GENERISCHER ANSATZ ZUR KANTENBASIERTEN LANDEBAHNDETEKTION IN HOCHAUFLÖSENDEN FARBBILDERN

Stephan Blokzyl und Matthias Vodel und Wolfram Hardt
Technische Universität Chemnitz
Fakultät für Informatik
Professur Technische Informatik
D-09111 Chemnitz
{stephan.blokzyl,matthias.vodel,wolfram.hardt}@cs.tu-chemnitz.de

Kurzfassung

Die genaue Positionsmessung eines Fluggerätes referenziert zur Landebahn ist vor allem in der letzten Phase des Landeanflugs (Final Approach) von großer Bedeutung. Neben konventioneller Navigationssensorik können elektro-optische Sensoren zur Detektion und Lagebestimmung von Landebahnen im finalen Landeanflug verwendet werden.
Der Beitrag stellt ein generisches, klassifikationsfreies Verfahren zur Erkennung von Landebahnen in Farbbildern vor. Dabei steht die Extraktion der linken und rechten Landebahnbegrenzung im Vordergrund. Durch kammartige Suche im gesamten Abbildungsraum werden lokale Farbunstetigkeiten detektiert und zu Linien gruppiert. Anschließend wird der Suchraum eingeschränkt und die Liniendetektionen lokal verbessert (Backloop). Der vorgestellte Algorithmus ist echtzeitfähig und robust über einem großen Dynamikbereich.

Schlüsselwörter Landebahnerkennung, Farbbildverarbeitung, Echtzeitverarbeitung, Eingebettete Systeme, Unbemannte Flugsysteme

1. EINFÜHRUNG

Sicherheitskritische Systeme, die in der Luftfahrt eingesetzt werden, müssen besonderen Anforderungen bezüglich Robustheit und Betriebssicherheit genügen. Dies umfasst die gesamte System- und Verarbeitungskette, ausgehend von der Sensorik, über die Datenverarbeitung, bis zur Aktorik.
In der bemannten sowie unbemannten Luftfahrt sind aufgrund limitierter Energieressourcen und Nutzlast harte Kompromisse zwischen Leistungsaufnahme, Gewicht und Bauraum zu beachten. Dementsprechend resultiert ein klarer Entwicklungstrend zu flexibel einsetzbaren, applikationsspezifisch optimierbaren Komponenten. Im Bereich der Sensorik sind hier vor allem elektrooptische (EO) Systeme in Kombination mit eingebetteten Systemen die Wahl. Diese eignen sich sehr gut für vielfältige Explorations- und Überwachungsaufgaben. EO Sensoren sind leicht, kostengünstig und bieten durch den Einsatz von stereoskopischen Techniken zusätzlich Tiefeninformation. Jedoch stellt die Verarbeitung von hochauflösenden Bilddaten eine besondere Herausforderung dar, vor allem im Kontext eingebetteter Systeme und unter Berücksichtigung von Echtzeitaspekten.

2. STATE OF THE ART

Um dem großen Verarbeitungsaufwand von hochauflösenden Bilddaten zu begegnen, werden Hardwarebeschleunigungsstrategien verfolgt. So haben insbesondere Field Programmable Gate Arrays (FPGAs) bereits in verschiedenen Arbeiten ihr Beschleunigungspotential für Sensordatenverarbeitungsszenarien bewiesen [1] [2] [3]. Sie bieten maximale Flexibilität durch Rekonfigurierbarkeit sowie eine hohe Verarbeitungsleistung durch Anpassung an die zugrundeliegende Applikation.
FPGAs werden auch für Bilddatenprozessierung [4] eingesetzt. Hardwareimplementierungen von Low-Level-Bildverarbeitungsmodulen, wie zum Beispiel die Korrektur von Linsenverzerrungen [5], Sobelfilterung [6] oder Optical-Flow-Verfahren [7], dokumentieren die Vielseitigkeit rekonfigurierbarer Hardware. Anwendungen, wie die Klassifikation von Objekten [8] [9] und Gesichtern [10], sowie Tracking- und Filtermethoden [11] [12] zeigen, dass auch komplexere Schritte der Bildverarbeitungskette in Hardware umsetzbar sind.
Diese Erkenntnisse macht sich das in [13] vorgestellte Konzept zur Hardwarebeschleunigung für Echtzeitverarbeitung von Bilddaten zu nutze. Es wird Datenparallelisierung verwendet, um die Menge der pro Verarbeitungseinheit zu behandelnden Daten zu reduzieren. Des Weiteren wird eine Task Parallelisierung eingeführt, die verschiedene, schwache Musterdetektoren miteinander zu einem robusten Detektor verknüpft.
Die verwendeten Detektoren werden in Primär- und Sekundärdetektoren eingeteilt. Der Primärdetektor

extrahiert die Segmente im Bild, die das gesuchte Objekt am stärksten charakterisieren. Er ist somit hochgradig anwendungsspezifisch. Des Weiteren ist er sehr sensitiv parametrisiert, sodass keine Detektionen (*true positives*) verloren gehen. Mithilfe der Ergebnisse der Sekundärdetektoren werden die primären Low-Level-Detektionen bewertet. Die konsolidierten Primärdetektionen sind abschließend in einem Nachverarbeitungsschritt Grundlage für die finale Strukturextraktion.

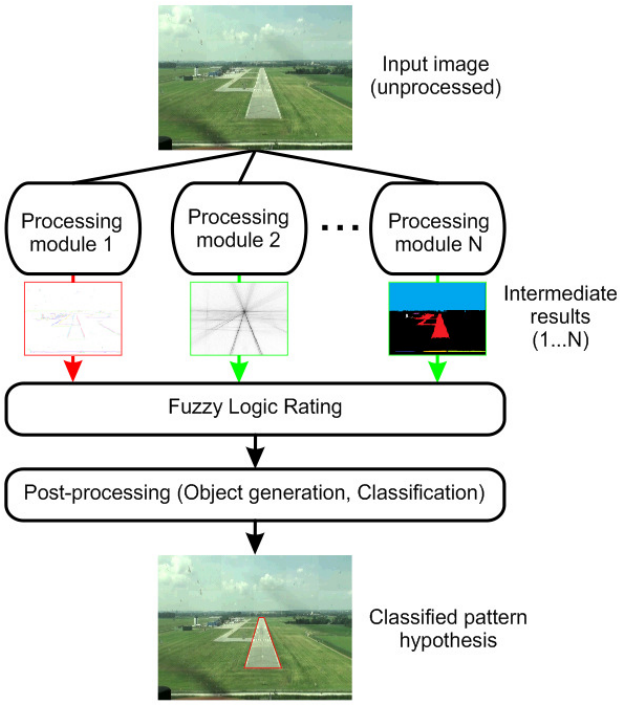


Abb. 1: Task Parallelisierung mit N parallelen Verarbeitungsmodulen (vgl. [13])

Dieses Konzept adaptiert die sequenzielle Bildverarbeitungskette an ein paralleles Hardwaredesign. Es nutzt das volle Beschleunigungspotential rekonfigurierbarer Hardware und steigert Robustheit und Qualität der Gesamtdetektion.

