gefunden, folgt die Prüfung, ob es sich um einen Aruco-Marker [3] handelt. Zur Verwendung kommt hierbei das standard Aruco-Dictionary [8]. Die perspektivische Verzerrung wird korrigiert. Anschließend wird mit Hilfe einer linearen Interpolation das Bild in eine 7×7 Matrix übersetzt, welche anschließend mit dem Otsu-Binarisierungsalgorithmus [9] wieder in das binäre übersetzt wird. Die gefundene Binärmatrix wird auf die Zugehörigkeit des Aruco-Codebereichs mit Hilfe der Signaturmatrix geprüft. Anschließend werden die auf das Bild projizierten gefundenen Eckpunkte des Markers Koordinaten im Raum zugeordnet, welche zuvor eingemessen wurden und die in einer Datenbank hinterlegt sind.

Gewinnung der Positionslösung

Anschließend wird versucht, mit Hilfe des PnP-Algorithmus von OpenCV, welcher auf der iterativen Levenberg-Marquardt Optimierung basiert, die Pose der Kamera relativ zum Marker zu finden.

Daraufhin wird der Reprojektionsfehler berechnet. Überschreitet dieser einen gewissen Schwellenwert, so wird davon ausgegangen, dass das iterative Lösungsverfahren fehlgeschlagen ist. Dies kann zum Beispiel der Fall bei einer Fehldekektion sein, ein Marker wurde fälschlich identifiziert oder aber es wurde der Versuch von spoofing unternommen, indem ein weiterer Marker aus dem gleichen Code-Bereich in das Bild gebracht wurde. Die Lösung des PNP-Problems wird dann noch einmal mit dem RANSAC-Verfahren unternommen. Dieses Verfahren weist eine höhere Robustheit gegen Außereißer auf, indem es diese aus der Lösung ausschließt.

Anschließend erfolgt eine Schätzung der Güte C der Positionslösung. Nach dem hier vorgestellten Modell verhält sich diese antiproportional zu der Fläche A eines konvexen Polygons, welche die Projektion der Marker auf die Bildfläche einhüllt und proportional zu dem Abstand d zwischen Kamera und Markermittelpunkt. Ein zusätzlicher, linearer Faktor k wird empirisch durch die Simulation ermittelt und spiegelt die intrinsischen Parameter der Kamera wieder.

$$C = k \cdot \frac{d}{A} \quad (2.1)$$

Durch den redundanten Aufbau der Landemarkierung kann trotz des Ausfalls eines oder mehrerer Marker eine Positionslösung generiert werden und eine Landung erfolgen. Darüber hinaus gibt es eine Rückmeldung darüber, welche Marker bei einem Landeanflug nicht erkannt wurden, wie es zum Beispiel bei einer Verdeckung der Fall ist. So kann der Zustand der Markierung von dem System selbst verfolgt und bei Bedarf von Mensch eingeschritten werden um zum Beispiel die Markierung zu reinigen oder zu erneuern.

Integration der Positionslösung in den Flugcontroller

Als letzter Schritt folgt die Integration der Positionslösung in den Flugcontroller, um die präzise Landung zu ermöglichen. Für die Software des Flugcontrollers erfolgt die Wahl des Flightstacks Arducopter.

Die Standardimplementierung für die Präzisionslandung erwartet hierbei als Eingang einen Winkel der Line of Sight zwischen der optischen Achse der Kamera und der Landemarkierung, sowie deren Distanz zueinander. Auf der Drohne werden ständig die Lagewinkel der Drohne mit Hilfe der Intertial Measurement Unit (IMU) gemessen. Da die Kamera fest mit der Drohne verbunden ist, ist somit auch die Orientierung der Kamera bekannt. Die gemessenen Winkel der Kamera können zu einem Einheitsvektor übersetzt werden, welcher von dem Körperfesten Koordinatensystem der Kamera zu der Landemarkierung zeigt.

$$||\vec{v}_{body_unit}|| = 1 \quad (2.2)$$

Mit Hilfe einer Transformationsmatrix kann der Vektor von dem körperfesten Koordinatensystem in das North-East-Down (NED)-Koordinatensystem überführt werden.

$$\vec{v}_{ned_unit} = T_{ned_body} \cdot \vec{v}_{body_unit} \quad (2.3)$$

Anschließend wird der Einheitsvektor mit der Distanz zu der Landeposition multipliziert, um so einen Vektor zu erhalten, der die Ablage der Drohne von der Landeposition im NED-Koordinatensystem beschreibt.

$$\vec{v}_{ned} = \vec{v}_{ned_unit} \cdot d_{target_distance} \quad (2.4)$$

Dieses Verfahren ist besonders in größeren Höhen geeignet, wo die Güte der Positionslösung nach Gleichung 2.1 durch die geringe Größe der Landemarkierung im Bild und den hohen Abstand beeinträchtigt ist.

