



Technische Universität Hamburg-Harburg

Bildverarbeitung für Waldbranderkennung

Diplomarbeit

von

Eric Stephen Nyassi Ngatcha

Ausgabetermin: 1. August 2011

Abgabetermin: 1. Februar 2012

1. Prüfer: **Prof. Dr. Ralf Möller**

2. Prüfer: **Prof. Dr. Rolf-Rainer Grigat**

Betreuer: **Dipl.-Ing. Karsten Martiny**

Technische Universität Hamburg-Harburg

Institut für Softwaresysteme

Schwarzenbergstraße 95

21073 Hamburg



Zusammenfassung

Die vorliegende Arbeit gibt eine Lösung auf die Frage, wie Bildverarbeitung für Waldbranderkennung On-Board eines UAVs durchgeführt werden kann.

Dazu wird zuerst ein Verfahren implementiert, das Farbmodelle verwendet, um in einem Bild Feuerregionen von anderen Regionen abzugrenzen.

Anschließend wird eine Methode angewendet, die Feuer innerhalb eines Videos detektiert. Diese Methode basiert auf der Analyse der Ergebnisse, die von einer Kantendetektion entstanden sind.

Erklärung

Hiermit erkläre ich, Eric Stephen Nyassi Ngatcha, dass die vorliegende Arbeit von mir selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel erstellt wurde.

Hamburg, den 01.02.2012

Vorwort

Hiermit möchte ich mich bei Herrn Prof. Dr. Ralf Möller herzlich bedanken, der mir ermöglicht hat, die vorliegende Diplomarbeit auszuarbeiten.

Ebenfalls bedanke ich mich bei meinem Betreuer Dipl.-Ing. Karsten Martiny für seine konstruktiven Vorschläge.

Eric Stephen Nyassi Ngatcha
Hamburg, den 01.02.2012

Inhaltsverzeichnis

Abbildungsverzeichnis	xi
1 Einführung	1
1.1 Motivation	2
1.2 Ziel der Diplomarbeit	2
1.3 Gliederung	2
2 Digitale Bildverarbeitung	5
2.1 Bilddarstellung	6
2.1.1 Darstellung des Bildes als kontinuierliche Funktion	6
2.1.2 Darstellung des Bildes als diskrete Funktion	6
2.2 Farbmodelle	6
2.2.1 RGB-Farbmodell	7
2.2.2 Lab-Farbmodell	7
2.3 Histogramm	8
2.4 Segmentierung	9
2.5 Kanten	10
2.6 Verfahren zur Kantendetektion	12
2.7 Ausstattung des UAV zur On-Board-Bildverarbeitung	13
2.7.1 On-Board-Rechner	13
2.8 Aufnahme von Bildern oder Videos	14
3 Verfahren zur Feuererkennung	17
3.1 Gassensor	17
3.2 Hitzedetektion	18
3.3 Bildbasierte Feuerdetektion	18
3.3.1 Konvertierung von RGB zu Lab	20
3.3.2 Binarisierung	22
3.3.3 Segmentierung	24
3.3.4 Wiederholung	24
4 Videobasierte Feuererkennung	29
4.1 Überblick der videobasierten Feuererkennung	29
4.2 Detektion basiert auf Bewegung und Feuerfarben	30
4.2.1 Detektion von sich bewegenden Pixeln	31
4.2.2 Feuerpixel Detektion	31
4.3 Detektion basierend auf Flammenkanten	34

4.3.1	Selektion von Feuer-Framekandidaten	35
4.3.2	Selektion von Flammenregionen	35
4.3.3	Kantendetektion	36
4.3.4	Entscheidung	39
4.4	Simulationsversuch	42
4.4.1	Simulation durch Flammenkanten	43
5	Ergebnisse und Auswertung	47
5.1	Ergebnisse der bildbasierten Feuerdetektion	47
5.1.1	Ergebnisse von simulierten und verrauschten Bildern	50
5.1.2	Fazit	53
5.2	Ergebnisse der videobasierten Feuererkennung	53
5.2.1	Test von Videos mit sicheren Feuerinhalten	55
5.2.2	Test von Videos ohne Feuerinhalte	60
5.2.3	Gesamte Auswertung	62
5.2.4	Fazit	63
6	Zusammenfassung und Ausblick	65
6.1	Zusammenfassung	65
6.2	Ausblick	66
	Literaturverzeichnis	69

Abbildungsverzeichnis

1.1	UAV mit eingebauter Kamera im Polizeieinsatz [Hub11]	1
2.1	Farbmodell in RGB	7
2.2	Farbmodell in Lab	7
2.3	Histogramm von einem Farbbild in der ersten Reihe und Histogramm von einem Graubild in der letzten Reihe	8
	(a) Farbbild [Bae09]	8
	(b) Histogramm	8
	(c) Graubild	8
	(d) Histogramm	8
2.4	Segmentierung mit $S = 130$	9
	(a) Lena	9
	(b) Binärbild	9
2.5	Links oben: ideale Stufenkante; Rechts oben: ideale Rampenkante; Links unten: ideale Dachkante; Rechts unten: ideale Treppenkante [Ste93]	11
2.6	Kantendetektion mit Hilfe des Canny-Verfahrens	12
	(a) Lena	12
	(b) Kanten	12
2.7	UAV mit eingebauter Kamera und eingebautem Rechner	13
2.8	UAV interne Bildverarbeitung	14
3.1	Schematische Darstellung des Verfahrens, das die Detektion von dem Feuer in einem Bild ermöglicht. Dieser Vorgang wird zweimal durchgeführt.	19
3.2	Transformation von RGB-Bild in Lab-Bild und L-,a-,b-Kanalbild	21
	(a) Waldbrand [Kid11]	21
	(b) Waldbrand(Lab)	21
	(c) L-Kanal	21
	(d) a-Kanal	21
	(e) b-Kanal	21

3.3	Binarisierung von Bildern	23
	(a) Eingangsbild	23
	(b) Binärbild L-Kanal	23
	(c) Binärbild a-Kanal	23
	(d) Binärbild b-Kanal	23
	(e) Binärbild nach Gleichung 3.6	23
	(f) Kombiniertes Bild	23
3.4	Segmentierung	24
	(a) Eingangsbild	24
	(b) Kombiniertes Bild	24
	(c) Segmentiertes Feuer	24
3.5	Segmentiertes Feuer in Lab-Farbmodell	25
	(a) Eingangsbild	25
	(b) L-Kanal	25
	(c) a-Kanal	25
	(d) b-Kanal	25
3.6	Binarisierung von Bildern des segmentierten Feuers	26
	(a) Eingangsbild	26
	(b) Binärbild L-Kanal	26
	(c) Binärbild a-Kanal	26
	(d) Binärbild b-Kanal	26
	(e) Binärbild nach Gleichung 3.6	26
	(f) Kombiniertes Bild	26
3.7	Detektiertes Feuer	27
	(a) Eingangsbild	27
	(b) Kombiniertes Bild	27
	(c) Feuer	27
4.1	Detektion von bewegten Pixeln	30
	(a) Schematische Darstellung zur Detektion von Feuerpixeln	30
4.2	Detektion von sich bewegenden Pixeln	31
	(a) Originalbild 1	31
	(b) Bewegte Pixel	31
	(c) Bewegte Pixel 2	31
4.3	Detektion von Feuerpixeln	33

(a)	Originalbild 1	33
(b)	Feuerpixel1	33
(c)	Feuerpixel2	33
(d)	Feuerpixel3	33
4.4	Schritte der Detektion basierend auf Flammenkanten	34
4.5	Selektion von Feuer-Framekandidaten	35
(a)	ForestvideoFrame1	35
(b)	Framekandidat	35
4.6	Selektion von Flammenregionen	36
(a)	ForestvideoFrame1	36
(b)	Flammenregion	36
4.7	Entscheidungsbaum zur Anzeige von Alarmbilder	42
4.8	Simulierte Feuerdetektion	43
(a)	Schematische Darstellung der Simulation einer Feuerdetektion	43
5.1	Testbilder	48
(a)	Bild mit eindeutigem Feuer [CtBC05]	48
(b)	Bild mit Feuer und nicht Feuer-Elementen [WW09]	48
(c)	Bild ohne Feuer [Jim12]	48
5.2	Testbild mit Feuer	48
(a)	Bild mit eindeutigem Feuer	48
(b)	Binäres Bild	48
(c)	Ergebnis	48
5.3	Testbild mit Feuer und nicht Feuer-Elementen	49
(a)	Bild mit Feuer und nicht Feuer-Elementen	49
(b)	Binäres Bild	49
(c)	Ergebnis	49
5.4	Testbild mit nicht Feuer-Elementen	49
(a)	Bild mit nicht Feuer-Elementen	49
(b)	Binäres Bild	49
(c)	Ergebnis	49
5.5	Simulierte Testbilder in der ersten Reihe und verrauschte Bilder in der zweiten Reihe	50
(a)	Simuliertes Feuer 1 [Com09]	50
(b)	Simuliertes Feuer 2 [P.P09]	50

(c)	Simuliertes Feuer 3 [Chr10]	50
(d)	Normales Bild	50
(e)	Verrauschtes Bild mit gauss'schem Rauschen	50
(f)	Verrauschtes Bild mit gleichverteiltem Rauschen	50
5.6	Simulierte Testbilder	51
(a)	Simuliertes Bild 1	51
(b)	Binäres Bild	51
(c)	Ergebnis	51
(d)	Simuliertes Bild 2	51
(e)	Binäres Bild	51
(f)	Ergebnis	51
(g)	Simuliertes Bild 3	51
(h)	Binäres Bild	51
(i)	Ergebnis	51
5.7	Testbild mit Rauschen	52
(a)	Verrauschtes Bild mit gauss'schem Rauschen	52
(b)	Binäres Bild	52
(c)	Ergebnis	52
(d)	Verrauschtes Bild mit gleichverteiltem Rauschen	52
(e)	Binäres Bild	52
(f)	Ergebnis	52
5.8	Testvideos mit Feuer und Feuerähnlichen Elementen	54
(a)	Video 1	54
(b)	Video 2	54
(c)	Video 3	54
(d)	Video 4	54
(e)	Video 5	54
(f)	Video 6	54
(g)	Video 7	54
(h)	Video 8	54
(i)	Video 9 [pix12]	54
(j)	Video 10 [Twi11]	54
(k)	Video 11 [tik11]	54
5.9	Selektion von Feuer-Framekandidaten	55
(a)	Video 3	55

(b)	Video 3 Framekandidaten	55
(c)	Video 8	55
(d)	Video 8 Framekandidaten	55
5.10	Selektion von Flammenregionen	56
(a)	Video 3 Framekandidaten	56
(b)	Video 3 Flammenregion	56
(c)	Video 8 Framekandidaten	56
(d)	Video 8 Flammenregion	56
5.11	Kantendetektion	57
(a)	Video 3 Framekandidaten	57
(b)	Video 3 Grauwertbild	57
(c)	Video 3 Binärbild	57
(d)	Video 3 Dilatation	57
(e)	Video 3 Erosion	57
(f)	Video 3 Kantenbild	57
(g)	Video 8 Framekandidaten	57
(h)	Video 8 Grauwertbild	57
(i)	Video 8 Binärbild	57
(j)	Video 8 Dilatation	57
(k)	Video 8 Erosion	57
(l)	Video 8 Kantenbild	57
5.12	Detektion, nachdem die erste Entscheidungsbedingung erfüllt wurde. . .	58
(a)	Video 8	58
(b)	Erstes Entscheidungsalarmbild für Video 8	58
(c)	Video 3	58
(d)	Erstes Entscheidungsalarmbild für Video 3	58
5.13	Detektion, nachdem die zweite Entscheidungsbedingung erfüllt wurde. . .	59
(a)	Video 8	59
(b)	Zweites Entscheidungsalarmbild für Video 8	59
(c)	Video 3	59
(d)	Zweites Entscheidungsalarmbild für Video 3	59
5.14	Detektion, nachdem die dritte Entscheidungsbedingung erfüllt wurde. . .	59
(a)	Video 8	59
(b)	Drittes Entscheidungsalarmbild für Video 8	59
5.15	Selektion von Feuer-Framekandidaten in Video ohne Feuer	60

(a)	Video 9	60
(b)	Video 9 Framekandidaten	60
(c)	Video 10	60
(d)	Video 10 Framekandidaten	60
5.16	Selektion von Flammenregionen in Videos ohne Feuer	61
(a)	Video 9 Framekandidaten	61
(b)	Video 9 Flammenregion	61
(c)	Video 10 Framekandidaten	61
(d)	Video 10 Flammenregion	61
5.17	Kantendetektion	61
(a)	Video 9 Kantenbild	61
(b)	Video 10 Kantenbild	61

Kapitel 1

Einführung

Mit Unmanned Aerial Vehicles (UAV) werden wiederverwendbare Fluggeräte beschrieben. Es wird zwischen ferngesteuerten und autonomen Fluggeräten unterschieden. Hier bedeutet autonom, dass das UAV einer festen vorprogrammierten Route folgt und diese dann selbst einhält [Hei06]. Raketen können nicht als UAV bezeichnet werden. Raketen können autonom ein vorprogrammiertes Ziel verfolgen, aber sie sind nicht wiederverwendbar.



Bild 1.1: UAV mit eingebauter Kamera im Polizeieinsatz [Hub11]

UAVs werden unter anderem zur Aufklärung, Überwachung und Analyse von Geschehen eingesetzt. Die technische Ausstattung von UAVs hängt meistens von dem Einsatzgebiet ab. Sie können mit Kameras, Kommunikationsgeräten und anderen Hardwares ausge-

stattet werden. Das Fluggerät, die Kommunikationsinfrastruktur und die Base-Station bilden das sogenannte UAV-System. Heutzutage werden UAVs in vielen Bereichen eingesetzt. Beispiele hierfür stellen Kriegsgebiete, Katastrophengebiete und die Kartographie dar.

Ein UAV kann in manchen Fällen als das erweiterte Auge von einem Benutzer, der in der Base-Station das UAV überwacht, betrachtet werden. UAVs werden in Fällen eingesetzt, in denen der Einsatz von Menschen eine große Gefahr darstellen kann. In dem Fall vom **Waldbrand** können UAVs eingesetzt werden, um die Verbreitung des Feuers abzuschätzen.

1.1 Motivation

Die UAVs können mit einer oder mehreren Kameras ausgestattet werden und es kann erwartet werden, dass diese autonom fliegen. Dabei sollten sie grundsätzlich die aufgenommenen Bilder oder bestimmte Informationen an die Base-Station weiterleiten, damit diese Bilder oder allgemeine Informationen ausgewertet werden. Diese Vorgehensweise ist eine der Grundaufgabe von UAVs. Mit dem heutigen Stand der Technologie ist diese Grundleistung von UAVs dennoch nicht effizient genug oder sehr begrenzt.

Die Autonomie von UAVs sollte nicht nur darin bestehen, vorprogrammierte Wege zu durchlaufen und zurück zu der Basis-Station zu fliegen, sondern auch weitere Aufgaben selbst auszuführen. Die vorliegende Arbeit soll sich deshalb mit der Bewertung von Videos und Bildern durch UAVs beschäftigen.

1.2 Ziel der Diplomarbeit

Das Ziel dieser Arbeit ist es, einen Weg zu finden, damit digitale Bildverarbeitung On-Board UAV durchgeführt werden kann. Es sollte für ein UAV möglich sein, Feuer zu erkennen, ohne dass ein Benutzer oder Mitarbeiter in der Base-Station eingreift, um den Entscheidungsprozess des UAVs zu beeinflussen.

1.3 Gliederung

Diese Arbeit ist wie folgt gegliedert:

1. Im Kapitel 2 werden Grundlagen zur digitalen Bildverarbeitung vorgestellt. Schwerpunktmäßig geht es um die zum Verständnis der weiteren Ausführung notwendigen Grundlagen.