3. LANDEBAHNDETEKTION IN FARBBILDERN

Auf der Grundlage der Task Parallelisierung wird in diesem Beitrag ein Verfahren vorgestellt, welches die Landebahnstruktur mit Hilfe von Farbliniendetektoren erkennt. Die Farblinien werden von einem parallelisierten Primärdetektor (Datenparallelisierung) und ein Sekundärdetektor extrahiert die Horizontlinie.
Die Zwischenergebnisse der Primär- und Sekundärdetektoren werden in einem Fusionsschritt zusammengeführt und bewertet. Ein Nachverarbeitungsschritt bestimmt die Landebahnposition im Bildraum und ermittelt eine Gesamtgüte des Detektionsergebnisses.

3.1. Farbkantendetektion (Primärdetektor)

Für die Kantendetektion in Farbbildern steht eine Vielzahl leistungsfähiger Algorithmen zur Verfügung. Eine ausführliche Übersicht gibt [14]. Der Einsatz von statistischen Methoden [15], Tensor Voting [16] oder Quaternionfaltung [17] weist auf die Komplexität vieler Verfahren hin. Sie eignen sich im Allgemeinen nicht, sinnvoll in Hardware implementiert und beschleunigt zu werden.
Einen geeigneten Algorithmus stellt Lacroix in [18] vor. Dieser berechnet ein Farbgradientenfeld, das jedem Bildpunkt $\boldsymbol{p} = ({}^i x, {}^i y)^T$ einen gerichteten, zweidimensionalen Farbgradienten $G(x^i, y^i)$ zuordnet.

$$G({}^i x, {}^i y) = \left(G_x; G_y\right)^T$$

$$\text{mit } G_x = s \cdot d\left(\hat{\mathrm{A}}_{\boldsymbol{r}}; \hat{\mathrm{A}}_{\boldsymbol{l}}\right) \text{ und } G_y = s \cdot d\left(\hat{\mathrm{A}}_{\boldsymbol{u}}; \hat{\mathrm{A}}_{\boldsymbol{o}}\right)$$

$\hat{\mathrm{A}}_{\boldsymbol{o}|\boldsymbol{u}|\boldsymbol{l}|\boldsymbol{r}}$ ist der gemittelte Farbwert der Bereiche oberhalb, unterhalb, links bzw. rechts des Punktes $\boldsymbol{p}$ (siehe Abb. 2). $d(\boldsymbol{a}, \boldsymbol{b})$ ist die Euklidische Distanz der Vektoren $\boldsymbol{a} = (a_1, \dots, a_n)^T$ und $\boldsymbol{b} = (b_1, \dots, b_n)^T$, und der Parameter s ergibt sich aus:

$$s = \begin{cases} -1 & if \quad (a_1 < b_1 \,|a_i = b_i;\, \forall i = 1, \dots, j-1 < n) \wedge (a_j < b_j) \\ 0 & if \quad a_i = b_i;\, \forall i = 1, \dots, n \\ 1 & if \quad (a_1 > b_1 \,|a_i = b_i;\, \forall i = 1, \dots, j-1 < n) \wedge (a_j > b_j) \end{cases}$$

Für die hier verwendete *RGB*-Repräsentation der Bildpunktfarbe nutzt Lacroix ein lexikographisches Ordnungsmodell zur Bestimmung von s.

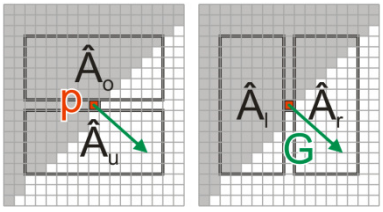


Abb. 2: Felder für die Farbwertmittelung [18]

Die Komponenten G_x und G_y des Farbgradienten G beschreiben den Sprunganteil in x- und y-Richtung im Bild. Für die Sprungstärke N folgt:

$$N({}^i x, {}^i y) = \sqrt{G_x^2 + G_y^2} = \sqrt{d\left(\hat{\mathrm{A}}_{\boldsymbol{r}}; \hat{\mathrm{A}}_{\boldsymbol{l}}\right)^2 + d\left(\hat{\mathrm{A}}_{\boldsymbol{u}}; \hat{\mathrm{A}}_{\boldsymbol{o}}\right)^2}$$

Je größer der Betrag N ist, desto stärker ist der Kantenpunkt im Farbbild. Bildpunkte mit einer großen Gradientenkomponente in x-Richtung können Bestandteil einer vertikalen Kantenlinie sein. Punkte mit großem G_y sind Kandidaten für eine horizontale Kantenlinie.
Die Abbildung der Landebahn ist während des finalen Landeanfluges als signifikantestes Objekt und annähernd senkrecht zum Horizont im Bildraum zu erwarten. Somit sind lange, vertikale Kantenlinien gute Kandidaten für eine Landebahnbegrenzung. Für die Detektion einer vertikal orientierten Farbkante genügt die Anwendung der $\hat{\mathrm{A}}_{\boldsymbol{r}}|\hat{\mathrm{A}}_{\boldsymbol{l}}$-Maske. Sie wird zeilenweise über das Bild verschoben und für jeden Punkt $\boldsymbol{p}$ der zugehörige Gradient $G_x({}^i x, {}^i y)$ berechnet. Die Dimensionen der Maskenfelder $\hat{\mathrm{A}}_{\boldsymbol{r}}$ und $\hat{\mathrm{A}}_{\boldsymbol{l}}$ sind mit B in x- und 1 in y-Richtung festgelegt (vgl. Abb. 4).

Die Abb. 3 zeigt das Gradientenfeld des Originalbildes (oben links). Farbkanten mit einem Orientierungsbetrag größer $\pi/2$ sind hell codiert ($255 \geq I({}^{i}x, {}^{i}y) > 127$). Kanten mit einem Orientierungsbetrag kleiner $\pi/2$ sind dunkel dargestellt ($127 > I({}^{i}x, {}^{i}y) \geq 0$). Die Stärke des Farbsprungs ist proportional zur absoluten Helligkeitsdifferenz zum mittleren Grauwert 127. Es ist gut zu erkennen, dass sich die vertikalen Landebahnkanten deutlich ausprägen.