In niedrigeren Höhen steigt die Güte der Positionslösung. Unterschreitet diese einen Schwellenwert, so kann die Ablageposition von Drohne zu Landemarkierung direkt und ohne Umwege aus der Markierung mit Hilfe der Lösung des PnP-Problems gelesen und in den Flugcontroller eingespeist werden. Hierfür wurde der ArduCopter-Code modifiziert und es entfallen Fehler, die durch Messfehler der IMU, Ungenauigkeiten bei der IMU-Kamera synchronisierung oder bei einem schräglagigen Einbau der Kamera entstehen würden.

3 Simulation

Für die Simulation wurde die Umgebung Gazebo [10] genutzt. Dabei wurden die intrinsischen Parameter der Kamera mit einem horizontalen FOV von 68° bei einer Auflösung von 1216×1024 Pixel und der Aufbau der optischen Markierung nach Abbildung 2.2 mit einer Breite von 1,4m nachmodelliert. Die Pose zwischen Marker und Kamera wurde nach dem Zufallsverfahren variiert. Anschließend berechnet der Algorithmus die Position zwischen Kamera und Landemarkierung sowie den Winkel der Line Of Sight (LOS) der Kamera. Bei jeder Iteration wird die Lösung aus der Bildverarbeitung zusammen mit der Groundtruth aus der Simulation gespeichert. Hierdurch ist es möglich, den Messfehler zu bestimmen, welcher in der Tabelle 1 und 2 sowohl für die Position als für die Winkelbestimmung dargestellt ist. Die Positionslösung ist hierbei bereits nach dem Gütekriterium gefiltert und daher nur in einer Höhe bis zu 2m vorhanden.

Tabelle 1: Positionsgenauigkeit in verschiedenen Höhenbändern

Höhenband [m]	Anzahl der Messpunkte	$\mu \pm 1\sigma [m]$
0 - 1	42	0.0022 ± 0.0065
1 - 2	108	0.0021 ± 0.0207

Tabelle 2: Winkelgenauigkeit in verschiedenen Höhenbändern

Höhenband [m]	Anzahl der Messpunkte	$\mu \pm 1\sigma [rad]$
0 - 1	45	0.0109 ± 0.0180
1 - 2	124	0.0024 ± 0.0058
2 - 3	141	0.0017 ± 0.0027
3 - 4	135	0.0012 ± 0.0023
4 - 5	150	0.0013 ± 0.0010

Das Gütekriterium wurde empirisch ermittelt und eliminiert Außerreißer in der Positionslösung. In einer Höhe von unter 2m werden 88,76% der Positionslösungen direkt in den Flugcontroller eingespeist.

Die Winkelmessung ist über alle gemessene Höhenbänder nach Tabelle 2 stabil.

4 Reale Testflüge

Bei dem Testsystem handelt es sich um eine gefesselte Drohne, bei der die Stromversorgung aber auch die Datenübertragung über ein zu einem Hangar geführtes Kabel dargestellt wird. Das Kabel wird über ein mechatronisches System nachgeführt und gestrafft, damit dieses im Flug nicht durchhängt.

Das System fungiert als fliegende Überwachungskamera. Bei einem Testablauf wird ein Einbruch simuliert. Der Hangar öffnet sich, die gefesselte Drohne hebt ab und fliegt zu der Position des Einbruchs, um diesen zu filmen. Anschließend schwebt die Drohne zurück über den Hangar und beginnt den Landeanflug. In einer Höhe von ca. 13m ist die Landemarkierung in dem Kamerabild ersichtlich und es beginnt das präzise Landen auf dem Hangar.

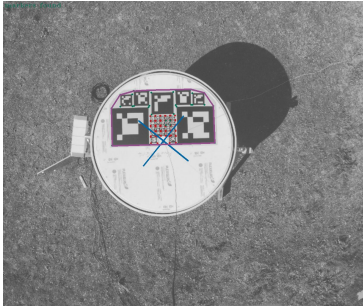


Abbildung 4.1: Sicht der Landekamera im Flug

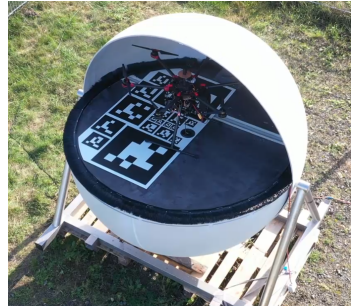


Abbildung 4.2: Kugelhangar beim Schließvorgang

Durch die Dimensionierung der Drohne und des Hangars darf die Abweichung von der anvisierten Landeposition eine absolute Abweichung von ca. 20cm nicht überschreiten, da dieser ansonsten nicht schließen kann.

Im Folgenden sollen die Ergebnisse von 11 Testflügen untersucht werden, die mit dem System unternommen wurden.

Es zeigt sich eine Standardabweichung von 0,045m bei einem Erwartungswert von 0,039m Abweichung. Die maximale Abweichung der Landeposition beträgt 0.104m. Es wurde mit Windgeschwindigkeiten von bis zu 7m/s geflogen.