-
2. In Kapitel 3 geht es um die Verfahren oder Methoden, die zur Feuererkennung eingesetzt werden. Die Anwendung von Gassensoren und Hitzedetektoren wird hier angesprochen. Außerdem wird ein Verfahren für die Erkennung des Feuers in Bildern erläutert.
 3. Videobasierte Feuerdetektionsverfahren werden in dem Kapitel 4 behandelt; hier wird auch über drei Entscheidungsprozesse zur Feuerdetektion diskutiert. In diesem Kapitel wird auch ein Simulationsversuch erläutert, in dem die erzielten Ergebnisse sofort angewandt werden, um Feuer in einer Simulationsumgebung erkennen zu können.
 4. Kapitel 5 enthält eine Auswertung und die Ergebnisse der Verfahren, die hier zur Feuerdetektion angewendet wurde.
 5. In Kapitel 6 wird die Arbeit zusammengefasst und ein Ausblick gegeben.

Kapitel 2

Digitale Bildverarbeitung

Im vergangenen Jahrhundert hat die Technologie eine rasante Entwicklung genommen. Diese Entwicklung wurde natürlich durch eine sehr gute Kommunikation unter den Forschern ermöglicht, wobei die visuelle Information in der Form von Grafiken und Abbildungen eine große Rolle gespielt hat.

Die Fortschritte in der Computertechnologie haben die Herstellung von Software und Hardware für die Verarbeitung von Bildern und Videos ermöglicht.

In [Abm94] werden zwei primäre Ziele der Bildverarbeitung benannt: Ein Ziel ist es, aus Bildern wieder Bilder zu erstellen, um aus modifizierten Bildern mehr Informationen für die visuelle Betrachtung und Auswertung zu bekommen, während das andere Ziel ist, Bilder automatisch zu erkennen. Diese beiden Hauptziele der Bildverarbeitung umfassen auch die **Analyse von Bildern**, die **Szenenanalyse**, die **Generierung von Bildern** und **Übertragung von Bildern**.

Die digitale Bildverarbeitung repräsentiert die unterschiedlichen Verfahren, die angewendet werden, um aus Bildern und Videos in numerischer Form bestimmte Informationen zu bekommen. Zu diesen Verfahren zählt man **Mustererkennung**, **Bilderkennung** und **optische Vermessung**.

Digitale Bildverarbeitung wird in vielen wissenschaftlichen Bereichen eingesetzt:

- Autonome Roboter
- Seismologie
- Astronomie
- Ozeanographie
- Luftaufklärung
- Medizin

In den folgenden Kapiteln werden gezielt einige wichtige Punkte der Bildverarbeitung erläutert. Das Ziel ist es, dem Leser Grundkenntnisse zu vermitteln. Es werden die Methoden und Definitionen angesprochen, die für diese Arbeit eine wichtige Rolle gespielt haben.

2.1 Bilddarstellung

In [Bov05] wird ein Bild als Repräsentation einer Abbildung definiert, die mit der Zeit unverändert bleibt. Nach der Definition von Wolfgang Abmayer [Abm94] ist ein Bild ein flaches Objekt, dessen Helligkeit oder Farbe sich von Punkt zu Punkt verändert.

2.1.1 Darstellung des Bildes als kontinuierliche Funktion

In der mathematischen Darstellung kann man ein Bild als eine Funktion $f(x,y)$ darstellen, wobei x die Spalten und y die Zeilen sind. Die Farbe eines Bildpunktes lässt sich als Kombination der Funktionen $r(x,y)$ "Rotanteil", $g(x,y)$ "Grünanteil", $b(x,y)$ "Blauanteil" betrachten. Der Wert von $f(x,y)$ definiert die Helligkeit oder den Grauwert eines Punktes (x,y) . Der Helligkeitswert der Funktion wird in dem Intervall $[0 - 255]$ angegeben.

$$0 \leq f(x,y) \leq 255 \quad (2.1)$$

2.1.2 Darstellung des Bildes als diskrete Funktion

Damit ein Bild im Rechner verarbeitet werden kann, muss es digitalisiert werden. Dafür müssen die Bildwerte und seine Ortsvariablen zunächst diskretisiert werden. Das Bild wird als eine Matrix mit X Spalten und Y Zeilen dargestellt. Die Bildmatrix besteht aus "Pixeln" (diskreten Bildpunkten) $F(x,y)$.

$$\begin{cases} 1 \leq x \leq X \\ 1 \leq y \leq Y \end{cases} \quad (2.2)$$

2.2 Farbmodelle

Ein Farbbild ist ein Bild, das durch additive oder subtraktive Farbmischungen aus den Bildfunktionen von drei Spektralbereichen aufgebaut ist. Das Farbbild kann ein Durchsichts- oder Aufsichtsbild sein, wozu auch das Projektionsbild gehört [PRB12].

2.2.1 RGB-Farbmodell

Im RGB (**R**ot-**G**rün-**B**lau) Modell werden die Farben über die additive Farbmischung erstellt.

- *Weiß* ist definiert durch $R = G = B = 1$
- *Schwarz* ist definiert durch $R = G = B = 0$
- *Grau* ist das Ergebnis von $R = G = B$ und $0 < R, G, B < 1$

Eine Kombination aus Rot und Grün ergibt "GELB", Rot und Blau ergibt "MAGENTA", Grün und Blau ergibt "CYAN".

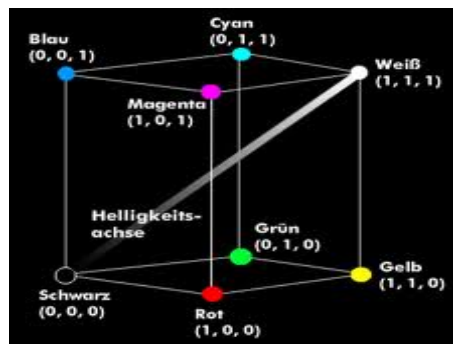


Bild 2.1: Farbmodell in RGB

Das Bild 2.1 [KL11a] zeigt einen Einheitswürfel, in dem primäre und sekundäre Farben dargestellt sind.

2.2.2 Lab-Farbmodell

Ein Lab-Bild oder CIE $L^*a^*b^*$ -Bild besteht aus drei Kanälen; ein Helligkeitskanal L (L^*) "Luminanz" und zwei Farbkanäle: a (a^*) von Grün bis Rot, b (b^*) von Blau bis Gelb.

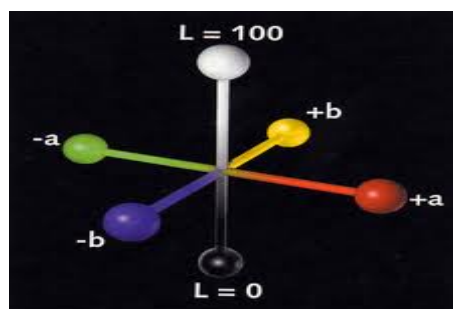


Bild 2.2: Farbmodell in Lab

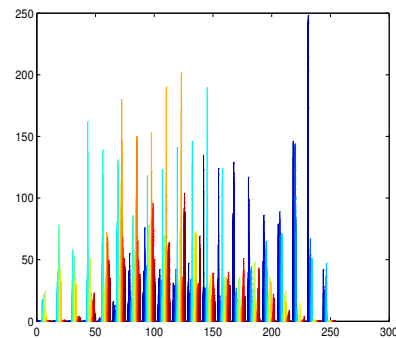
Die Repräsentation von den Kanälen ist in dem Bild(2.2) [Rei08] dargestellt. Lab trennt die Helligkeits- und die Farbinformation. Im L-Kanal werden die Tonwertunterschiede und die Zeichnung abgebildet. a und b halten die Farbunterschiede fest [Hec02].

2.3 Histogramm

Die Kerngrößen von Bildern oder Bildobjekten (Regionen) sind Parameter oder Parametertupel, die zur Charakterisierung und Klassifizierung von globalen oder lokalen Eigenschaften dienen. Unter den globalen Kerngrößen gibt es: **das Histogramm, den mittleren Grauwert und den globalen Kontrast**. Das Histogramm ist die wichtigste globale Kerngröße und definiert die statistische Häufigkeit der Grauwerte bzw. Farbwerte in einem Bild [LOPR97].



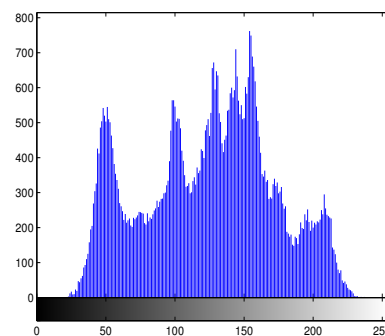
(a) Farbbild [Bae09]



(b) Histogramm



(c) Graubild



(d) Histogramm

Bild 2.3: *Histogramm von einem Farbbild in der ersten Reihe und Histogramm von einem Graubild in der letzten Reihe*

In den Bilder 2.3(b) und 2.3(d) kann man das Histogramm von Lena (Beispielbild in der Bildverarbeitung) als Farb- bzw. Graubild sehen.

2.4 Segmentierung

In der digitalen Bildverarbeitung wird die Segmentierung dazu verwendet, um bestimmte Informationen oder Objekte eines Bildes zu extrahieren. Die extrahierten Objekte werden für die weitere Analyse benötigt. Nach einer Bildsegmentierung kann man ein **Binärbild** erhalten. Ein Binärbild besteht aus zwei Farben: *SCHWARZ* und *WEISS*. Der Pixelwert von Schwarz ist **0** und der Pixelwert von Weiß ist **1**. Seien ein Graubild $F(x,y)$ und ein Schwellwert S mit $0 \leq S \leq 255$ gegeben. Ein Binärbild $B(x,y)$ lässt sich mit dem Gleichungssystem (2.3) gewinnen.

$$B_S(x,y) = \begin{cases} 1 & \text{für } F(x,y) \geq S \\ 0 & \text{für } F(x,y) < S \end{cases} \quad (2.3)$$



(a) Lena



(b) Binärbild

Bild 2.4: Segmentierung mit $S = 130$

In der Abbildung 2.4 wird das Graubild 2.4(a) in ein Binärbild 2.4(b) umgewandelt.

2.5 Kanten

Kanten repräsentieren die Grenze von physikalischen Objekten oder die Änderungen an deren Oberfläche.

Im Bereich der Bildverarbeitung sind Kanten die Diskontinuität der Helligkeit eines Bildes. Man muss auch anmerken, dass unterschiedliche Kanten auftreten können. Durch die Intensitätstärke der Helligkeit kann man unterschiedliche Kantenmodelle definieren. Nach [Ste93] kann zwischen folgende Kantenarten unterschieden werden:

- Die ideale Stufenkante
 - Die ideale Rampenkante
 - Die ideale Dachkante
 - Die ideale Treppenkante
 - Die reale Kante
- ideale Stufenkante : In einem Definitionsbereich, in dem der Grauwert wechselt, haben alle x abhängig von negativem oder positivem Bereich einen gemeinsamen Wert.

$$S(x) = \begin{cases} a & \text{für } x < 0 \\ b & \text{sonst} \end{cases} \quad (2.4)$$

- ideale Rampenkante : Hier wechselt der Grauwert langsam in Abhängigkeit von x . Man muss in diesem Fall feststellen, dass es kompliziert ist, den Definitionsbereich von x zu definieren. w ist hier die Breite der Rampe.

$$R(x) = \begin{cases} a & \text{für } x < -\frac{w}{2} \\ a + (b - a)(x + \frac{w}{2}) & \text{für } -\frac{w}{2} \leq x \leq \frac{w}{2} \\ b & \text{für } x > \frac{w}{2} \end{cases} \quad (2.5)$$

- ideale Dachkante : Die Dachkante ist eine andere Form der Rampenkante.

$$D(x) = \begin{cases} a & \text{für } x < -\frac{w}{2} \\ a + 2(b - a)(x + \frac{w}{2}) & \text{für } -\frac{w}{2} \leq x \leq 0 \\ a + 2(b - a)(\frac{w}{2} - x) & \text{für } 0 \leq x \leq \frac{w}{2} \\ a & \text{für } x > \frac{w}{2} \end{cases} \quad (2.6)$$

- ideale Treppenkante : Diese Kantenart besteht aus mehreren Stufenkanten.

$$T(x) = \begin{cases} a & \text{für } x < -\frac{w}{2} \\ b & \text{für } -\frac{w}{2} \leq x \leq \frac{w}{2} \\ c & \text{für } x > \frac{w}{2} \end{cases} \quad (2.7)$$

- reale Kantenform : Die reale Kantenform ist eine der idealen Kanten, aber in dieser Form gibt es auch Rauschen, die vorhanden sind. In diesem Fall benutzt man die erste und zweite Ableitung, um den Unterschied von Grauwerten zu finden.

Die Ergebnisse der Kantendetektion werden in Mustererkennung, optischer Vermessung und Segmentierung verwendet. Bild 2.5 zeigt die unterschiedlichen Kanten.

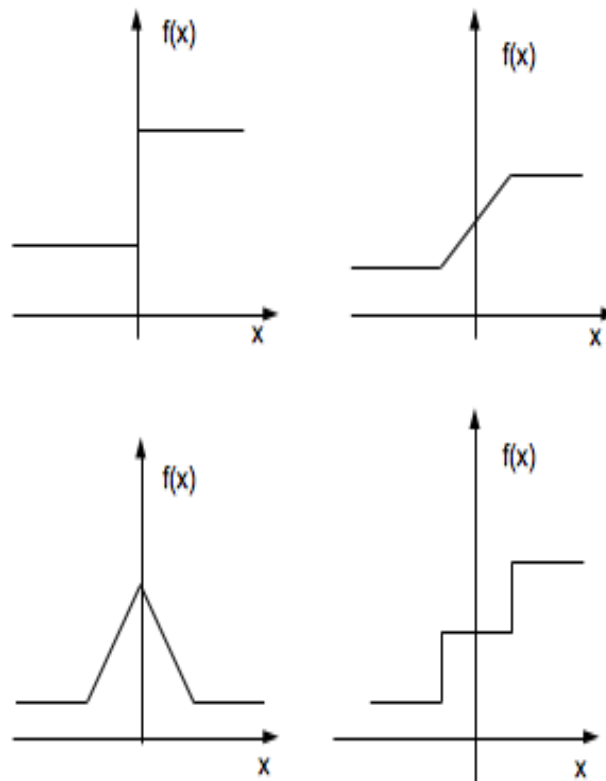
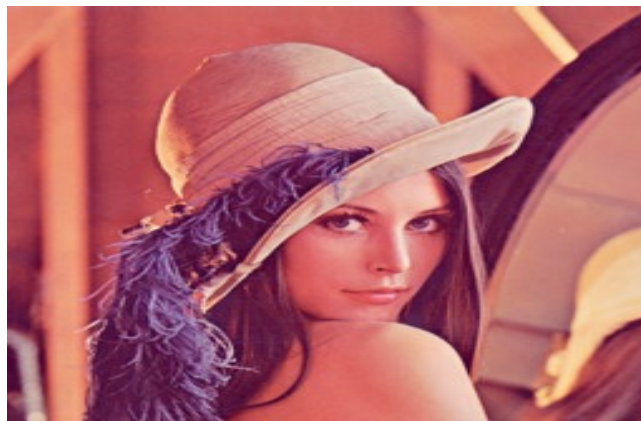


Bild 2.5: Links oben: ideale Stufenkante; Rechts oben: ideale Rampenkante; Links unten: ideale Dachkante; Rechts unten: ideale Treppenkante [Ste93]

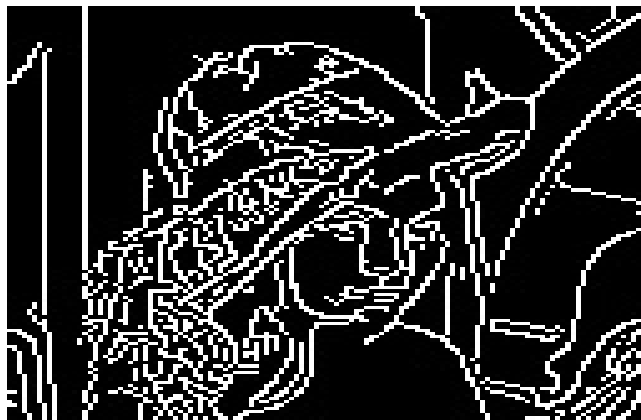
2.6 Verfahren zur Kantendetektion

Die Kantendetektion kann mit Hilfe von vielen Operatoren berechnet werden. Die gängigsten Operatoren sind:

- Robert
- Sobel
- Canny
- Diriche
- Marr and Hildreth
- stochastische Berechnung von Markov
- Diskrete Hilbert-Transformation



(a) Lena



(b) Kanten

Bild 2.6: *Kantendetektion mit Hilfe des Canny-Verfahrens*

Die meisten dieser Verfahren basieren auf Berechnungen von Ableitungen in erster und zweiter Ordnung. Bei der ersten Ableitung kann man das lokale Maximum einer Kante erkennen und bei der zweiten Ableitung gibt es einen Nulldurchgang. In der Bildverarbeitung werden für zweidimensionale Bilder zuerst die partiellen Ableitungen in X und Y Richtungen berechnet.