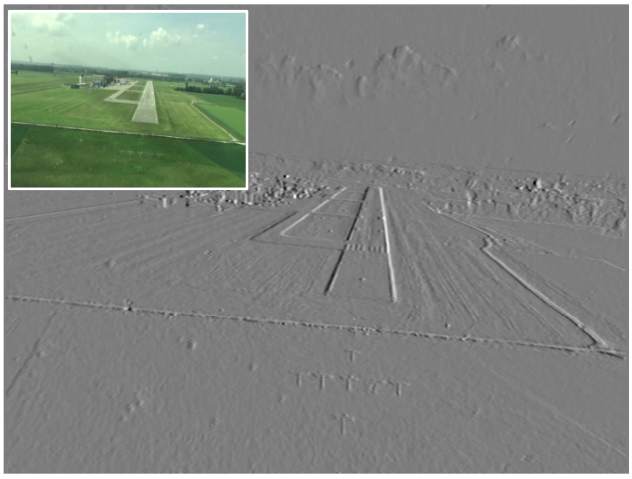


Abb. 3: Original- (oben links) und Falschfarbendarstellung des zugehörigen Gradientenfeldes

Alle extrahierten Kantenpunkte werden durch die Eigenschaften Orientierung $\varphi({}^{i}x, {}^{i}y) \in \{-1; +1\}$, Position $({}^{i}x, {}^{i}y)$, und Kantenstärke N im Bildraum beschrieben. Für die Kantenpunkte mit einem Orientierungsbetrag größer $\pi/2$ gilt $\varphi = +1$ und für alle anderen $\varphi = -1$. Somit lassen sich die Detektionen auf Basis ihrer Orientierung φ in zwei Klassen aufteilen. Nachfolgend werden diese als φ^{+} und φ^{-} bezeichnet. Beide Ergebnismengen werden parallel weiterverarbeitet.

3.1.1. Liniensegmentierung

Bei der Analyse des Gradientenfeldes in Abb. 3 fallen die hohen Kantenstärken der Landebahnränder auf. In Abhängigkeit der Farbkonstellation, können die Detektionen des einen Landebahnrandes der Klasse φ^{+}, und die des anderen der Klasse φ^{-} zugeordnet werden. Die Signifikanz der Kantenpunkte begründet im Folgenden die Strategie, ausschließlich Farbkanten mit dem pro Zeile stärksten Gradientenbetrag zu betrachten.
Bei der Liniensegmentierung werden solche Farbkantenpunkte zu Kantenlinien zusammenfasst, die im Bezug zur Linie

- zur gleichen Klasse (φ^{+} bzw. φ^{-}) gehören,
- in der richtigen Richtung liegen ($\alpha \to 0$) und
- einen maximalen Abstand ε nicht überschreiten.

Dabei werden zeilenweise Gradienten in x-Richtung traversiert und jeweils der mit dem größten Betrag selektiert. Existiert in der folgenden Zeile ein Punkt, dessen Gradientenbetrag in seiner Zeile maximal ist, und der die oben genannten Kriterien erfüllt, so wird die Linie um diesen erweitert. Wurde noch keine Linie gefunden, werden zwei aufeinanderfolgende Punkte mit zeilenmaximalem Gradientenbetrag als Startlinie verwendet.
Die Abb. 4 illustriert die Liniensegmentierung. Der Kantenpunkt P genügt dem Abstands- und Richtungskriterium und besitzt die gleiche Orientierung φ. Somit wird das Liniensegment in (a) um P erweitert (b). Es ergibt sich ein größeres Liniensegment von S nach E_{neu}.
Versuchsreihen haben gezeigt, dass nicht jede Bildzeile traversiert werden muss. Wird die Schrittweite S empirisch auf 2 festgelegt, zeigen sich robuste, schnell konvergierende Ergebnisse. Dieser Effekt ergibt sich durch die pixelbasierte Diskretisierung von Bilddaten. Eine Abweichung um ein Pixel in x-Richtung bei einem $\Delta y = 1$ (Zeilenschrittweite 1) führt zu einer Winkeldifferenz von $\alpha = 0.79rad$. Die gleiche Differenz in x bei $\Delta y = 2$ ergibt eine Winkeldifferenz von $\alpha = 0.46rad$. Die Abweichung in x-Richtung wird also im zweiten Fall weniger schlecht bewertet. Dies führt zu einer Robustheitssteigerung bei $S = 2$. Untersuchungen haben gezeigt, dass sich die Ergebnisse wieder verschlechtern, wenn $S \gg 2$ gesetzt wird.

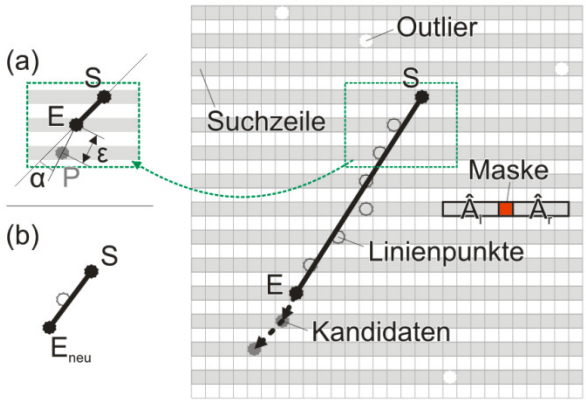


Abb. 4: Gruppierung von Sprungpunkten zur Linie bei Verwendung nur jeder zweiten Bildzeile

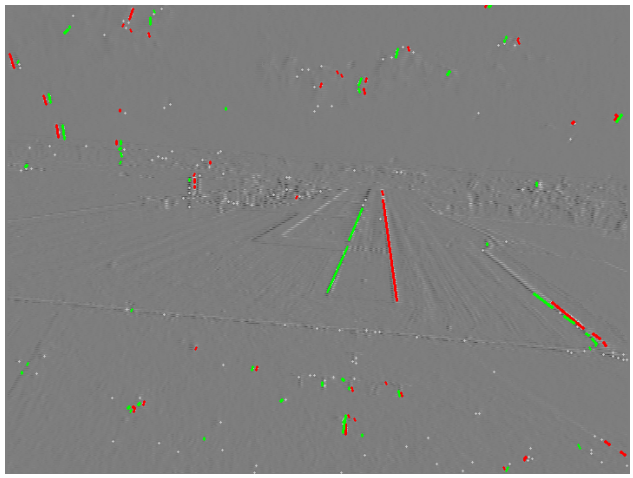


Abb. 5: Segmentierte Linien der Klassen $L_{\varphi^{+}}$ (rot) und $L_{\varphi^{-}}$ (grün)

Im Ergebnis der Liniensegmentierung entstehen zwei Linienmengen, eine für die φ^-- und eine für die φ^+-Klasse.

$$L = \{L_{\varphi^-}; L_{\varphi^+}\}$$

$$\text{mit } L_{\varphi^-} = \{l_0^-; \dots; l_m^-\}$$

$$\text{und } L_{\varphi^+} = \{l_0^+; \dots; l_n^+\}; \quad m, n \in \mathbb{N}_0.$$

Die Linien der Mengen L_{φ^-} und L_{φ^+} sind im Bildkoordinatensystem charakterisiert durch ihren

- Startpunkt $S = ({}^i x_S, {}^i y_S)$,
- Endpunkt $E = ({}^i x_E, {}^i y_E)$,
- ihre Orientierung ${}^i\gamma = \tan^{-1}(({}^i x_E - {}^i x_S)/({}^i y_E - {}^i y_S))$
- und die Länge ${}^i R$.