5 Zusammenfassung

In diesem Beitrag wurde ein optisches Verfahren für die Präzisionslandung von automatischen fliegenden Systemen vorgestellt.

Dieses Verfahren ermöglicht eine hohe Verfügbarkeit, indem es sich durch das automatische Einstellen der Software- und Hardwareparameter an die Belichtungszustände, die bei einem Einsatz im Außenbereich vorzufinden sind, anpasst.

Das System ist derart redundant aufgebaut, so dass es sehr robust gegen partiellen Verschleiß oder Verdeckung von Markern beispielsweise infolge der Witterung reagiert. Ein Ausfall von einzelnen Markern kann vom System erkannt und gemeldet werden.

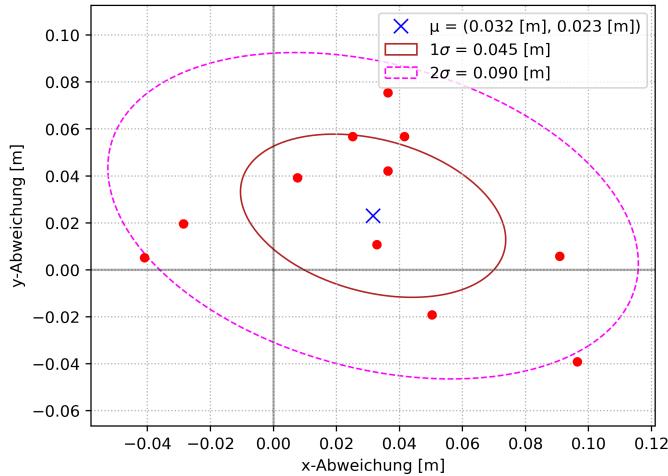


Abbildung 4.3: Positionsabweichung der Drohne von der Ziellandeposition

Durch die Prüfung der Kriterien für die Positionsgüte kann das System eine Rückmeldung über die eigene Integrität geben. Anhand der Positionsgüte wird zwischen zwei Verfahren gewählt, um die Landeplatz-Position in das Flugsteuerungssystem einzuspeisen. Das erste Verfahren kommt bei schlechter Sichtbarkeit der Landemarkierung zum Einsatz, wie es in größeren Höhen der Fall sein kann und integriert Sensormessungen des Flugsteuerungssystem in die Positionslösung, um eine robuste Positionslösung zu erhalten. Das zweite Verfahren kommt bei einer guten Sichtbarkeit der Landemarkierung zum Einsatz, wie es auf den letzten Metern des Landeanfluges der Fall ist. Hier kann die Positionsabweichung direkt aus der Markierung abgeleitet werden und der Fokus liegt auf einer hohen Präzision um ein zentimetergenaues Landen zu ermöglichen.

Simulationsergebnisse zeigen bei der Positionslösung auf dem für die Landepräzision besonders relevanten Höhenband von 0m-1m einen Erwartungswert von 0.0022m mit einer Standardabweichung von 0.0065m.

Elf reale Flugversuche unter Windeinfluss mit bis zu 7m/s zeigten hierbei eine maximale Abweichung der Landeposition von 0.104m. Der Erwartungswert liegt bei 0,039m bei einer Standardabweichung von 0,045m.

Das System eignet sich somit zum automatischen Landen in einem Dronenhangar, der ganzjährig und zu jeder Tageszeit betrieben wird und findet somit Einsatz bei verschiedenen Projekten der IAV GmbH.

Literatur

1. F. Zimmermann, C. Eling, L. Klingbeil, and H. Kuhlmann, "Precise Positioning of Uavs - Dealing with Challenging Rtk-Gps Measurement Conditions during Automated Uav Flights," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42W3, pp. 95–102, Aug. 2017.
2. T. Kos, I. Markežic, and J. Pokrajcic, "Effects of multipath reception on gps positioning performance," in *Proceedings ELMAR-2010*, 2010, pp. 399–402.
3. S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, p. 2280–2292, 06 2014.
4. M. Sezgin *et al.*, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic imaging*, vol. 13, no. 1, pp. 146–168, 2004.
5. S. Suzuki and K. be, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32 – 46, 1985. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0734189X85900167>
6. S.-T. Wu, A. C. G. d. Silva, and M. R. G. Márquez, "The Douglas-peucker algorithm: sufficiency conditions for non-self-intersections," *Journal of the Brazilian Computer Society*, vol. 9, pp. 67 – 84, 04 2004. [Online]. Available: <http://www.scielo.br/scielo.php?script=sci-arttext&pid=S0104-65002004000100006&nrm=iso>
7. J. Ferrão, P. Dias, and A. Neves, "Detection of aruco markers using the quadrilateral sum conjuncture," *Lecture Notes in Computer Science*, vol. 10882, pp. 363–369, 06 2018.

8. S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and R. Medina-Carnicer, "Generation of fiducial marker dictionaries using mixed integer linear programming," *Pattern Recognition*, vol. 51, pp. 481 – 491, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0031320315003544>
9. N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
10. N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, Sep 2004, pp. 2149–2154.