Das Problem in diesem Verfahren entsteht, wenn die Bilder immer komplexer werden, denn dadurch erhöht sich die Rechenzeit.

2.7 Ausstattung des UAV zur On-Board-Bildverarbeitung



Bild 2.7: UAV mit eingebauter Kamera und eingebautem Rechner

In diesem Abschnitt werde ich ein paar Grundideen ansprechen, auf deren Basis die digitale Bildverarbeitung von einem UAV autonom durchgeführt werden kann.

Das Bild (2.7) [Kos04] zeigt eine Ausstattung des UAVs, die man einbauen kann, damit die Aufgabe der Bildverarbeitung On-Board UAVs in Echtzeit geführt werden. Die Plattform des UAVs für die Bildverarbeitung kann aus einer **Kamera** und einem **On-Board-Rechner** bestehen.

2.7.1 On-Board-Rechner

Damit die Verarbeitung in Echtzeit gemacht wird, ist es wichtig, einen leistungsfähigen Prozessor einzubauen. Die Prozessoren, die häufig in UAVs verwendet werden, sind FPGA und DSP.

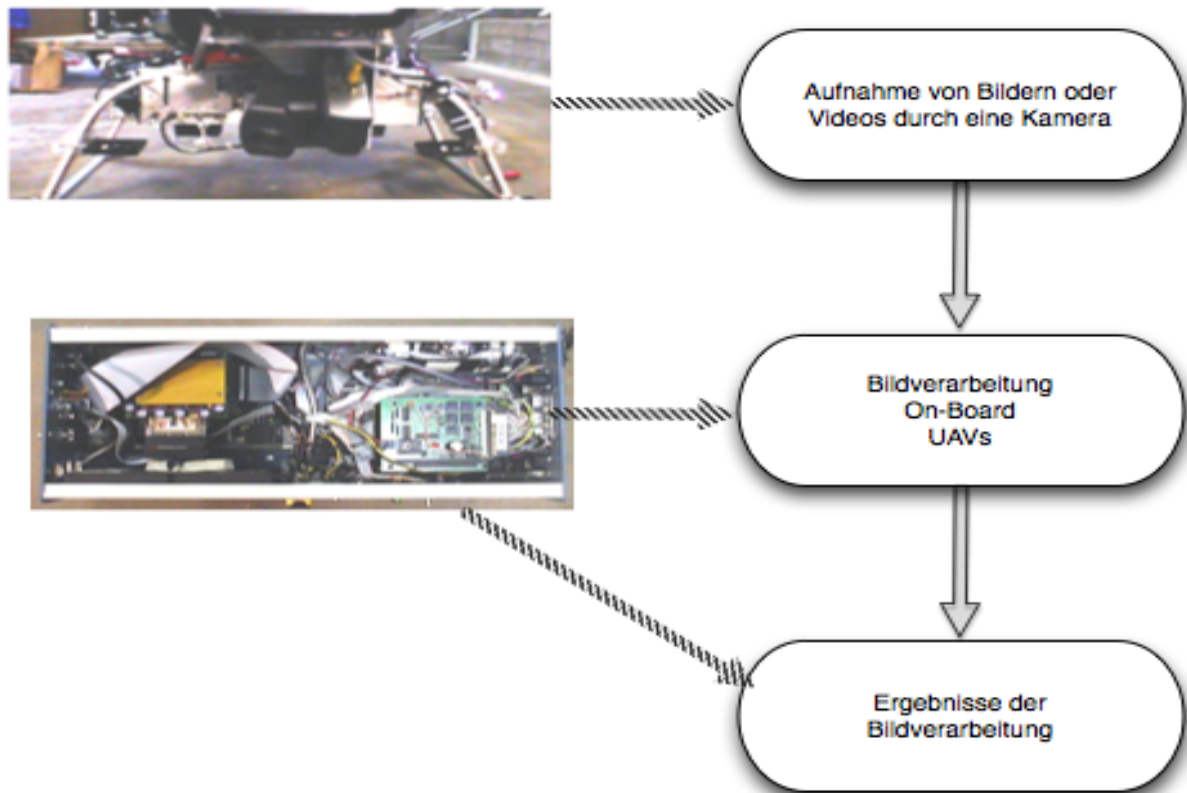


Bild 2.8: UAV interne Bildverarbeitung

Ein *FPGA* (*Field Programmable Gate Array*) besteht aus Tausenden von Logikzellen, die miteinander verschaltet werden. Seine Vorteile liegen in der Ausführungsgeschwindigkeit und der Möglichkeit Berechnungsvorgänge zu parallelisieren [Str06].

DSP (*Digital Signal Prozessor*) sind Prozessoren, die auf Signalverarbeitung spezialisiert sind. DSP sind echtzeitfähig und werden in der Regel in C oder C++ programmiert [Str06]. Es ist in vielen Fällen für den Entwickler besser und effizient DSP zu verwenden, weil er die Algorithmen für die digitale Bildverarbeitung auf dem PC programmieren und anschließend auf einem DSP einbauen kann.

2.8 Aufnahme von Bildern oder Videos

Mit der Hilfe von openCV können bestimmte Funktionen und Strukturen verwendet werden, um Bilder oder Videos aufzunehmen.

Mit `typedef struct CvCapture` kann ein Video durch eine AVI-Datei oder eine Kamera aufgenommen werden.

`typedef struct IplImage` ist die Struktur, in der das Bild gespeichert wird.

`CvCapture* cvCaptureFromCAM(int index)` ist die Funktion, die den Stream

einer Kamera startet. Das erste Parameter bestimmt die Nummer der Kamera, die verwendet wird. Sie gibt einen Pointer auf `CvCapture` zurück.

Mit `IplImage* cvQueryFrame(CvCapture* capture)` kann ein Frame aus einem Stream gelesen werden [the09].

Kapitel 3

Verfahren zur Feuererkennung

In der Bundesrepublik Deutschland ereignen sich alljährlich ungefähr eine halbe Millionen Schadenfeuer mit erwähnenswerten Folgen, die sich insgesamt auf eine Summe von 6 bis 7 Mrd. DM pro Jahr belaufen. 400 bis 600 Brandtote sind zu beklagen. Die Schadenfeuertendenz ist steigend.[...] Jede wirksame Brandbekämpfung setzt eine frühzeitige Erkennung eines entstehenden Feuers voraus.¹

Dieses Kapitel beschäftigt sich mit den Verfahren zur Feuerdetektion. Während erwünschtes Feuer für uns Menschen sehr nützlich sein kann, kann unerwünschtes Feuer eine enorme Gefahr für unser Ökosystem darstellen. Ein Feuer in einem Tunnel kann Menschenleben kosten. Der Waldbrand zerstört nicht nur den Wald, sondern kann durch seine unvorhersehbare Richtung, die er einschlägt, auch für die Menschen sehr bedrohlich werden. Ebenso können erhebliche wirtschaftliche Schäden entstehen.

Die Entwicklung von Feuerfrüherkennungssystemen spielt eine sehr wichtige Rolle in der Bekämpfung von Feuer. Heutzutage beschäftigen sich viele Wissenschaftler mit unterschiedlichen Aspekten der Feuerbekämpfung. In der Regel geht es darum, eine effiziente Methode zu entwickeln, die es ermöglicht, Feuer schneller und möglichst genau zu erkennen.

3.1 Gassensor

Während eines Brandes werden Gase in der Natur freigesetzt. Es gibt dabei typische Gasarten, die eine Feuerdetektion ermöglichen. Es handelt sich um Kohlenmonoxid CO , Kohlenstoffdioxid CO_2 , Wasserstoff H_2 und Sauerstoff O_2 . Anhand von diesen Gasen wurden Verfahren entwickelt, die das Ziel haben, Feuer an einem Ort zu detektieren. Durch die Berechnung der Konzentration von diesen Gasen und der Rauchdichte lässt

¹Forum Forschung 98. www.uni-due.de/~yzf010/publikationen/forum/forum98/08etech/0802/fram.htm (31-01-2012)

sich nicht nur feststellen, ob Feuer an einem Ort vorhanden ist, durch auch wird der Typ des Feuers bestimmt. Bekannte Feuertypen sind der **Holzbrand**, der **Schwelbrand**¹, der **Baumwollbrand**, der **Plastikbrand** und der **Flüssigkeitsbrand** [LK03].

3.2 Hitzedetektion

Hier wird ein linienförmiger Wärmemelder verwendet. Es besteht aus einem als Datenbus wirkenden Flachbandkabel, in das in regelmäßigen Abständen kleine Wärmesensoren integriert sind [Sec12]. Im Abstand von 7,2 Metern kann auf diese Weise innerhalb des Kabels eine exakte Temperaturmessung erfolgen. Ein Alarm wird nach einer Analyse des Temperaturverhaltens ausgelöst. Dies erfolgt meistens, wenn sich die Temperaturen um 40 °C und 90 °C befinden [Mea94].

3.3 Bildbasierte Feuerdetektion

Ein UAV mit eingebauter Kamera kann Bilder von einer Landschaft (z.B. einem Wald) aufnehmen und diese im Anschluss sofort weiterverarbeiten, um bestimmte Informationen (z.B. über einen Waldbrand) zu erhalten.

In diesem Abschnitt wird eine Methode verwendet, um Feuer in einem Bild zu detektieren. Diese Methode basiert auf Farbmodellen. Das angewendete Verfahren: **Fast and Efficient Method for Fire Detection Using Image Processing** [Cel10] wurde von Turgay Celik vorgeschlagen. In diesem Verfahren gehen wir davon aus, dass die Kamerabilder das RGB-Farbmodell haben. Die Bilder werden in das Lab-Farbmodell konvertiert und danach werden farbbasierte Verfahren verwendet, um die Feuerpixel zu detektieren.

Eine Darstellung des Verfahrensablaufs ist in der Abbildung (3.1) vorhanden. Es ist bekannt, dass Feuer eine starke Luminanz besitzt und die Farbe des Feuers befindet sich zwischen Rot und Gelb, aber generell geht die Tendenz zu Rot. Durch diese Eigenschaft des Feuers empfiehlt sich das Lab-Farbmodell als sehr geeignet; siehe Bild (2.2).

¹Ein Schwelbrand unterscheidet sich vom klassischen Brand bzw. vom sogenannten Feuerbrand dahingehend, dass keine sichtbaren Flammen existieren. Brände dieser Art entstehen, wenn keine ausreichende Zufuhr an Sauerstoff gewährleistet ist und sich somit keine Flammen bilden. Stattdessen lodert bzw. glüht der Brand vor sich hin, wobei er sich jedoch immer noch ausbreiten kann. [Ans12]

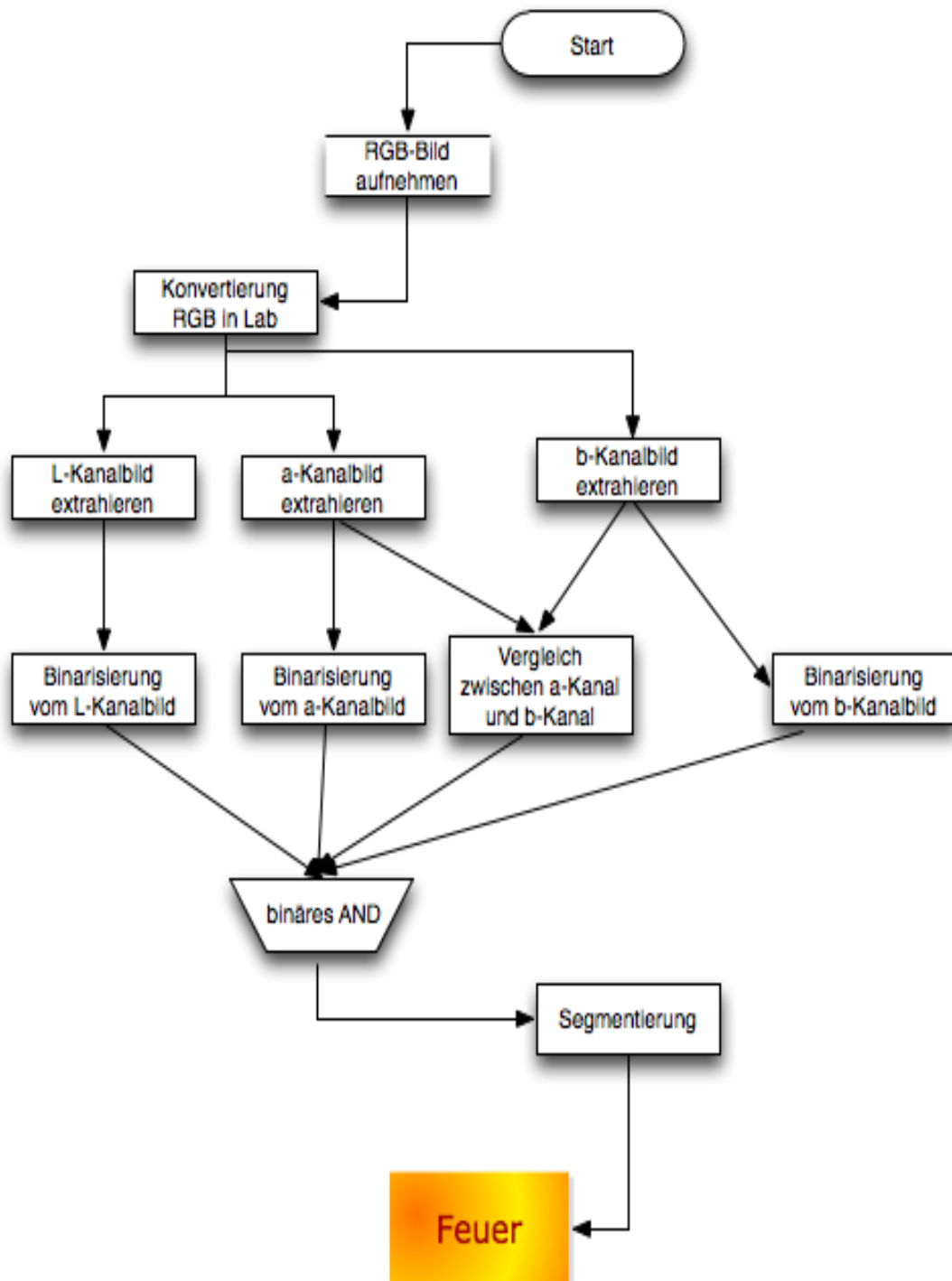


Bild 3.1: Schematische Darstellung des Verfahrens, das die Detektion von dem Feuer in einem Bild ermöglicht. Dieser Vorgang wird zweimal durchgeführt.