3.1.2. Lokale Linienverbesserung

Alle Linien der Menge L sind das Ergebnis einer Suche im vollständigen, globalen Bildraum. Wie die Abb. 5 zeigt, werden die Farbkanten nicht immer in ihrer Gesamtheit segmentiert. Dies resultiert aus der Verwendung der Kantenpunkte mit dem größten Gradientenbetrag pro Suchzeile.
Die gefundenen Linien $l_i^{\pm}$ sind daher Grundlage zur Bestimmung eines lokalen Suchraums (vgl. Abb. 6). In einem als *Backloop* bezeichneten Verfahren werden die Bildregionen erneut analysiert, deren Charakteristik das Vorhandensein einer gültigen Detektion vermuten lässt [19].

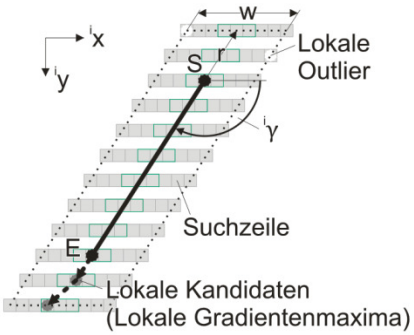


Abb. 6: Lokaler Suchraum in Bildkoordinaten

Ausgehend von Start- (S) bzw. Endpunkt (E) wird in Richtung ${}^i\gamma$ und Entfernung $\pm r$ eine obere bzw. untere Suchgrenze bestimmt. In diesem lokalen Suchraum, mit der Breite w, wird mit Hilfe des in Abschnitt 3.1.1 beschriebenen Verfahrens erneut segmentiert. Zur Steigerung der Robustheit können bei der Linienverbesserung Lücken übersprungen werden. Lassen sich Kantenpunkte nach Überspringen von Zeilen mit Ausreißern (Outlier) in hinreichender Güte anfügen, wird die Linie entsprechend verlängert. Die Anzahl der ausgelassenen Zeilen darf ein Maximum nicht überschreiten. Das Auslassen von Zeilen ist nur zulässig, wenn eine minimale Anzahl von Punkten anschließend hinzugefügt werden kann.
Empirische Untersuchungen haben gezeigt, dass anstatt der Erweiterung der Linien durch Anfügen eines neuen Punktes, die Bildung einer Regressionsgrade robuster ist. Für die Berechnung dieser linearen Gerade werden alle einer Linie zugeordneten Punkte verwendet. Die Start- und Endpunkte der verbesserten Farblinie werden durch die Lotfußpunkte der originären Start-und Endpunkte auf der Regression bestimmt.

Abb. 7: Lineare Regression zur Linienverbesserung

Im Ergebnis der lokalen Linienverbesserung entstehen vollständig erfasste Liniensegmente (vgl. Abb. 8).

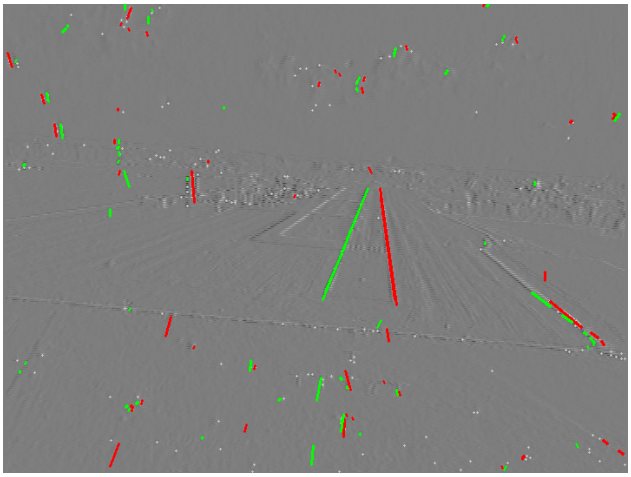


Abb. 8: Ergebnis der Farblinienverbesserung

3.2. Horizontextraktion (Sekundärdetektor)

Als Horizont wird eine Linienapproximation in Bildkoordinaten verstanden, die Boden und Himmel voneinander trennt. Unter der Annahme, dass der Horizont im oberen Bildbereich zu erwarten und heller als der Boden ist, kann er mittels Bildbinarisierung extrahiert werden.
Der Farbvektor $\boldsymbol{c}_p = (r_p; g_p; b_p)^T$ jedes Bildpunktes $\boldsymbol{p}$ wird dabei auf eine Menge $\{0; 1\}$ abgebildet (siehe Gleichung (1)). Alle Pixel, die zum Himmel (Vordergrund) gehören, sind mit 1 und alle anderen mit 0 (Hintergrund) markiert.

$$BIN({}^i x, {}^i y) = \begin{cases} 0 & if \quad T > \dfrac{2 \cdot R({}^i x, {}^i y) + 4 \cdot G({}^i x, {}^i y) + B({}^i x, {}^i y)}{8} \\ 1 & if \quad T \leq \dfrac{2 \cdot R({}^i x, {}^i y) + 4 \cdot G({}^i x, {}^i y) + B({}^i x, {}^i y)}{8} \end{cases} \qquad (1)$$

Zur Schließung von Löchern im Vordergrund wird das Binärbild BIN morphologisch behandelt. Dazu wird erst eine Dilatation ($\oplus$) und anschließend eine Erosion ($\ominus$) mit einem kreisförmigen Strukturelement K durchgeführt.

$$BIN_{closed} = (BIN \oplus K) \ominus K$$

Für das Einpassen der Linie, die den Verlauf des Horizontes im Bild approximiert, werden spaltenwei-

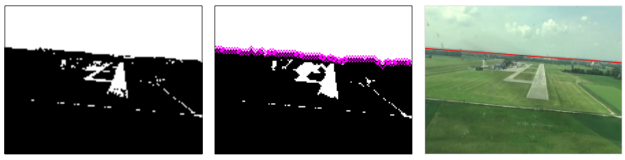
se alle 1-0-Übergänge gesucht. Sie sind Grundlage für die Berechnung einer linearen Regressionsgerade ${}^{i}y_h = f_h\,({}^{i}x) = {}^{i}m_h \cdot {}^{i}x + {}^{i}n_h$ und der Horizontorientierung ${}^{i}\theta_h = tan^{-1}({}^{i}m_h)$, mit:

$$ {}^{i}m_h = \frac{\sum_{j=1}^{n}({}^{i}x_j - {}^{i}\bar{x})({}^{i}y_j - {}^{i}\bar{y})}{\sum_{k=1}^{n}({}^{i}x_k - {}^{i}\bar{x})^2}; \quad {}^{i}n_h = {}^{i}\bar{y} - {}^{i}m_h \cdot {}^{i}x $$

Für eine robuste Funktion des Horizontdetektors werden bestimmte Vorbedingungen angenommen:

(a) Der Horizont befindet sich im oberen Bildbereich und ist sinnvoll durch einen Linie approximierbar (keine starken urbanen Strukturen oder Geländeerhebungen).
(b) Das System wird tagsüber und unter Wetterbedingungen betrieben, in denen der Horizont heller als der Boden ist.

Sind diese Voraussetzungen erfüllt, kann die Horizontextraktion als Sekundärdetektor eingesetzt werden. Die ermittelte Horizontposition wird zur Bewertung der gefundenen, lokal verbesserten Farblinien benützt.



[a] [b] [c]

Abb. 9: Ergebnis der Horizontextraktion ([a] Binärbild, [b] Detektionen im geschlossen Binärbild, [c] Horizontlinie im Originalbild)

3.3. Datenfusion und Konturbestimmung

Die Ergebnisse des Primär- und Sekundärdetektors werden in einem Fusionsschritt miteinander verknüpft. Auf Grundlage der Features der Primärdetektionen $\boldsymbol{x}^{(1)} = (S, E, {}^{i}\gamma, {}^{i}R)^T$ (vgl. Kapitel 3.1.1) und der Eigenschaften der Sekundärdetektion $\boldsymbol{x}^{(2)} = ({}^{i}m_h, {}^{i}n_h)^T$ erfolgt eine Evaluation aller gefundenen Farblinien nach [13].
Die Features ${}^{i}\gamma$ und ${}^{i}R$ der Linien werden mit der Potentialfunktion $\mu(x_i)$ entsprechend Gleichung (2) bewertet. Der Erwartungswert $E({}^{i}\gamma) = tan^{-1}({}^{i}m_h) + \pi/2$ beschreibt die Orientierung der Landebahnränder auf Basis der Horizontorientierung. Der Erwartungswert $E({}^{i}R) = \max_{0 \le k \le |L|}({}^{i}R_k)$ leitet sich aus der maximalen Linienlänge aller gefundenen Linien ab.

$$ \mu(x_i) = \frac{1}{1 + \left(\frac{|E(x_i) - x_i|}{c}\right)^d} \qquad (2) $$

Diese Modellierung ist zulässig, da die Landebahn in der finalen Anflugphase annähend senkrecht zum Horizont liegt, und ihre Ränder von zwei der längsten Farblinien im Bildraum repräsentiert werden (vgl. Abschnitt 3.1).
Neben Orientierung und Länge werden auch die Positionen der Linienstart- und Endpunkte analysiert. So können Linien vernachlässigt werden, deren Start oder Endpunkt im Bildraum oberhalb der ermittelten Horizontlinie liegt. Die Abb. 10 illustriert (nicht maßstabsgerecht) die verwendete Bewertungsfunktion. Dabei werden ${}^{i}y_S$ bzw. ${}^{i}y_E$ mit dem ${}^{i}y_h$ der jeweiligen Bildspalte verglichen.

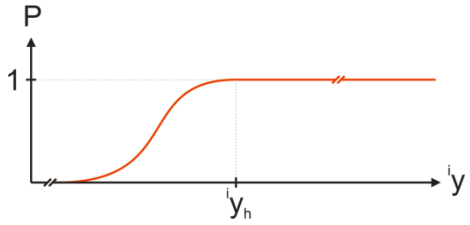


Abb. 10: Bewertungsfunktion für Linienposition

Die Verwendung der Potentialfunktion zur Fuzzy-basierten Detektionsbewertung ermöglicht die Modellierung der Unsicherheit der Primär-und Sekundärdetektoren (Linienextraktion, Horizontmessung). Ihre Unschärfe lässt Ungenauigkeiten und Dynamiken zu, wie sie im Allgemeinen bei nichtsynthetischen Szenen zu erwarten sind (vgl. [13]).
Alle mit ω_i gewichteten Featurebewertungen $\mu(x_i)$ werden mit einem *Gewichteten, kompensatorischen Hamacheroperator* [20] zu einer Gesamtbewertung $\mu(\underline{x}^{(1)})$ für jede Linie fusioniert:

$$ \mu(\underline{x}^{(1)}) = \frac{\sum_{i=1}^{n}(\omega_i)}{\sum_{i=1}^{n}\left({}^{\omega_i}/_{\mu(x_i)}\right)} \quad . \qquad (1) $$

Im Ergebnis entsteht eine Menge von bewerteten Farblinien, die Grundlage für die abschließende Extraktion der Landebahnstruktur ist. Die Abb. 11 zeigt das Fusionsergebnis von Primär- und Sekundärdetektor. Die Sättigung der Farblinien codiert den Betrag der Gesamtbewertung $\mu(\underline{x}^{(1)})$. Je höher ihr Betrag, desto stärker ist die Farbsättigung.