3.3.1 Konvertierung von RGB zu Lab

Die folgenden Gleichungen wurden in [Iro12], [Cel10] erläutert.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$L = \begin{cases} 116 * \left(\frac{Y}{Y_n}\right)^{\frac{1}{3}} - 16 & \text{für } \left(\frac{Y}{Y_n}\right) > 0.008856, \\ 903.3 * \frac{Y}{Y_n} & \text{sonst,} \end{cases}$$

$$0 \leq L \leq 100$$

$$a = 500 * \left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right),$$

$$-110 \leq a \leq 100$$

$$b = 200 * \left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right),$$

$$-110 \leq b \leq 100$$

$$X_n = 0.008856$$

$$Y_n = 1$$

$$Z_n = 1.088754$$

$$f(t) = \begin{cases} t^{\frac{1}{3}} & \text{für } t > 0.008856, \\ 7.787 * t + \frac{16}{116} & \text{sonst,} \end{cases} \quad (3.1)$$

X_n , Y_n , Z_n sind die tri-stimulis-Werte. Unsere Augen benötigen drei Werte, um eine Farbe zu definieren. Diese Werte werden **tri-stimulis** genannt [Col05].

In der Abbildung 3.2 steht ein Beispielbild, das Feuer enthält. Hier wurde das Bild in Lab-Farbmodell transformiert und danach jedes Kanalbild extrahiert. Die Kanalbilder werden in den nächsten Schritten analysiert, um das Feuer aus dem Originalbild zu extrahieren.



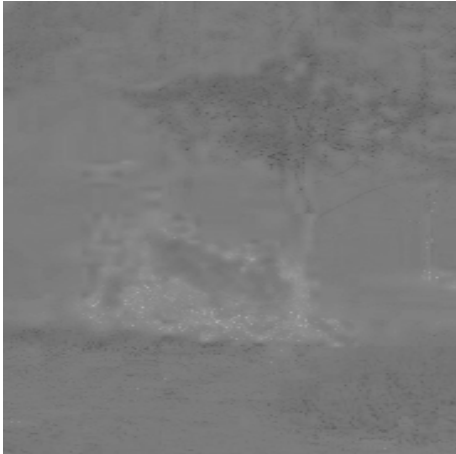
(a) Waldbrand [Kid11]



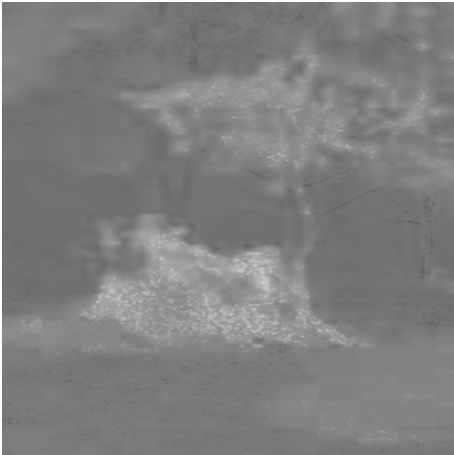
(b) Waldbrand(Lab)



(c) L-Kanal



(d) a-Kanal



(e) b-Kanal

Bild 3.2: Transformation von RGB-Bild in Lab-Bild und L-,a-,b-Kanalbild

3.3.2 Binarisierung

Das Ziel der Binarisierung besteht darin, die Pixel, die Feuer repräsentieren, zu lokalisieren. Eine Binarisierung von L-Bild, a-Bild und b-Bild wird stattfinden. Dazu werden zuerst die Mittelwerte L_m, a_m, b_m berechnet.

$$\begin{cases} L_m = \frac{1}{N} \sum_x \sum_y L(x,y) \\ a_m = \frac{1}{N} \sum_x \sum_y a(x,y), \\ b_m = \frac{1}{N} \sum_x \sum_y b(x,y), \end{cases} \quad (3.2)$$

N ist die Gesamtanzahl von Pixeln in einem Bild; (x,y) die Position eines Pixels. Nun werden die Pixelwerte von L-,a-,b-Kanalbildern in $[0,1]$ normiert.

Wenn die Mittelwerte zur Verfügung stehen, werden die Pixelwerte mit den jeweiligen Mittelwerten verglichen. Die Pixelwerte, die größer oder gleich des Mittelwertes sind, werden als Feuerpixel betrachtet und die Pixelpunkte an den entsprechenden Stellen bekommen den Wert 1. Sonst werden die Pixel einfach unterdrückt, das heißt auf 0 gesetzt.

Das Feuer ist eine der hellsten Regionen in einem Bild und hat fast immer die rote Farbe. Falls ein gelber Pixelwert größer oder gleich eines roten Pixelwertes ist, dann sollte das Pixel als Feuerpixel betrachtet werden. Es heißt, wenn ein Pixel in dem b-Kanalbild größer oder gleich eines Pixels in dem a-Kanalbild ist, wird dieses Pixel als Feuerpixel wahrgenommen.

Die Pixel sind zwar in zwei unterschiedlichen Bildern, aber sie sollten an der gleichen Stelle sein. Dies wird in der Gleichung (3.6) dargestellt.

$$R1(x,y) = \begin{cases} 1 & \text{für } L(x,y) \geq L_m, \\ 0 & \text{sonst,} \end{cases} \quad (3.3)$$

$$R2(x,y) = \begin{cases} 1 & \text{für } a(x,y) \geq a_m, \\ 0 & \text{sonst,} \end{cases} \quad (3.4)$$

$$R3(x,y) = \begin{cases} 1 & \text{für } b(x,y) \geq b_m, \\ 0 & \text{sonst,} \end{cases} \quad (3.5)$$

$$R4(x,y) = \begin{cases} 1 & \text{für } b(x,y) \geq a(x,y) \\ 0 & \text{sonst,} \end{cases} \quad (3.6)$$

Sobald man die entsprechenden Bilder von R1, R2, R3 und R4 hat, werden sie kombiniert. Die Kombination besteht hier aus einer AND-Operation.

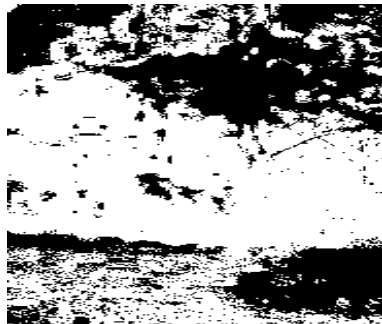
$$K(x,y) = R1(x,y) \oplus R2(x,y) \oplus R3(x,y) \oplus R4(x,y)$$



(a) Eingangsbild



(b) Binärbild L-Kanal



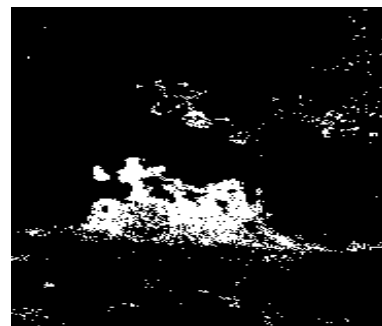
(c) Binärbild a-Kanal



(d) Binärbild b-Kanal



(e) Binärbild nach Gleichung 3.6



(f) Kombiniertes Bild

Bild 3.3: *Binarisierung von Bildern*

Gemeinsame Pixel werden in dem Bild(3.4(b)) dargestellt und nicht gemeinsame Pixel

werden einfach unterdrückt.

3.3.3 Segmentierung

Das Bild 3.4(b) wird dann Pixel für Pixel mit dem Originalbild verglichen, um die Feuerregionen zu extrahieren.

$$F(x,y) = \begin{cases} O(x,y) & \text{für } O(x,y) > 0 \wedge K(x,y) = 1 \\ 0 & \text{sonst,} \end{cases} \quad (3.7)$$

$F(x,y)$ segmentiertes Bild; $O(x,y)$ originales Bild.

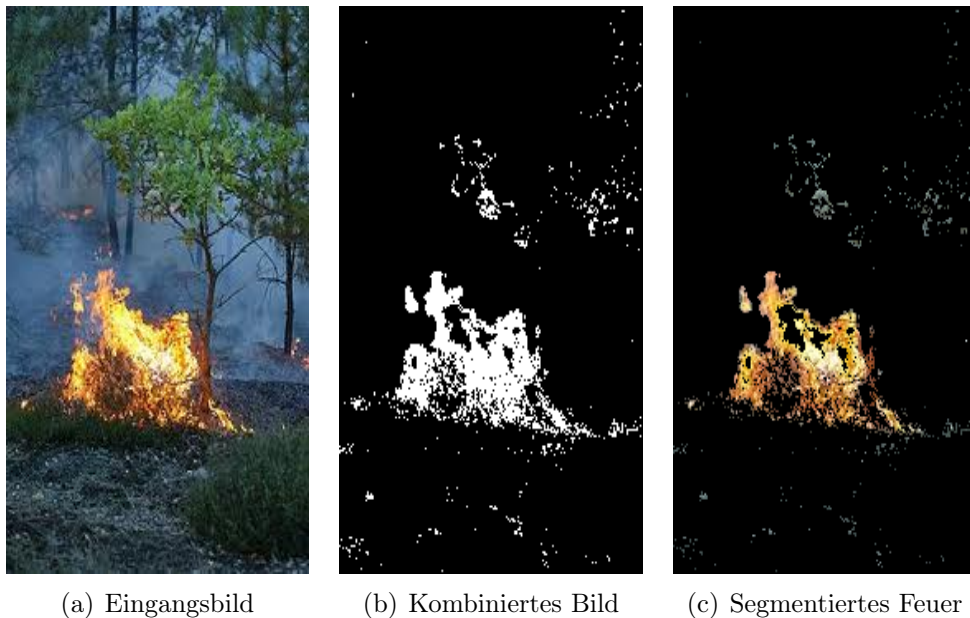


Bild 3.4: *Segmentierung*

In dem Bild 3.4(c) werden eine Feuerregion und Regionen, die kein Feuer darstellen, erkannt.

3.3.4 Wiederholung

Das segmentierte Bild 3.4(c) wird weiter verarbeitet. Das Bild wird wieder in ein Lab-Farbmodell konvertiert und die unterschiedlichen Kanalbilder extrahiert. Eine Binarisierung wird durchgeführt und am Ende die Segmentierung.

Alle Verfahren, die hier benutzt werden, um das extrahierte Feuerpixel 3.4(c) zu erhalten, werden erneut auf dem Bild(3.4(c)) angesetzt. Am Ende wird die Feuerregion hervorgehoben.

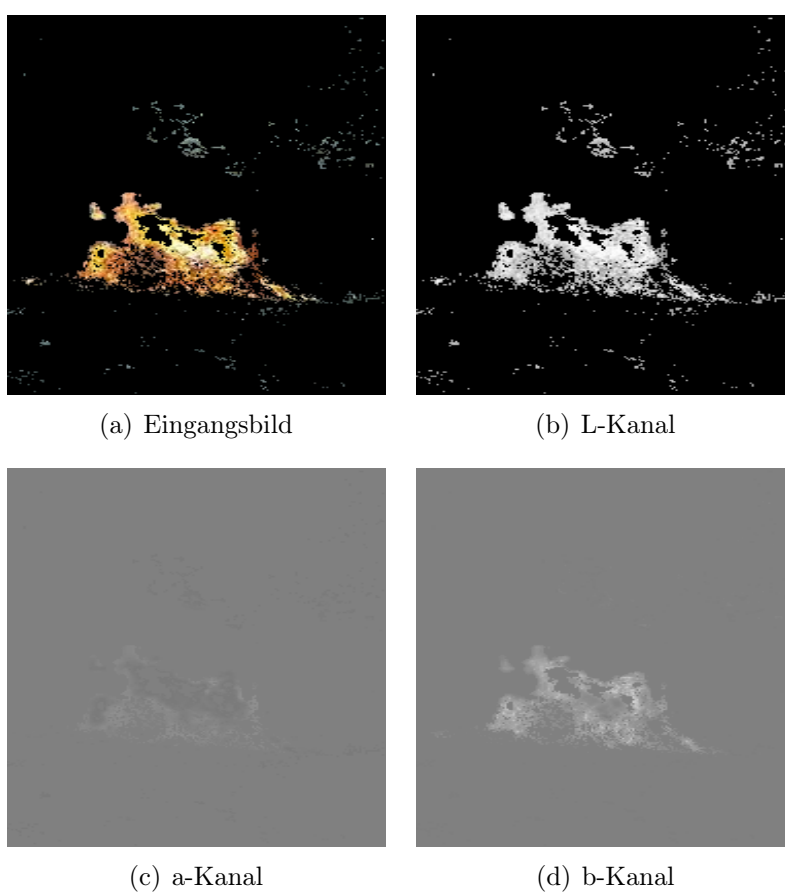


Bild 3.5: *Segmentiertes Feuer in Lab-Farbmodell*

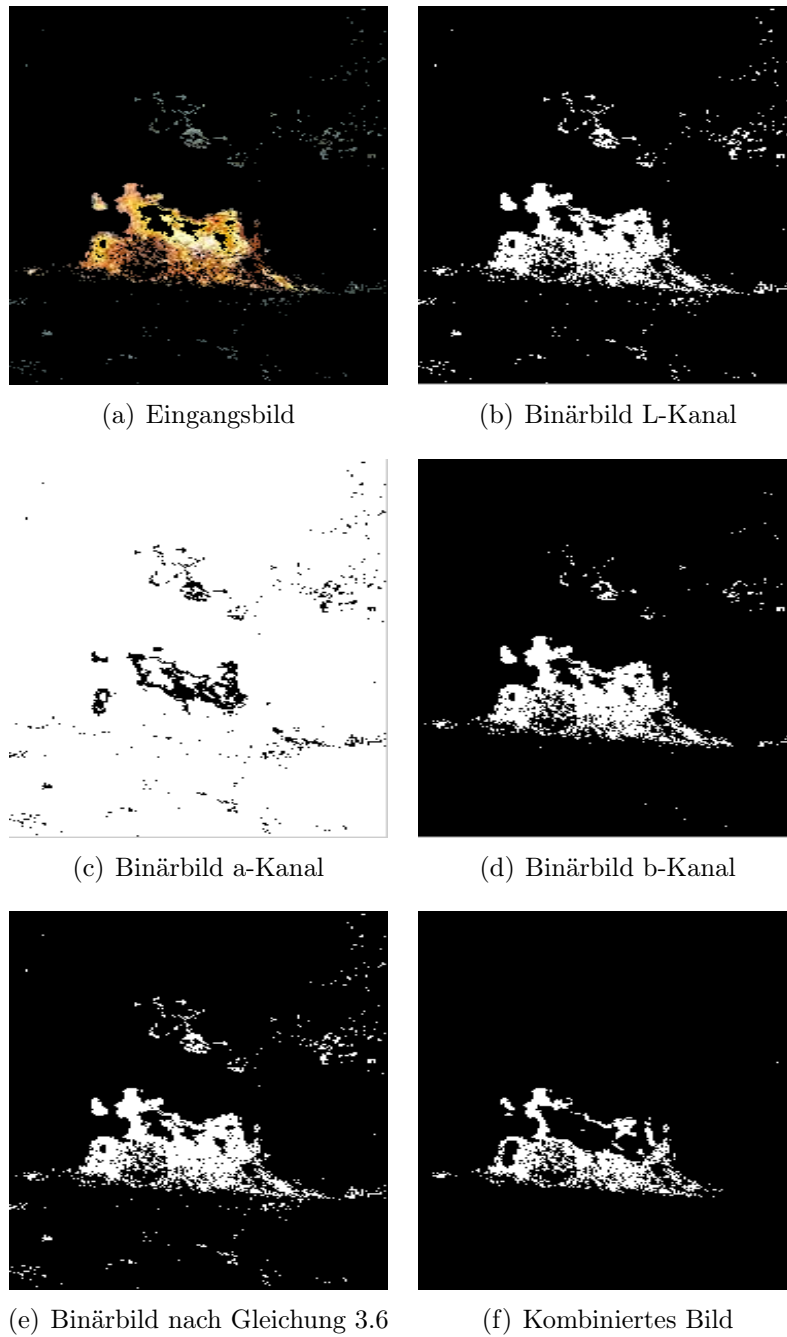
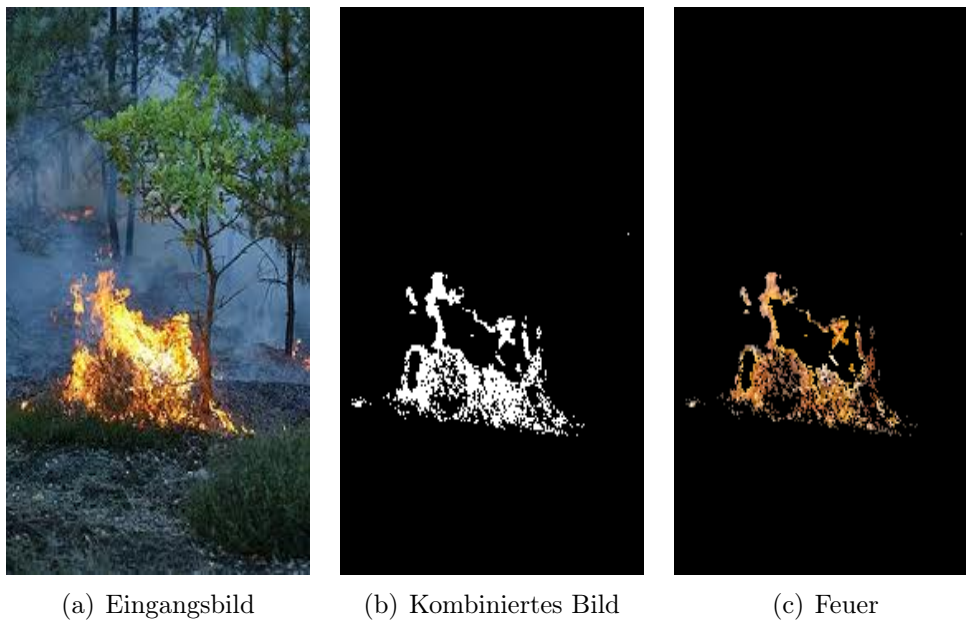


Bild 3.6: *Binarisierung von Bildern des segmentierten Feuers*



(a) Eingangsbild

(b) Kombiniertes Bild

(c) Feuer

Bild 3.7: *Detektiertes Feuer*

Das Verfahren basiert auf der Analyse von Farben und ist dazu geeignet, Feuer auf einem Bild zu lokalisieren. Dennoch ist es nicht sehr zuverlässig, weil die falsche Detektionsrate erhöht wird. Es bedeutet, dass Gegenstände, die die gleichen Farben wie Feuer haben, auch als Feuer detektiert werden.