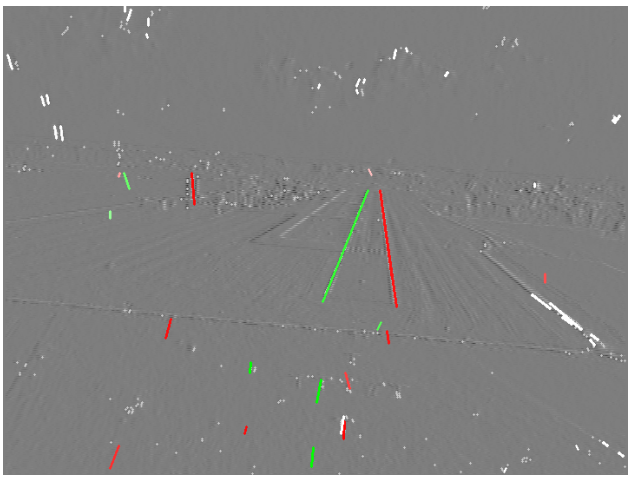


Abb. 11: Ergebnis der Fusion der Ergebnisse von Primär- und Sekundärdetektion

Für die Konturextraktion der Landebahn wird je eine Linie der Klasse φ^- und φ^+ mit der höchsten Gesamtbewertung $\mu_{\pm}(\underline{x}^{(1)})$ gewählt. Ihre Start- $(S_{\pm})$

und Endpunkte ($E_{\pm}$) werden zur Bestimmung einer Umrisslinie verwendet, die die eingeschlossene Fläche minimiert. Die resultierende Kontur beschreibt die Struktur und Position der Landebahn im Bild (siehe Abb. 13).
Um die Güte der resultierenden Gesamtdetektion interpretieren zu können, wird ein Indikator bestimmt, der alle Charakteristika der Gesamtdetektion in die Qualitätsbewertung einbezieht (vergleiche Abb. 12):

(a) Schiefheit der Parallelität ${}^{i}\delta_1$
(b) Winkeldifferenz ${}^{i}\delta_2$ und
(c) Längendifferenz $\left|{}^{i}R_{+} - {}^{i}R_{-}\right|$ der beschreibenden Farblinien

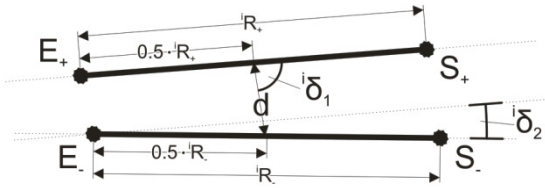


Abb. 12: Parameter zur Parallelitätsbewertung [19]

Auf Basis der Gleichung (2) werden für alle Eigenschaften (a) bis (c) die Potentiale u_1, u_2, und u_3 bestimmt. Der Parameter d ist dabei mit $d = 2$ definiert.

$$u_1 = \frac{1}{1 + \left(\frac{0.5 \cdot \pi - {}^{i}\delta_1}{c_{\delta_1}}\right)^2}$$

$$u_2 = \frac{1}{1 + \left(\frac{{}^{i}\delta_2}{c_{\delta_2}}\right)^2}$$

$$u_3 = \frac{1}{1 + \left(\frac{\left|{}^{i}R_{+} - {}^{i}R_{-}\right|}{c_R}\right)^2}$$

Mithilfe der Gleichung (3) wird für die Parallelität eine Fitness μ_{P} bestimmt.

$$\mu_{\mathrm{P}} = \frac{3}{\frac{1}{u_1} + \frac{1}{u_2} + \frac{1}{u_3}} \tag{3}$$

Zusammen mit $\mu_{+}\left(\underline{x}^{(1)}\right)$ und $\mu_{-}\left(\underline{x}^{(1)}\right)$ für die Linien der Klassen φ^{-} und φ^{+} kann eine Gesamtgüte μ_{total} der Detektion nach Gleichung (4) erfolgen.

$$\mu_{total} = \frac{2}{\frac{1}{2 \cdot \mu_{-}\left(\underline{x}^{(1)}\right)} + \frac{1}{2 \cdot \mu_{+}\left(\underline{x}^{(1)}\right)} + \frac{1}{\mu_{\mathrm{P}}}} \tag{4}$$

Die Parameter $c_{\delta_{1|2}}$ und c_R modellieren die Strenge (Form) der Potentialfunktion [13] und sind frei wählbar.
Der extrahierten Landebahnkontur wird im Ergebnis ein Vertrauensindikator im Intervall $[0; 1]$ zugeordnet. Strebt der Wert μ_{total} gegen 1, spricht dies für eine hohe Güte der Landebahndetektion. Ist μ_{total} kleiner einer anwenderfestgelegten Schwelle, ist die Korrektheit der Detektion in Frage gestellt.

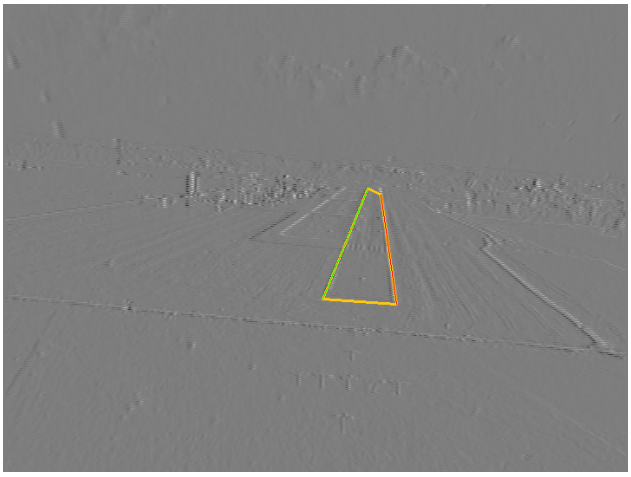


Abb. 13: Landebahnkontur-Overlay in der Falschfarbendarstellung des Gradientenfeldes ($\mu_{total} \cong 0.903$ mit $c_{\delta_1} = c_{\delta_2} = 3$, $c_R = 50$)

4. ERGEBNISSE UND AUSBLICK

Der in diesem Beitrag vorgestellte Ansatz zur kantenbasierten Landebahndetektion ist klassifikationslos und unabhängig von Landebahnmarkierungen oder zusätzlichen synthetischen Markern. Er arbeitet unabhängig von Farbton, Sättigung, Helligkeit und Kontrast im Bild, und demonstriert somit seine Anwendbarkeit in Umgebungen umfangreicher Dynamik. Ein Primärdetektor sucht nach Farblinien und ein Sekundärdetektor steuert die Auswahl der gefundenen Farblinien zur Objektgeneration. Dabei ist die Auswahl und Parametrisierung von Primär- und Sekundärdetektor sehr applikationsspezifisch (siehe Tab. 1). Die Einsatzbedingungen des Systems für eine Ladebahndetektion in der finalen Landeanflugphase lassen sich wie folgt zusammenfassen:

(a) Der Horizont nach den Kriterien entsprechend Kapitel 3.2 detektierbar.
(b) Der Anflugpfad ist gerade und parallel zur Landebahnorientierung.
(c) Die Landebahn ist signifikantestes Bildobjekt in der finalen Landeanflugphase.

Diese Kriterien deduzieren eine starke Spezialisierung der Methode. Untersuchungen zeigen hingegen, dass eine Vielzahl von Landeanflugszenarien genau diesen Randbedingungen genügen (siehe Abb. 14 und Abb. 15).
Dennoch ist für die robuste Funktion des Landebahndetektors die Güte der Horizontdetektion ausschlaggebend. Fällt sie aus, entfällt das Wissen über die Horizontposition und -orientierung und mit ihm die Grundlage zur Bewertung und Auswahl der Farblinien. Wird die Parallelitätsbewertung nach Gleichung (4) als zusätzliches Feature bei der Linienselektion eingesetzt, kann ein Ausfall oder eine

schlechte Detektionsqualität bei der Horizontextraktion kompensiert werden. Auch ein zeitliches Tracking der Landebahndetektion über mehrere Bilder, also die Anwendung von Vorwissen, kann die Robustheit des Verfahrens weiter steigern. Der Einsatz zusätzlicher Sekundärdetektoren (z.B. Landebahnmarkierungserkennung) verringert die Störanfälligkeit bei Existenz von signifikanten Strukturen im Bild, die nicht zur Landebahn gehören (z.B. Taxiways).

Parameter	Wert	Beschreibung
B	$3 \dots 21$	Suchmaskenbreite
r	$\frac{H}{2}$	Suchradius der lokalen Suchregion
N	2	Mindestanzahl der Punkte pro Linie
w	$5 \cdot B$	Breite der lokalen Suchregion
S	2	Linienschrittweite
α_{global}	$\frac{\pi}{8}$	Max. zulässige Winkeldifferenz, global
ε_{global}	5.0	Max. zulässiger Punktabstand, global
α_{lokal}	$\frac{\pi}{12}$	Max. zulässige Winkeldifferenz, lokal
ε_{lokal}	2.5	Max. zulässiger Punktabstand, lokal
G	3.0	Max. zulässige Lücke in Linie
O	O_{bgr}	Verwendete Lexikographische Ordnung
ω_i	1.0	Gewichte für Hamacheroperator

Tab. 1: Empirisch ermittelte Systemparameter

Ferner diskutiert Lacroix in [18], dass die Verwendung des RGB-Farbraums zur Bestimmung des Farbgradientenfeldes nicht optimal ist. Er zeigt an einem Beispiel, dass optisch starke Farbsprünge bei einer bestimmten lexikographischen Ordnung O einen kleineren euklidischen Abstand als andere besitzen, die weniger deutlich im Bild ausprägt sind. So kann beispielsweise die Verwendung von O_{rgb} ganz andere Resultate erzeugen als der Einsatz von O_{bgr}. Die Ergebnisse der Prozessierung der in dieser Arbeit verwendeten nichtsynthetischen Szenen zeigen jedoch, dass der Einfluss der lexikographischen Ordnung vernachlässigbar klein ist. Denn für die lokale Linienverbesserung werden ausschließlich regionale Farbunstetigkeiten betrachtet. Die globale Signifikanz des lokalen Gradientenbetrags spielt dabei keine Rolle.

Der Ansatz zur kantenbasierten Landebahndetektion ist aufgrund der Determiniertheit seiner maximalen Laufzeit (Worst Case Execution Time, WCET) echtzeitfähig. Die Gradientenfeldberechnung (vgl. Abb. 3) ist ausschließlich von der Dimension (Bildbreite W und -höhe H) des Eingangsbildes abhängig und benötigt $H \times (W - 2 \cdot w)$ Berechnungen. Bei der lokalen Linienverbesserung werden im schlechtesten Fall $\sqrt{W^2 + H^2}$ Suchschritte für jede zu verbessernde Linie ausgeführt (z.B. bei einer Linie entlang der Bilddiagonalen und hinreichend groß gewähltem r, siehe Abb. 6). Die Anzahl der zu verbessernden Linien ist mit maximal $H/2$ gegeben. In diesem Fall setzen sich alle Linien der Menge L aus nur zwei Punkten zusammen. Die Berechnung der Umrisslinie, die abschließend die Kontur der Landebahn beschreibt, genügt ebenfalls einem deterministischen Verfahren. Diese Eigenschaften führen zu einer genauen Voraussagbarkeit der WCET. Sie ist unabhängig von Bildcharakteristik oder -inhalt. Diese Echtzeitfähigkeit, die Indikation der Detektionsgüte und die Robustheit über einem großen Dynamikbereich bescheinigen dem vorgestellten Verfahren Zertifizierbarkeit.

Der Ansatz ist bereits prototypisch in Software implementiert und verarbeitet Full-HD Farbbilddaten (1080p) mit einer durchschnittlichen Framerate von 6Hz (Workstation mit AMD Phenom II X4, 4GB RAM). Diese Verarbeitungsraten sollen durch den Einsatz von rekonfigurierbarer Hardware (FPGAs) signifikant erhöht werden. Aus diesem Grund verwendet der Ansatz eine maskenbasierte Suche, die sich durch sehr gute Hardwareumsetzbarkeit auszeichnet. Die pixel- und zeilenweise Akquise, wie sie unabhängig von Abtastart und Interface des verwendeten EO Sensors vorliegt, unterstützt diese effiziente Realisierung in Hardware. Dabei puffert der FPGA so viele Pixelwerte, bis die Suchmaske soweit gefüllt ist, dass der erste Gradient berechnet werden kann. Mit jedem weiteren ankommenden Pixelwert wird der nächste Gradient bestimmt. So ist ein wesentlicher Bestandteil des Primärdetektors ohne Iterationen und Rekursionen effizient umsetzbar. Des Weiteren ist die lokale Linienverbesserung für jede globale Liniendetektionen gleich und unabhängig voneinander ausführbar (siehe Kapitel 3.1.2). Bei einer FPGA-Implementierung werden diese Operationen so häufig instanziiert, wie es die Logikressourcen der verwendeten Hardware zulassen. Im Ergebnis wird die lokale Linienverfeinerung der globalen Liniendetektionen signifikant parallelisiert und beschleunigt. Auch der Sekundärdetektor sowie der Fusionsschritt sind so gestaltet, dass eine Implementierung in Hardware uneingeschränkt möglich ist. Der kantenbasierte Landebahndetektor nutzt somit die in [13] eingeführte Daten und Task Parallelisierung und ist vollständig auf einem FPGA realisierbar.