Kapitel 4

Videobasierte Feuererkennung

Zu den vielfältigen Anwendungsmöglichkeiten der Bildverarbeitung gehört die Videotechnik, da die Bildverarbeitung sowohl den Bereich der Festbilder als auch den der Bewegtbilder umfasst [KL11b].

Ein Video repräsentiert Abbildungen, die sich mit der Zeit verändern oder bewegen. Der Einsatz von Kameras im Bereich der Feuererkennung führt dazu, dass die herkömmlichen Feuererkennungssysteme wie Gassensoren, chemische Sensoren und Temperatursensoren immer seltener verwendet werden. Die Detektion des Feuers mit den herkömmlichen Sensoren hängt meistens von deren Qualität ab; diese sind häufig auch sehr teuer. Der falsche Detektionsanteil ist auch erhöht. Bei der Benutzung eines Temperatursensors kann ein Feueralarm ausgelöst werden, wenn in einem Raum stark geraucht wird. Ein weiteres Problem von chemischen Sensoren besteht darin, dass die Detektion in sehr wenigen Fällen in Echtzeit erfolgt. Mit der rasanten Entwicklung der digitalen Kameras benötigt man nur eine leistungsfähige Software, die die Feuerdetektion in einem UAV in Echtzeit durchführt.

Der Begriff Leistung wirft verschiedene Fragen auf, wie zum Beispiel:

- Ist eine Software leistungsfähiger als alle anderen, wenn ihre Detektion sehr schnell ausgeführt wird?
- Ist eine Software leistungsfähiger als alle anderen, wenn sie fehlerfrei Feuer detektiert und dabei eine sehr große Berechnungszeit benötigt?

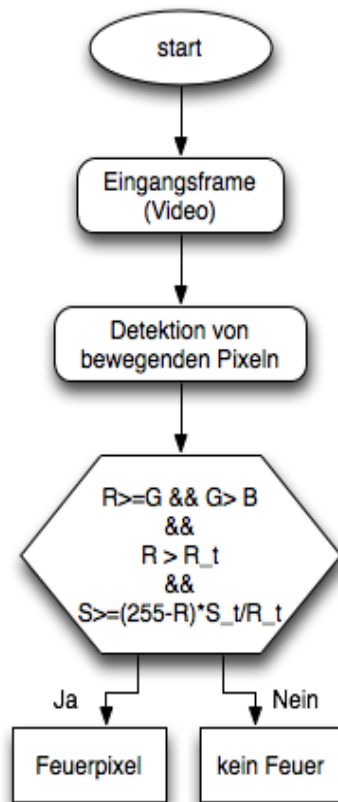
4.1 Überblick der videobasierten Feuererkennung

Heutzutage gibt es viele Recherchen in dem Bereich von videobasierter Feuererkennung. Die meisten davon beschäftigen sich mit der Farbanalyse, um Feuer in Videos zu detektieren. In [HSL⁺93] wurden die Farben des Videos analysiert, um Feuer zu detektieren. Bei [CWC04] wurden drei Farbbedingungen im RGB-Farbmodell erstellt, damit Feuer-

pixel klassifiziert werden können. Danach wurde eine dynamische Analyse der Flammen durchgeführt, um Feuer zu detektieren. In der Forschung von [TKK10] wurden Hidden Markov Modelle verwendet, um die Präsenz des Feuers in einem Video zu entdecken, während [TDGC05] die Wavelet-Analyse angewendet hat. Damit sollten Informationen über die Frequenz des Feuers gewonnen werden.

4.2 Detektion basiert auf Bewegung und Feuerfarben

Die Feuererkennung in einem Video kann grundsätzlich durch das Herausfiltern von bewegenden Objekten und die Analyse des Farbverhaltens durchgeführt werden.



(a) Schematische Darstellung zur Detektion von Feuerpixeln

Bild 4.1: *Detektion von bewegten Pixeln*

Für die vorliegende Arbeit wurden Videos aus dem Internet heruntergeladen. Mit Hilfe von **OpenCV** kann man Videodateien vom Typ *AVI* abspielen und verarbeiten. Hier werden die Videos, mit einer festen Rate, in eine Framesequenz unterteilt.

4.2.1 Detektion von sich bewegenden Pixeln

Normalerweise lassen sich die Bewegungen in einem Video entdecken, wenn man die Intensität von zwei benachbarten Frames miteinander subtrahiert. Dadurch werden alle Pixel, die sich nicht bewegen, unterdrückt.

Es gibt auch die Möglichkeit einen Durchschnittswert von drei benachbarten Frames für alle Pixel zu berechnen (4.1) und dann den Betrag einer Subtraktion von zwei anschließenden Mittelwerten zu berechnen (4.2) [TKK10].

$$h_t(X,Y) = \frac{f_t(X,Y) + f_{t+1}(X,Y) + f_{t+2}(X,Y)}{3}, \quad (4.1)$$

$$Diff_t(X,Y) = |h_{t+1}(X,Y) - h_t(X,Y)|. \quad (4.2)$$

Der Vorteil der Gleichungen (4.1) und (4.2) besteht darin, dass wichtige Informationen behalten werden und unnötige einfach gelöscht werden.

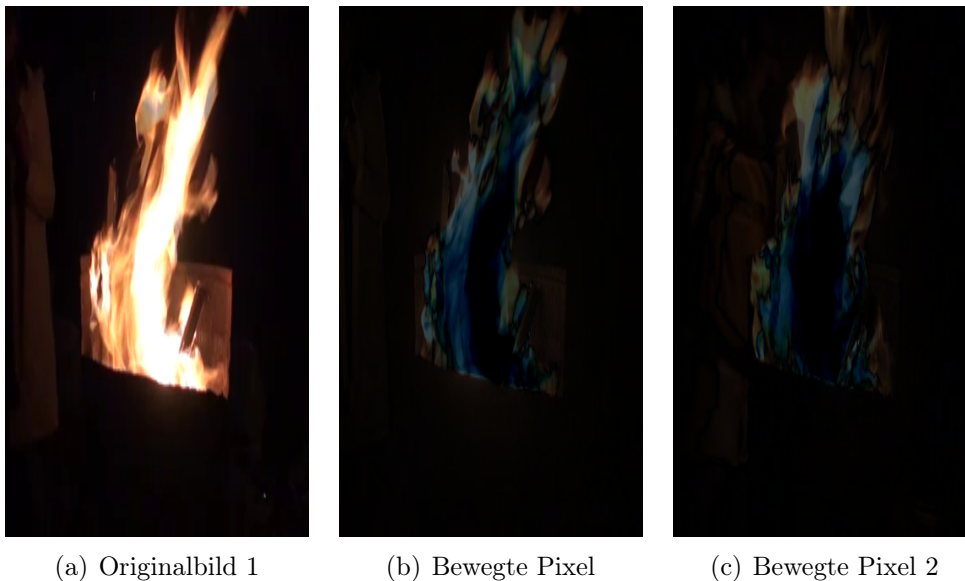


Bild 4.2: Detektion von sich bewegenden Pixeln

Das Ergebnis der Anwendung des Verfahrens ist in der Abbildung (4.2) dargestellt. In dem Frame 4.2(a) hat man ein sehr stark brennendes Feuer mit einer Person im Hintergrund. Die Person bewegt sich nicht, deswegen wird sie in dem Bild 4.2(b) unterdrückt. Sobald sie sich bewegt, wird sie als bewegendes Objekt detektiert. Bild 4.2(c).

4.2.2 Feuerpixel Detektion

Sobald man die bewegten Pixel hat, wird das Verhalten von Farben untersucht. Dabei wird überprüft, ob die einzelnen Pixel die folgenden drei Bedingungen erfüllen.

Wie es in den oberen Kapiteln schon erwähnt wurde, hat das Feuer Farben zwischen Rot und Gelb. Dies bedeutet, dass für ein Feuerpixel folgendes gilt:

$$\mathbf{Rot} \geq \mathbf{Grün}, \mathbf{Grün} > \mathbf{Blau}$$

Das ist die erste Bedingung, die die Farbe des Feuers determiniert [CWC04].

Weil Rot die dominante Farbe des Feuers ist, muss der Rotanteil des Feuerpixels sehr stark sein. Das heißt, der Rotanteil muss größer als eine definierte Schwelle R_T sein. Dadurch kann eine Kamera das Feuer auch in der Nacht eindeutig aufnehmen. Es sollte dann gelten:

$$\mathbf{Rot} > R_T$$

Das ist die zweite Bedingung [CWC04].

Dennoch kann eine Hintergrundhelligkeit die Sättigung von Feuer stören oder das Aussehen eines Objekts verändern, sodass das Objekt als Feuer interpretiert wird. Damit diese falsche Detektion verhindert wird, sollte die Sättigung S des Feuers einen bestimmten Schwellwert nie unterschreiten.

$$S \geq ((255 - Red) * \frac{S_T}{R_T})$$

S_T ist der Wert der Sättigung, wenn Rot den Wert von R_T für das selbe Pixel hat. Der Sättigungswert nimmt ab, während der Rotanteil zunimmt.

$$55 \leq S_T \leq 65 \text{ und } 115 \leq R_T \leq 135$$

$$\begin{cases} \text{Bedingung1} : R \geq G > B \\ \text{Bedingung2} : R > R_T \\ \text{Bedingung3} : S \geq (255 - R) * \frac{S_T}{R_T} \end{cases} \quad (4.3)$$

*WENN (Bedingung1) UND (Bedingung2) UND (Bedingung3) == Richtig
DANN Feuerpixel
SONST Kein Feuerpixel*

Das Bild 4.3 zeigt drei Frames nach der Durchführung der obigen Bedingungen.

Viele Videos mit Feuerinhalten und Videos mit feuerähnlichen Farbinhalten wurden getestet. Die Ergebnisse haben gezeigt, dass sich das vorgestellte Verfahren als schnelles Detektionsverfahren bewerten lässt. Dennoch ist die Rate von falsch positiven Detektionen sehr hoch.

Falsch positive Detektionen bedeuten, dass feuerähnliche Farben als Feuer detektiert werden.



Bild 4.3: Detektion von Feuerpixeln

4.3 Detektion basierend auf Flammenkanten

Das Ziel in diesem Schritt ist ein Verfahren zu benutzen, dass nicht nur Feuer detektiert, sondern auch die Anzahl von falschen Detektionen reduziert.

Die Arbeit in diesem Abschnitt baut auf Vorarbeiten aus der Literatur [ZYW⁺10] auf.

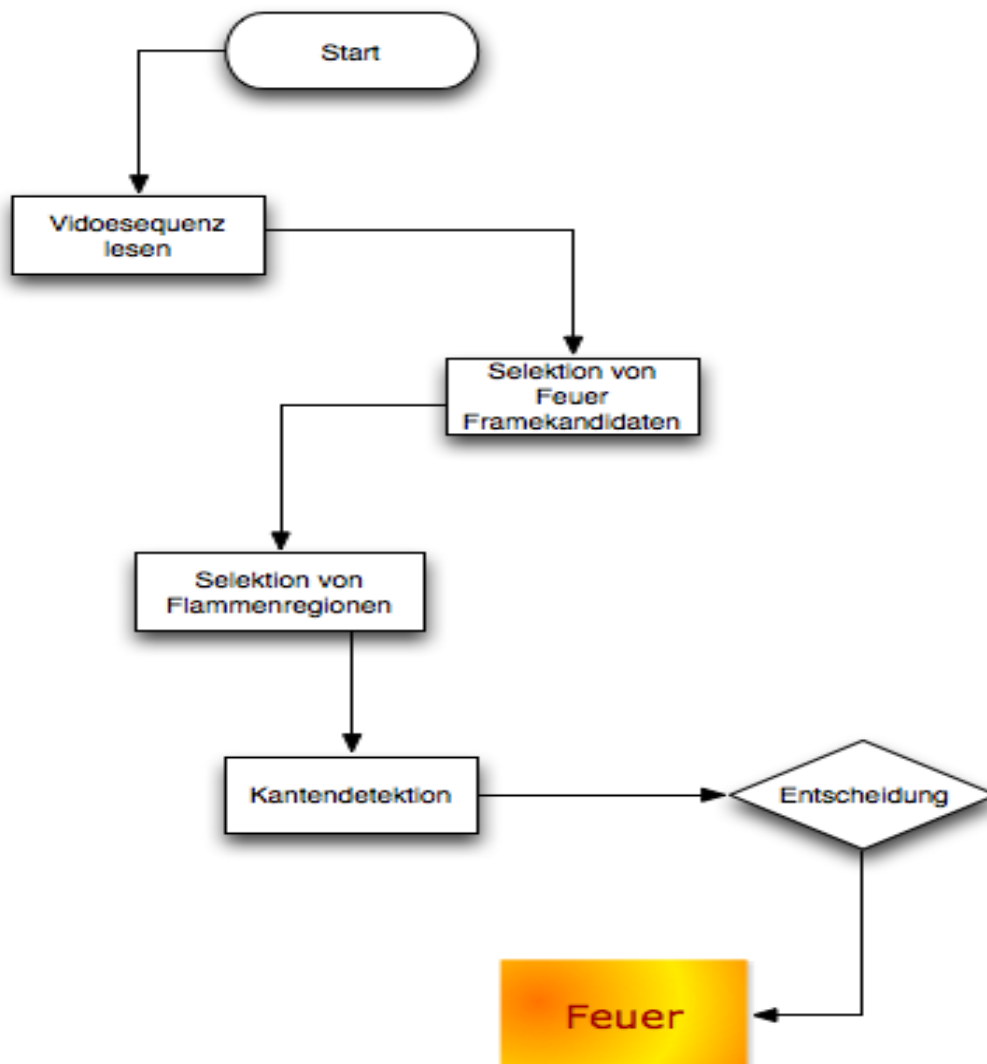


Bild 4.4: Schritte der Detektion basierend auf Flammenkanten

Das Verfahren besteht aus vier Schritten:

1. Selektion von Feuer-Framekandidaten
2. Selektion von Flammenregionen
3. Kantendetektion
4. Entscheidung

4.3.1 Selektion von Feuer-Framekandidaten

Feuer ist nicht zwingend in jedem Framevideo vorhanden. Damit die Leistung der gesamten Detektion erhöht wird, werden zunächst Frames selektiert, die Feuer oder feuerähnliche Elemente beinhalten.

$$\begin{cases} F \text{ ist ein Feuerpixel} & \text{für } F_R > R_T, F_G > G_T, F_B > B_T \\ F \text{ ist kein Feuerpixel} & \text{sonst} \end{cases} \quad (4.4)$$

F_R ist der Wert der roten Farbe, F_G ist der Wert der grünen Farbe und F_B ist der Wert der blauen Farbe.

Hier wird ein Pixel als Feuerpixel betrachtet, wenn sein Rotanteil, sein Grünanteil und sein Blauanteil die Gleichung (4.4) erfüllen.

$$R_T = 180, G_T = 110, B_T = 50$$

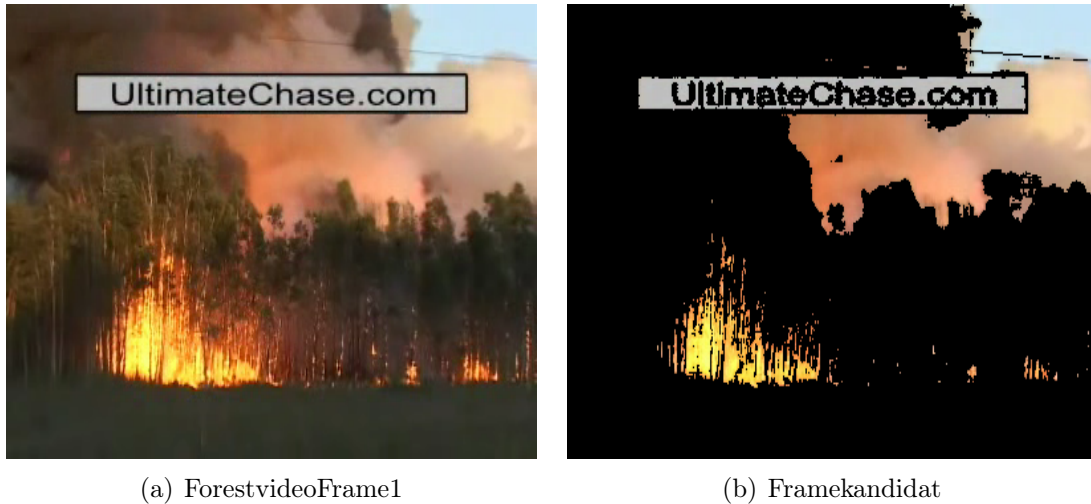


Bild 4.5: Selektion von Feuer-Framekandidaten

Die erhaltenen Feuer-Framekandidaten Bild 4.5(b) werden jetzt analysiert, um die tatsächlichen Feuerregionen zu extrahieren.

4.3.2 Selektion von Flammenregionen

In diesem Arbeitsschritt ist es, herauszufinden, ob die ausgewählten Pixel die richtigen Flammenpixel sind. Alle existierenden Flammenregionen werden selektiert, dafür werden auf die Bedingungen in (4.5) geachtet.

$$\begin{cases} F \text{ ist ein Flammenpixel} & \text{für } F_R > R_T, F_R \geq F_G > F_B, F_S \geq (255 - F_R) * \frac{S_T}{F_R} \\ F \text{ ist kein Flammenpixel} & \text{sonst} \end{cases} \quad (4.5)$$

F_S ist der Wert der Sättigung und $S_T = 60$.
Das Gleichungssystem (4.5) wurde im Abschnitt 4.2.2 erläutert.

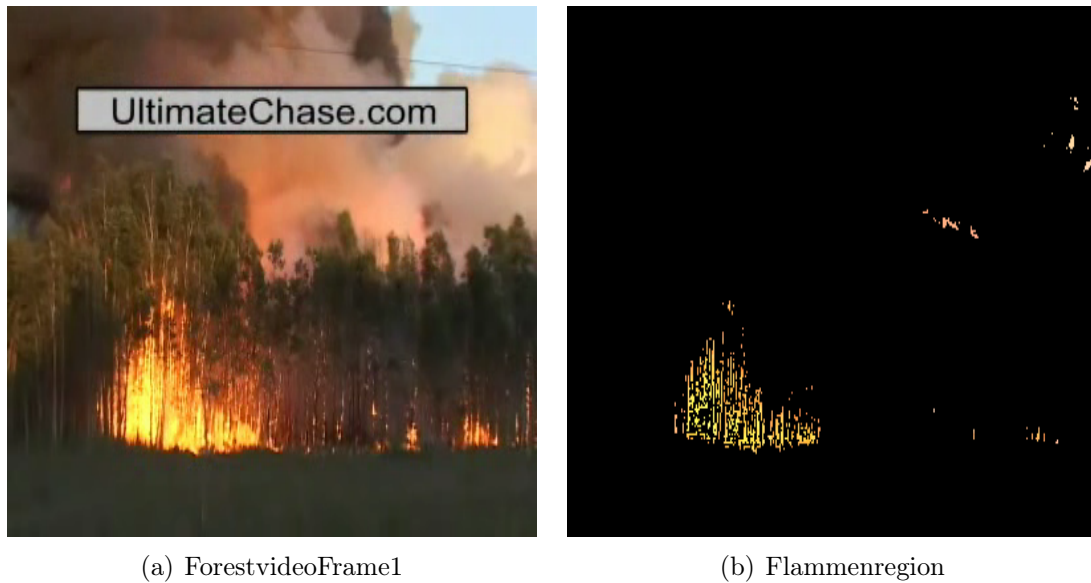


Bild 4.6: *Selektion von Flammenregionen*

Das Bild 4.6(b) enthält die vorhandene Feuerregion.

4.3.3 Kantendetektion

Dieser Abschnitt spielt eine wichtige Rolle bei der Entscheidung, ob ein echtes Feuer detektiert wurde oder ob es sich um eine Fehldetektion handelt. Zunächst werden Frames mit Flammenregionen in graue Bilder konvertiert und binarisiert. Dann werden Operationen wie die Dilatation und die Erosion durchgeführt. Am Ende wird das Verfahren von Canny zur Kantendetektion verwendet.

4.3.3.1 Graubild

Die Farbbilder werden in Graubilder umgewandelt, weil mit diesen Bildern Kanten einfach erkannt werden können.

$$F_{GB} = 0.59 * F_R + 0.3 * F_G + 0.11 * F_B.$$

F_{GB} ist ein Pixelwert in einem Grauwertbild.

F_R ist der Wert des roten Pixels.

F_G ist der Wert des grünen Pixels.

F_B ist der Wert des blauen Pixels.

4.3.3.2 Binarisierung

In Grauwertbildern ist es schwer, die Lage, den Umriss oder die Form von Objekten zu erkennen. Inhomogene Beleuchtung und Rauschen machen eine Erkennung sehr schwer. Schwarz-Weiß-Bilder sind für diesen Aufgabenbereich weitaus besser geeignet. Natürlich sollten bei der Umwandlung von Grauwerten in Schwarz-Weiß-Darstellung möglichst viele Informationen erhalten bleiben [Bru02].

4.3.3.3 Dilatation und Erosion

Durch die Binarisierung entstehen an manchen Stellen des Binärbildes Löcher oder isolierte Pixelpunkte, die für die Kantendetektion nicht optimal sind. In der digitalen Bildverarbeitung können bestimmte Operationen durchgeführt werden, um den erwähnten Problemen vorzubeugen.

1. Dilatation

Die Dilatation fügt einem Bild X ein Strukturelement B zu. Deshalb ist es auch eine bildverstärkende bzw. bildverdickende Operation. Es ist möglich Teilchengruppen zu vereinigen, Löcher zu füllen oder Risse zu schließen [Kre09].

2. Erosion

Die Erosion ist das Gegenstück der Dilatation und entfernt die passenden Bildpunkte. Die Erosion vermindert somit die Bilddaten. Anders gesagt, findet die Erosion Bildpunkte, die mit einem speziellen Muster von Elementen umgeben sind. Bei der Erosion werden schmale Stellen und kleine Objekte, deren Größe kleiner als die des Strukturelementes B ist, vollständig eliminiert [Kre09].

4.3.3.4 Canny-Verfahren

Das Canny-Verfahren wurde nach John Canny benannt und ist mit Abstand das weit verbreitetste Verfahren zur Kantendetektion.

Das Verfahren besteht aus vier Schritten [FB03]:

1. **Reduktion des Bildrauschens mit Gauß-Filter**

Mit der Ableitung kann man die Intensitätsänderung lokalisieren. Die Bildfunktion $f(x,y)$ wird mit den Richtungsableitungen der Gauß-Funktion in x - und y -Richtung gefaltet [Jog05].

Die Gauß-Funktion ist gegeben durch:

$$G_{\sigma}(x,y) = \exp\left(\frac{-(x^2 + y^2)}{2\sigma^2}\right) \quad (4.6)$$

Und die Faltung ist definiert durch:

$$D_x(x,y) = \frac{\partial G_\sigma(x,y)}{\partial x} * f(x,y) \quad (4.7)$$

$$D_y(x,y) = \frac{\partial G_\sigma(x,y)}{\partial y} * f(x,y) \quad (4.8)$$

2. Kanten hervorhebung über den Gradientenbetrag

Die Änderungsstärke der Intensität lässt sich mit dem Betrag des Gradienten definieren.

$$D(x,y) = \sqrt{D_x(x,y)^2 + D_y(x,y)^2} \quad (4.9)$$

3. Kantenlokalisierung: Nicht-Maximum-Unterdrückung

Da die Bildfunktion und das Ergebnis der Filteranwendung nur auf einem diskreten Gitter vorliegen und die Gradientenrichtung im Allgemeinen nicht genau in Richtung eines Gitterpunktes zeigt, werden zwei Gradientenwerte, die der Gradientenrichtung am nächsten liegen, durch lineare Interpolation berechnet. Hierzu wird zuerst aus dem Gradienten \vec{D} die Normale \vec{n} zur Kante bestimmt [Jog05]:

$$\vec{n}(x,y) = \frac{\vec{D}}{\sqrt{D_x(x,y)^2 + D_y(x,y)^2}} = n_x \vec{e}_x + n_y \vec{e}_y \quad (4.10)$$

Sei die Schwellwerte D_1, D_2 ; mit:

$$D_1 = \frac{n_x}{n_y} D(x+1, y-1) + \frac{n_y - n_x}{n_y} D(x, y-1) \quad (4.11)$$

$$D_2 = \frac{n_x}{n_y} D(x-1, y+1) + \frac{n_y - n_x}{n_y} D(x, y+1) \quad (4.12)$$

Mit diesen Werten können nutzbare Maxima für die Bestimmung von Kanten definiert werden.

$D(x,y)$ ist nur dann ein Maximum, wenn sowohl $D(x,y) \geq D_1$ als auch $D(x,y) \geq D_2$ gilt [Jog05].

4. Kantenlokalisierung: Hysterese

Das Hysterese-Verfahren wird benutzt, um die Kanten zu verdeutlichen. Hier wird ein Intervall $[T_{high}, T_{low}]$ bestimmt. Wenn der Gradientenbetrag größer oder gleich T_{high} ist, dann wird der entsprechende Punkt als Kantenpunkt betrachtet. Liegt der Gradientenbetrag über T_{low} und hat der entsprechende Punkt eine direkte Verbindung zu einem Kantenpunkt, wird der Punkt auch als Kantenpunkt betrachtet. Sonst wird es einfach verworfen.

4.3.4 Entscheidung

In [ZYW⁺10] wird die Entscheidung getroffen, ob Feuer in einem Video vorhanden ist, nachdem einige dynamische Eigenschaften des Feuers untersucht worden sind. Zu den Eigenschaften gehören:

- **Fläche der Flamme**
- **Umfang der Flamme**
- **Rundheit der Flamme**

Dabei wird bestätigt, ob Feuer vorhanden ist, nachdem ein Prozess beobachtet wurde. In dieser Arbeit wurde eine Erweiterung zum vorhandenen Verfahren durchgeführt. Hier wurde die **Fluktuation** des Feuers zusätzlich analysiert.

Die Entscheidung, ob Feuer erkannt oder nicht erkannt wurde, basiert dann auf drei voneinander unabhängigen Prozessen.

In einem Speicher werden n -Frames hintereinander gespeichert. Darin wird für die gesamten n beobachteten Frames entschieden, ob Feuer vorhanden ist.

«*Fläche*»:

Die Fläche in einem Binärbild oder Grauwertbild ist die Anzahl von Pixeln auf dem Bild. Die Fläche ist die wichtigste Eigenschaft eines digitalen Bildes [Sav11]. Der Waldbrand besteht aus instabilen Flammen. Die Fläche des Brandes wird als die Anzahl von Flammenpixeln definiert. Die Anzahl von Flammenpixeln ist mit der Zeit häufig steigend, insbesondere für Anfangsfeuer.

Sei $N_j^{(s)}$ ein binärer Indikator, der den Wert 1 annimmt, wenn eine gewisse Bedingung erfüllt ist, sonst gleich 0 ist; S_j die Fläche in einem Frame und R_s der Anteil von binären Indikatoren ungleich 0.

$$N_j^{(s)} = \begin{cases} 1 & \text{für } S_j > S_{j-1}, \\ 0 & \text{sonst} \end{cases} \quad (4.13)$$

$$R_s = \frac{\sum_{j=1}^{20} N_j^{(s)}}{20} \quad (4.14)$$

«*Umfang*»:

In der digitalen Bildverarbeitung definiert der Umfang die Summe aller Pixel, die zusammen die Fläche nach außen begrenzen [Neu09].

Sei $N_j^{(c)}$ ein binärer Indikator, der den Wert 1 annimmt, wenn eine gewisse Bedingung erfüllt ist, sonst gleich 0 ist; C_j der Umfang in einem Frame und R_c der Anteil von

binären Indikatoren ungleich 0.

$$N_j^{(c)} = \begin{cases} 1 & \text{für } C_j > C_{j-1}, \\ 0 & \text{sonst} \end{cases} \quad (4.15)$$

$$R_c = \frac{\sum_{j=1}^{20} N_j^{(c)}}{20} \quad (4.16)$$

«Rundheit»:

Definiert die Komplexität eines Objektes. Die Form von Feuer ist sehr instabil und irregulär, während das Profil von fixierten Elementen wie der Sonne, einem roten Auto oder einem roten Trikot regulär ist [YX11].

Sei $N_j^{(r)}$ ein binärer Indikator, der den Wert 1 annimmt, wenn eine gewisse Bedingung erfüllt ist, sonst gleich 0 ist; R_j die Rundheit des Feuers in einem Frame.

$$R_j = \frac{4\pi S_j}{C_j^2} \quad (4.17)$$

$$N_j^{(r)} = \begin{cases} 1 & \text{für } R_{Tmin} \leq R_j \leq R_{Tmax} \\ 0 & \text{sonst} \end{cases} \quad (4.18)$$

$$R_r = \frac{\sum_{j=1}^{20} N_j^{(r)}}{20} \quad (4.19)$$

$R_{Tmin} = 0.05$ und $R_{Tmax} = 0.8$. In diesem Intervall sollten nur Feuerelemente vorhanden sein. Objekte, die wie Feuer aussehen, aber kein Feuer sind, haben häufig eine Rundheit, die größer als 0.8 ist.

«Fluktuation»:

Der mittlere Pixelunterschied wird in dieser Arbeit als Fluktuation betrachtet.

Es ist bekannt, dass Feuer sich ständig bewegt. In dem Artikel «*Schlieren analysis of an oscillating gas-jet diffusion*» wurde bewiesen, dass die Flammenfluktuation des Feuers eine Frequenz von 10Hz beträgt. Dabei spielt es keine Rolle, welche Brennstoffe vorhanden sind [AA99]. Dies führt dazu, dass die Anzahl von Pixeln in einem Frame in der Zeit t ungleich die Anzahl von Pixeln in einem Frame in der Zeit $t - 1$ oder $t + 1$ ist. Nach der Analyse der Ergebnisse wurde bestimmt, dass der mittlere Pixelunterschied für 20 benachbarte Frames in einem Intervall von [15, 29] liegt.