5. ZUSAMMENFASSUNG

Das Paper führt ein generisches Verfahren zur Erkennung von Landebahnkonturen in Farbbildern ein. Farbunstetigkeiten werden zu Linien gruppiert, auf deren Basis der Suchraum eingeschränkt und erneut prozessiert wird. Ziel der lokalen Analyse ist die Verbesserung der globalen Liniendetektion (Back-

loop). Alle gefundenen Linien werden bewertet und die geeignetsten Kandidaten für die linke und rechte Landebahnbegrenzung ausgewählt. Diese sind Grundlage für die Kontur- und Positionsbestimmung der Landebahn im Bildraum. Eine abschließende Bewertung der Gesamtdetektion mit einem Vertrauensindikator schätzt die Zuverlässigkeit der Landebahnerkennung ein.
Der vorgestellte Algorithmus ist prototypisch in Software realisiert. Jedoch entwickelt die zugrundeliegende Daten- und Task Parallelisierung erst durch Umsetzung in Hardware das volle Beschleunigungspotential. Daher wird das Verfahren derzeit auf rekonfigurierbare Hardware (FPGA) portiert.
Bestätigt die Hardwareumsetzung die Ergebnisse, ermöglicht das System die Verwendung hochauflösender Farbbilddaten für die genaue Positionsbestimmung eines Fluggerätes im finalen Landeanflug. Der Einsatz in einem eigenständigem Navigations- (unbemanntes Flugsystem) oder Pilotenassistenzsystem kann einen wichtigen Beitrag zur Sicherheit in der kritischen Landeanflugphase leisten.

6. DANKSAGUNG

Die in dieser Arbeit vorgestellten Ergebnisse entstanden im Rahmen des Open Innovation Projektes in Zusammenarbeit mit Cassidian mit verschiedenen Forschungsinstituten und Universitäten. Ziel der von Cassidian finanzierten Kooperation ist die Erforschung und Weiterentwicklung technologischer Fähigkeiten für zukünftige unbemannte Flugsysteme.

7. REFERENZEN

[1] Lippmann, Blokzyl, Nagler und Hardt, „Portierung einer softwarebasierten Flugsteuerung auf FPGA-basierte Hardware," in *Tagungsband Dresdner Arbeitstagung Schaltungs- und Systementwurf (DASS2013)*, Dresden, 2013.

[2] Quinn und Graham, „FPGA-Based High Altitude Aircraft Systems," in *Military and Aerospace Programmable Logic Devices Conference (MAPLD2008)*, Annapolis; Maryland; USA, 2008.

[3] Prakash, Venkatasubramanyam, Krishnan und Nagendra, „Compact Field Programmable Gate Array (FPGA) Controller for Aircraft/Aerospace Structures," in *Proceedings of the International Conference on Aerospace Science and Technology*, Bangalore; India, 2008.

[4] Gribbon, Johnston und Bailey, „Implementing Image Processing Algorithms on FPGAs," in *Proceedings of the Eleventh Electronics New Zealand Conference, ENZCon'04*, New Zealand, Palmerston North, 2004.

[5] Gribbon, Johnston und Bailey, „A Real-time FPGA Implementation of a Lens Distortion Correction Algorithm with Bilinear Interpolation," in *Proceedings of the Image and Vision Computing New Zealand Conference 2003, Massey University*, Palmerston North, New Zealand, 2003.

[6] Saldanha, Hartmann und Bobda, „Scratch Detector – A FPGA Based System for Scratch Detection in Industrial Picture Development," Trento, 2010.

[7] Devi und Nagarajan, „FPGA Based High Performance Optical Flow Computation Using Parallel Architecture," in *International Journal of Soft Computing and Engineering (IJSCE)*, IJSCE, 2012, pp. 433-437.

[8] Papadonikolakis und Bouganis, „A Novel FPGA-based SVM Classifier," London, UK, 2010.

[9] Meng, Appiah, Hunter und Dickinson, „FPGA Implementation of Naive Bayes Classifier for Visual Object Recognition," London, 2011.

[10] M. Cho, Oberg und Kastner, „FPGA-Based Face Detection System Using Haar Classifiers," San Diego, United States, 2009.

[11] Chen und Guo, „The FPGA Implementation Of Kalman Filter," in *Proceedings of the 5th WSEAS International Conference on Signal Processing, Computational Geometry & Artificial Vision*, Malta, 2005.

[12] Bonato, Peron, Wolf, d. Holanda, Marques und Cardoso, „An FPGA Implementation for a Kalman Filter with Application to Mobile Robotics," in *International Symposium on Industrial Embedded Systems (SIES07)*, 2007.

[13] V. H. Blokzyl, „A Hardware-accelerated Real-time Image Processing Concept for High-resolution EO Sensors," in *Proceedings of the 61. Deutscher Luft- und Raumfahrtkongress (DLRK2012)*, Berlin, 2012.

[14] Zhu, Plataniotis und Venetsanopoulos, „Comprehensive analysis of edge detection in color image processing," *Optical Engineering,* pp. 612-625, 1999.

[15] Verzakov, Paclik und Duin, „Edge Detection in Hyperspectral Imaging: Multivariate Statistical Approaches," in *Proceedings of Structural and Syntactic Pattern Recognition (SSPR2006)*, Hong Kong, China, 2006.

[16] Moreno, Garcia, Puig und Julia, „Robust color edge detection through tensor voting," in *Proceeding of IEEE International Conference on Image Processing (ICIP2009)*, Cairo, Egypt, 2009.

[17] Xu, Ye und Luo, „Color edge detection using multiscale quaternion convolution," *International Journal of Information Sciences and Techniques,* pp. 354-358, Dezember 2010.

[18] Lacroix, „Color Line Detection," in *Proceedings of International Conference on Image Analysis and Processing (ICIAP 2011)*, Ravenna, Italy, 2011.

[19] Lindner, Blokzyl, Wanielik und Scheunert, „Applying Multi Level Processing for Robust Geometric Lane Feature Extraction," in *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, University of Utah, Salt Lake City, UT, USA, Sept. 5-7, 2010, 2010.

[20] Scheunert, Fuzzy-Mengen-Verknüpfung und Fuzzy-Arithmetik zur Sensor-Daten-Fusion, Duesseldorf: VDI Verlag GmbH, 2002.

(A) BILDBEISPIELE

Abb. 14: Ergebnisse der Offlineverarbeitung eines Landeanflugs auf Chemnitz-Jahnsdorf

[http://www.nrairport.com/images/phocagallery/AIRPORT/thumbs/phoca_thumb_l_rwy32%20final%20approach.jpg, Zugriff: 20.06.2013]

[http://de.m.wikipedia.org/wiki/Datei:Great_falls_final_approach.jpg, Zugriff: 20.06.2013]

[http://farm5.staticflickr.com/4035/4422567723_273c9fec41_o.jpg, Zugriff: 20.06.2013]

[http://160knots.com/images/fox/final.jpg, Zugriff: 20.06.2013]

Abb. 15: Detektionsergebnisse auf unterschiedlichen Testbildern