4.3.4.1 Erste Entscheidung zur Feuerdetektion

$$\text{Feuer erkannt} : R_s \geq R_{st}, R_c \geq R_{ct}, R_r \geq R_{rt}, Th \geq 15 \quad (4.20)$$

$R_{st} = 0.5$, $R_{ct} = 0.5$, $R_{rt} = 0.5$ und Th ist der mittlere Pixelunterschied für eine gegebene Anzahl von Frames.

Es kann vorkommen, dass ein Objekt mit feuerähnlicher Farbe die folgende Bedingung erfüllt: $R_s \geq R_{st}, R_c \geq R_{ct}, P_r \geq R_{rt}$; besonders, wenn die Kamera wackelt. Dennoch ist die Wahrscheinlichkeit, dass der mittlere Flammenpixelunterschied größer als 15 wird, gleich null.

Das System wird die Kante des Feuers anzeigen, wenn die Bedingung (4.20) eintritt.

4.3.4.2 Zweite Entscheidung zur Feuerdetektion

$$\text{Feuer erkannt} : R_r \geq R_{rt}, Th \geq 15 \quad (4.21)$$

Die Bedingung (4.20) kann manchmal nicht auftreten, wenn der Abstand zwischen Kamera und der Feuerposition nicht optimal ist. Gilt dennoch die Bedingung (4.21), wird eine Feuerdetektion stattfinden. Wenn die Rundheit eines Elementes sich in dem definierten Intervall befindet und der mittlere Flammenpixelunterschied geeignet ist, dann ist das Element meistens Feuer.

Das System zeigt die Kante des Feuers an, wenn die Bedingung (4.21) eintritt.

4.3.4.3 Dritte Entscheidung zur Feuerdetektion

$$\text{Feuer erkannt} : R_s \geq R_{st} - 1, P_c \geq P_{ct} - 1, Th \geq 29 \quad (4.22)$$

Es fällt zudem auf, dass sich der Wert der Feuerfläche in einem Video nicht immer aufsteigend ändert, sodass die Prozentzahl R_s immer größer oder gleich 0.5 ist. Das Gleiche gilt auch für den Umfang. Dies liegt daran, dass die Feuerintensität in einer Sequenz von Frames sich abwechselnd ändern kann. In diesem Fall wird überprüft, ob die Prozentzahlen für Fläche und Umfang eine gewisse Grenze nicht unterschreiten. $R_s \geq 0.4$, $R_c \geq 0.4$ und der mittlere Flammenpixelunterschied bei $Th \geq 29$ liegt.

Wenn die Bedingung (4.22) erfüllt ist, ist Feuer vorhanden.

Das System wird die Kante des Feuers anzeigen, wenn die Bedingung (4.22) eintritt.

Das Streben nach einer kompletten Vermeidung von falschen Detektionen hat zu einer Verschärfung des Detektionsverfahrens geführt. Bevor das Alarmbild angezeigt wird, wenn einer der drei Entscheidungsprozesse eintritt, wird zuerst auf die nächsten Entscheidungen gewartet.

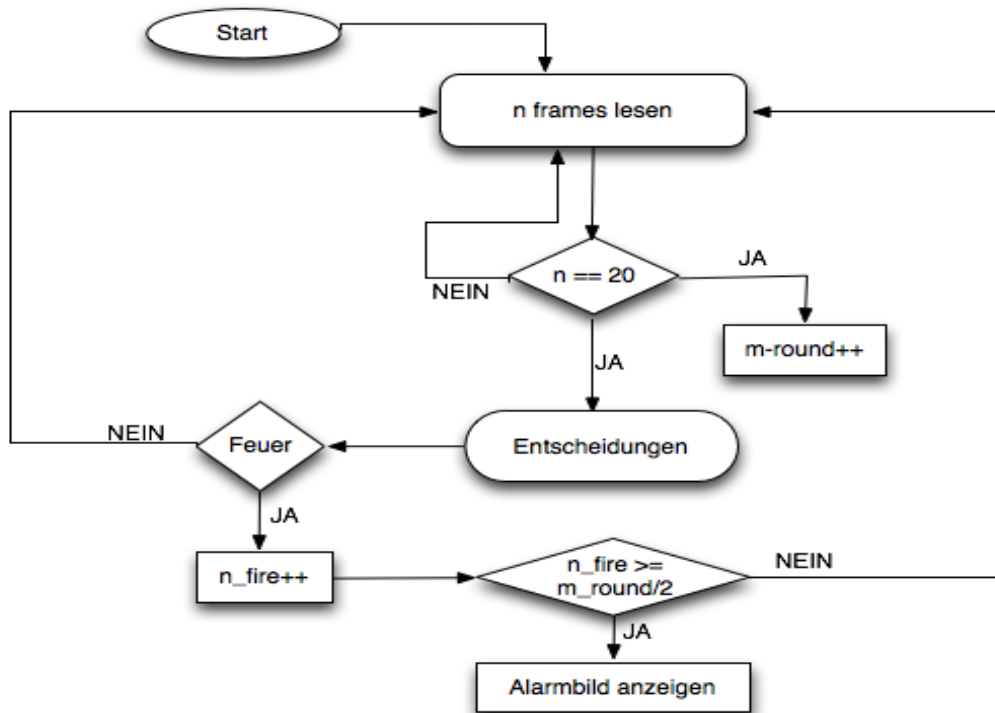


Bild 4.7: Entscheidungsbaum zur Anzeige von Alarmbilder

Zwei Variablen werden als Zähler verwendet.

- m_{round} wird immer um eins erhöht, wenn zwanzig Frames verarbeitet wurden
- n_{Fire} wird immer um eins erhöht, wenn eine der drei Entscheidungen eintrifft

Wenn der Zähler " n_{Fire} " für die eingetroffenen Entscheidungen größer oder gleich die Hälfte des Zählers " n_{Fire} " für die Runde ist, wird das Alarmbild angezeigt. Das Bild 4.7 zeigt eine schematische Darstellung des Verschärfungsprozesses.

4.4 Simulationsversuch

In diesem Abschnitt wird ein Vorschlag präsentiert, der es ermöglicht, anhand der Ergebnisse im Abschnitt (4.3) Feuer zu detektieren.

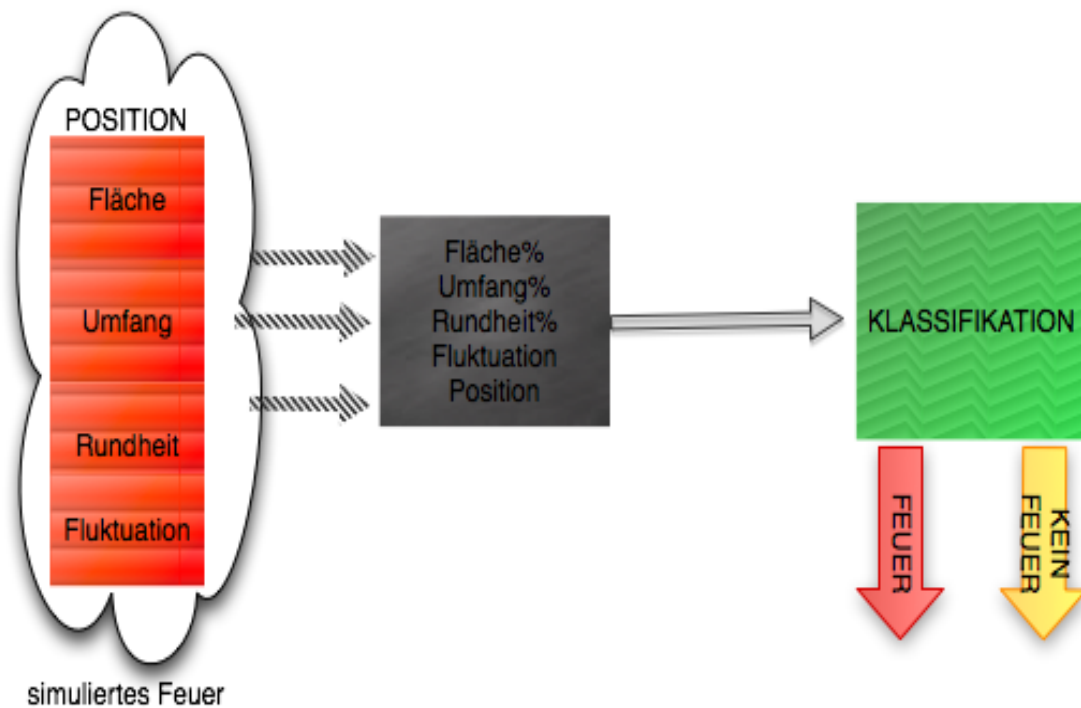
Obwohl die Ergebnisse der Bildverarbeitungsprozesse verwendet werden, sollte hier kein Kamerasensor zur Hilfe genommen werden.

Ein simuliertes Feuer wird in dieser Arbeit durch eine **Lichtstrahlung** definiert. Die Position des Feuers in der simulierten Umgebung wird als Lichtquelle verwendet.

Nun werden die Ergebnisse der *Detektion basierend auf Flammenkanten* des Abschnitts (4.3) als Attribute des Feuersensors benutzt.

4.4.1 Simulation durch Flammenkanten

Das Bild (4.8) zeigt, wie eine Simulation der Feuerdetektion aussehen kann.



(a) Schematische Darstellung der Simulation einer Feuerdetektion

Bild 4.8: *Simulierte Feuerdetektion*

Das Feuer lässt sich durch die folgenden Elemente simulieren:

1. **Fläche**
2. **Umfang**
3. **Rundheit**
4. **Fluktuation**
5. **Position**

In der Simulationsumgebung könnte sich ein Feuer oder ein feuerähnliches Element befinden. Ein Sensor, der sich in einer optimalen Position befindet, kann die Informationen, die das Element definieren, empfangen.

Der Sensor ist geregelt, so dass er für eine bestimmte Zeit T , n -Flächenwerte, n -Umfangswerte, n -Rundheitswerte, den Fluktuationswert und die Position aufnehmen und speichern kann. In dieser Arbeit wurde n auf 20 gesetzt.

Ein Feuersensor wird dann die folgende Regelung oder Einstellung haben:

Sensoreinstellung:

$\text{maxAbs} = 10$

$\text{FlächenAbs} = 6$

$\text{UmfangsAbs} = 6$

$\text{RundheitsAbs} = 6$

$\text{FluktuationsAbs} = 6$

Mit dieser Konfiguration erkennt der Sensor ab $10m$ die Position des Feuers. Die genaue Abhängigkeit der Sensorempfindlichkeit vom Abstand hängt von vielen Faktoren ab. Wenn sich der Sensor in die Richtung eines Waldbrandes bewegt, ist der maximale Abstand zwischen dem Sensor und dem Feuer deutlich größer als der Abstand zwischen dem Sensor und einem Feuer, das aus Kerzen besteht. Dies lässt sich durch die Intensitätsstärke erklären. Eine Lichtquelle, die einen Waldbrand simuliert, hat eine eindeutig stärkere Intensität als die Intensität einer Lichtquelle, die ein Kerzenfeuer simuliert. Ab $6m$ kann der Sensor die Fläche, den Umfang, die Rundheit und die Fluktuation des Feuers aufnehmen.

Mit maxAbs ist der maximale Abstand, um die Position des Feuers zu erhalten, ausgedrückt.

FlächenAbs ist eine Variable, die den Abstand zum Empfang von Flächenwerten definiert.

Mit UmfangsAbs ist der Abstand definiert, der den Empfang von Umfangswerten zulässt.

RundheitsAbs und FluktuationsAbs sind die Variablen für den Empfang von Rundheitswerten und Fluktuationswerten.

Die festgelegten Werte in den Sensoreinstellungen sind als Dummy-Werte zu verstehen, weil die Abstandswerte nur experimentell definiert werden können.

In der folgenden Tabelle (4.1) werden die Werte von zwanzig hintereinander folgenden Feuer-Frames und die Position des Feuers angegeben. Die Werte, die in der Tabelle vorhanden sind, werden Bestandteil der Sensorinformation.

Fläche	Umfang	Rundheit	Fluktuation	Position
1376	252	0.27	29	6
1455	252	0.29	29	6
1395	251	0.28	29	6
1397	246	0.29	29	6
1310	241	0.28	29	6
1379	247	0.28	29	6
1356	244	0.28	29	6
1352	259	0.25	29	6
1339	261	0.25	29	6
1319	255	0.25	29	6
1277	254	0.25	29	6
1325	251	0.26	29	6
1294	249	0.26	29	6
1317	253	0.26	29	6
1322	256	0.25	29	6
1309	251	0.26	29	6
1304	248	0.27	29	6
1304	260	0.24	29	6
1308	267	0.23	29	6
1247	258	0.23	29	6

Tabelle 4.1: Ergebnis der Detektion basierend auf Flammenkanten

Die Information, die der Sensor sendet, kann die folgende Form haben:

Sensor – Info:

1. $Pos(3.22\ 3.56\ 3.01)$
2. $Fla = 0.44\ m^2$
3. $Umf = 0.09\ m$
4. $Rnd = 0.23$
5. $Fluc = 29$

Diese Information (*Sensor-Info*) bedeutet, dass sich ein Feuer mit einer Fläche von $0.44\ m^2$, einem Umfang von $0.09\ m$, einer Rundheit von 0.23 und einer Fluktuation von 29 an der Stelle $(3.22\ 3.56\ 3.01)$ befindet.

In der Tabelle sind die Werte von der Fläche und dem Umfang in Pixel vorhanden. Da in der Simulationsumgebung keine Bildverarbeitungsmethode benutzt werden sollte, wurde die Pixeleinheit in die Längeneinheit umgewandelt.

Basierend auf eine Kamera mit einer festen Auflösung von 72 *ppi*, kann eine Umwandlung von Pixel zu Meter wie folgt definiert werden:

ppi steht für **P**ixel **p**ro **i**nch.

$$1 \text{ pixel} = \frac{1 \cdot 0.0254}{72} = 3,53 \times 10^{-4} \text{ m}$$

1 *inch* = 0.0254 *m*. *m* steht für Meter.

Nach der Speicherung dieser Werte oder nach dem Empfang der Sensor-Nachricht werden Flächenwerte miteinander verglichen. Die Umfangswerte werden ebenso miteinander verglichen. Die Rundheitswerte werden mit zwei Schwellen-Rundheitswerten verglichen. Wenn die empfangenen Werte einige Bedingungen erfüllen, dann werden die Positionswerte des Feuers in sphärische Koordinaten umgewandelt. Die Umwandlung findet hier statt, weil das Feuer als Lichtstrahlung dargestellt wird. Die Lichtstrahlung sollte ein Kreis sein.

Die Größe des Kreises hängt von der Entfernung zwischen dem Feuer und dem Sensor ab. Wenn die Entfernung des Feuers groß ist, dann wird der Kreis sehr klein sein. Eine kleinere Entfernung des Feuers führt zu einem größeren Kreis.

Die Erläuterungen in diesem Abschnitt basierend auf [Hol07].

Kapitel 5

Ergebnisse und Auswertung

Für die Implementierung der Arbeit wurde ein Computer mit einem **Intel (R) Core(TM) i5 CPU, 3.20 GHz** Prozessor verwendet. Als integrierte Entwicklungsumgebung wurde **Microsoft Visual Studio 2010** verwendet.

Es wurde im Kapitel 3 und Kapitel 4 gezeigt, dass Bilder und Videos analysiert werden können, um Feuer zu detektieren.

In dem Kapitel 3 wurde ein Bild verwendet, in dem ein Feuer eindeutig zu sehen ist. Ebenso wurde in dem Kapitel 4 zwei Videos benutzt, in denen das Vorhanden von Feuer genau sichtbar war.

Dennoch kann eine genauere Auswertung der Verfahren nur gemacht werden, wenn die Ergebnisse von unterschiedlichen Bildern und Videos zur Verfügung stehen.

5.1 Ergebnisse der bildbasierten Feuerdetektion

In diesem Abschnitt wird das Verfahren **Bildbasierte Feuerdetektion** mit unterschiedlichen Bildinhalten und Bildtypen getestet:

1. Bild mit eindeutigem Feuer
2. Bild mit Feuer und nicht Feuer-Elementen
3. Bild ohne Feuer
4. Simulierte Bilder
5. Verrauschte Bilder

Die fünf Testbilder wurden ausgesucht, weil aus deren Ergebnissen effektiv festgestellt wird, ob das Feuer richtig detektiert wurde oder ob die Detektion fehlerhaft ist.



Bild 5.1: Testbilder

Das Bild 5.2(a) enthält die Aufnahme von einem Waldbrand und hat eine Bildgröße von $106 \times 80 \text{ Pixel}$. Nach der Ausführung des Algorithmus der Feuerdetektion auf dem Bild wird als Zwischenschritt das Bild 5.2(b) erhalten; es handelt sich hier um ein binäres Bild, das verwendet wird, um das Endbild der Detektion zu bekommen: Bild 5.2(c). Das Ergebnis ist in diesem Fall sehr gut, weil zu erkennen ist, dass alle Elemente in dem Bild unterdrückt worden sind, die kein Feuer repräsentieren.

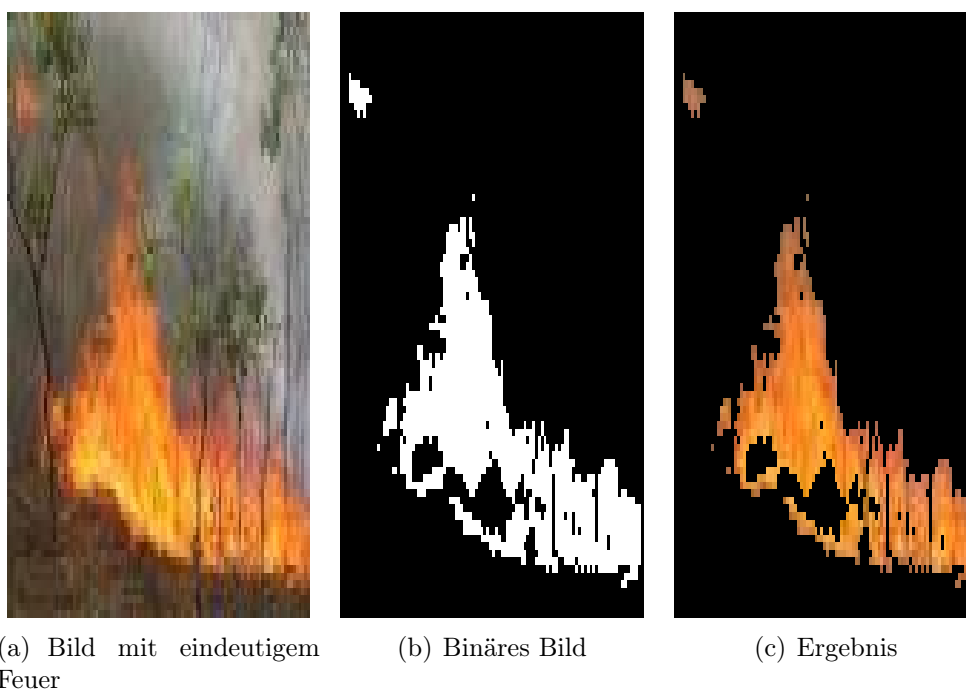


Bild 5.2: Testbild mit Feuer

Das Bild 5.3(a) enthält im Vergleich zu dem Bild 5.2(a) nicht nur Feuer, sondern auch Menschen, Bäume und Himmel. Die Bildgröße ist $600 \times 400 \text{ Pixel}$. Nach der Binarisierung wird auf dem Bild 5.3(b) erkannt, dass einige Objekte vorhanden bleiben, obwohl

sie kein Feuer darstellen. Auf dem Bild 5.3(c) ist zu sehen, dass nicht nur das Feuer detektiert wird, sondern auch Teile des menschlichen Körpers.

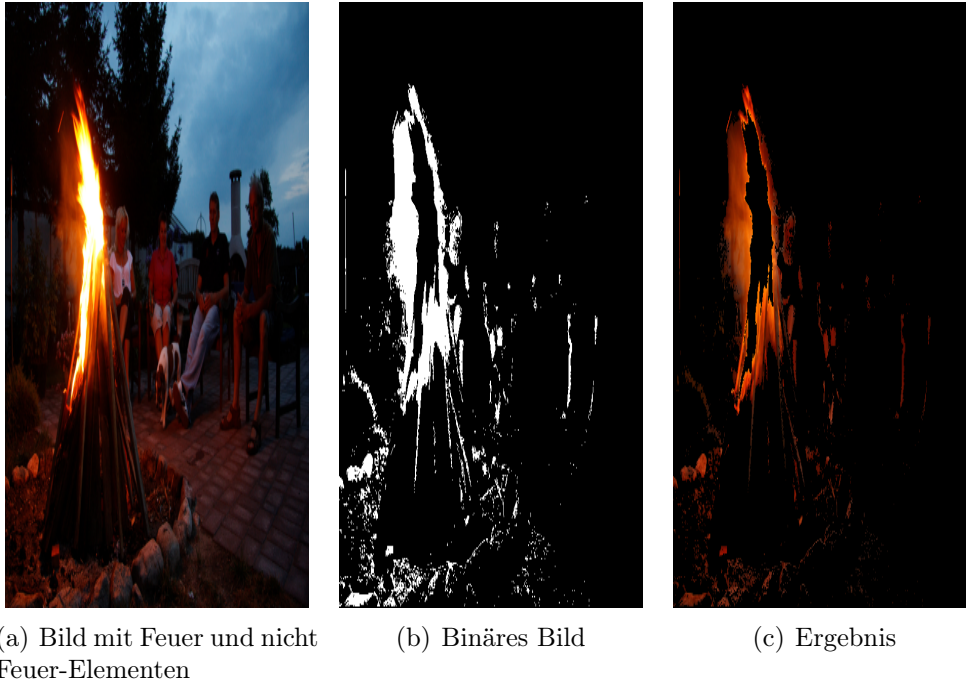


Bild 5.3: Testbild mit Feuer und nicht Feuer-Elementen

In dem letzten Testbild haben wir überhaupt kein Feuer und die Bildgröße beträgt $500 \times 375 \text{ Pixel}$. Nach der Durchführung des Verfahrens werden leider manche Teile des Himmels als Feuer detektiert.

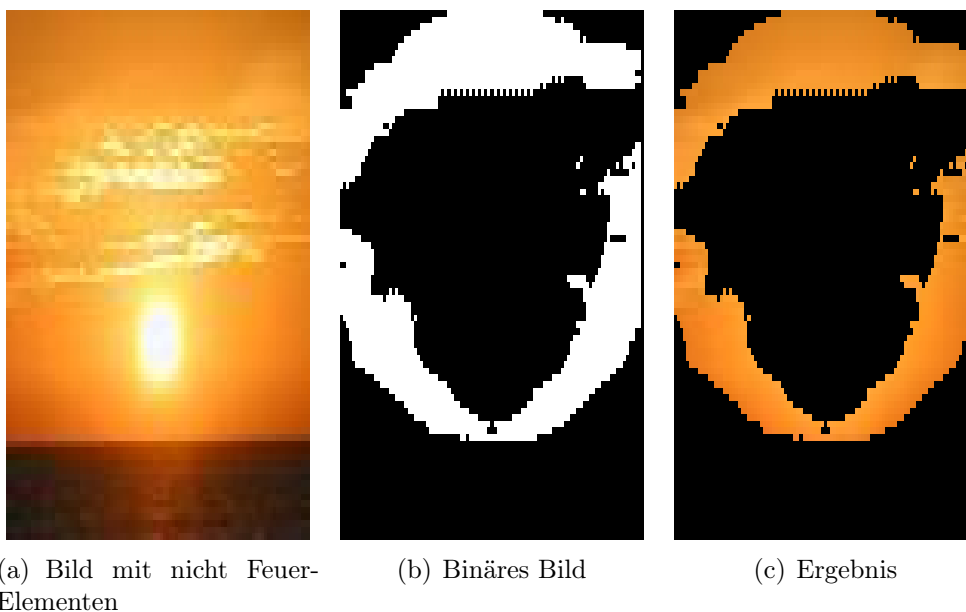


Bild 5.4: Testbild mit nicht Feuer-Elementen

5.1.1 Ergebnisse von simulierten und verrauschten Bildern

Aus dem Internet wurden drei simulierte Testbilder heruntergeladen. Mit den Bildern sollte überprüft werden, wie die Ergebnisse des Verfahrens in einem Simulationsprogramm aussehen könnten.

Mit Hilfe des Programms "GIMP" wurde ein Bild mit Rauschen verarbeitet. Die Idee dahinter ist, herauszufinden wie zuverlässig das Detektionsverfahren im Fall von ungünstigem Wetter oder von einer schlechten Bildqualität sein kann. Dazu wurde das Bild mit gauß'schem Rauschen und gleichverteiltem Rauschen geändert. Die Standardabweichung in beiden Fällen ist $\sigma = 31.1$.

Gleichverteiltes Rauschen hat eine Rechteck-Verteilung und ist unabhängig von dem Pixelwert. Gleichverteiltes Rauschen kann durch Quantisierungsfehler bei der analog-digital Konvertierung entstehen.

Gauß'sches Rauschen ist standard verteiltes Rauschen, welches unabhängig von dem Pixelwert ist. Gauß'sches Rauschen entsteht durch thermische Fluktuationen.¹

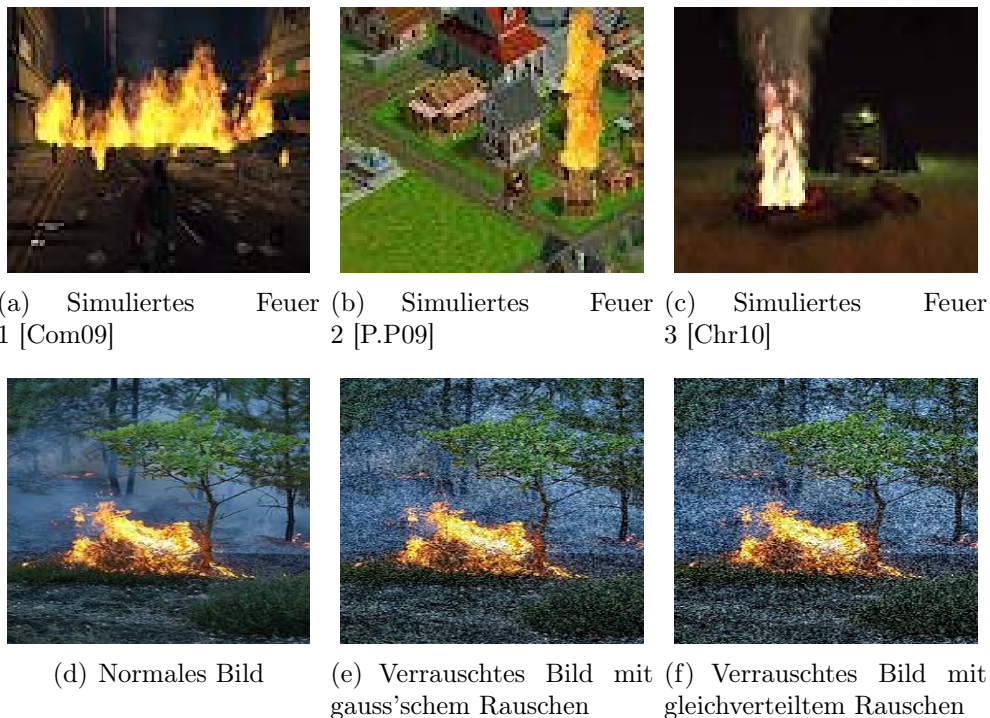


Bild 5.5: Simulierte Testbilder in der ersten Reihe und verrauschte Bilder in der zweiten Reihe

¹Die Definition vom Gauß'schen und Gleichverteilten Rauschen befindet sich in dem Programm GIMP unter Filter und dann Rausch-Generator finden.

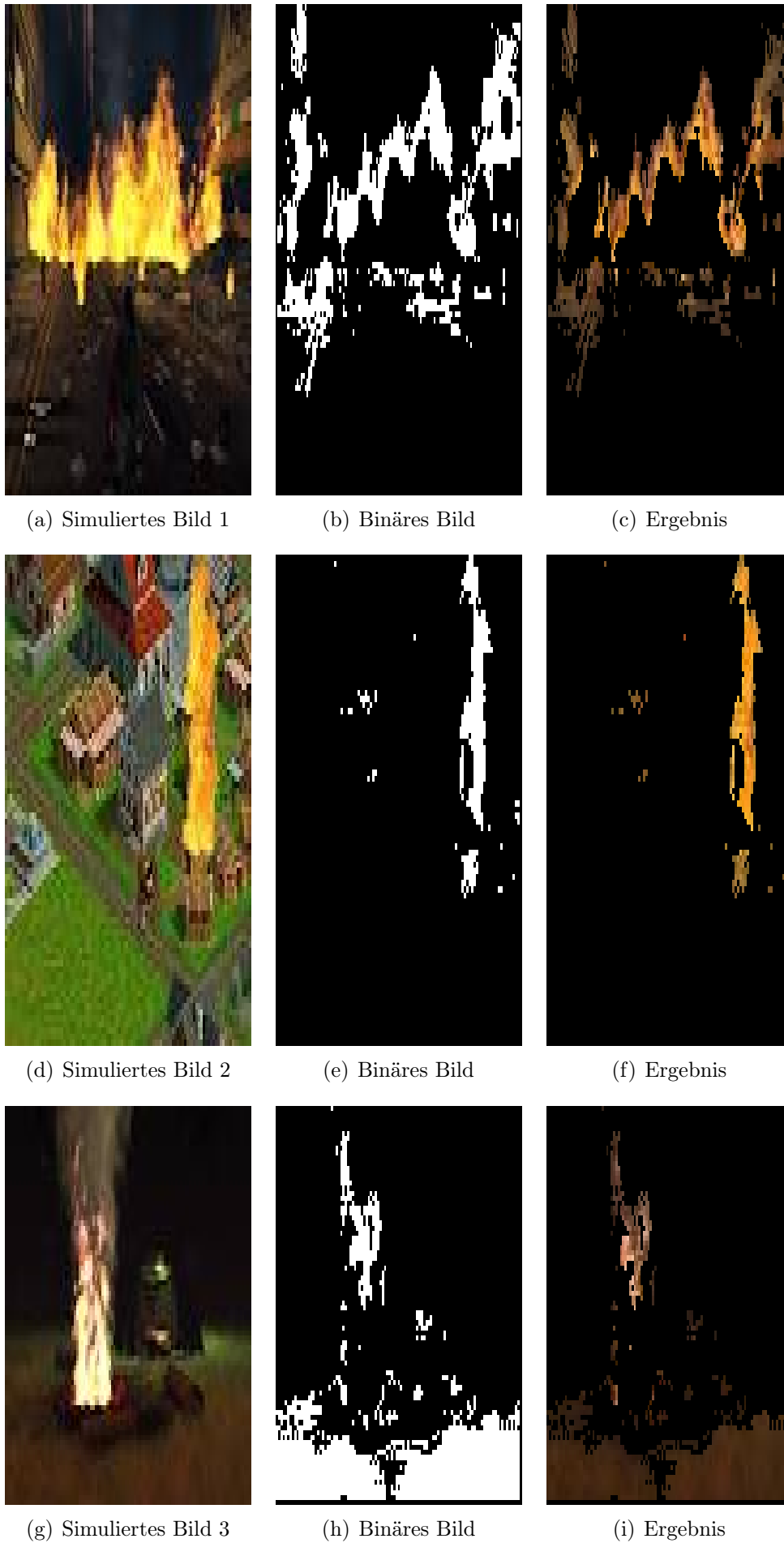
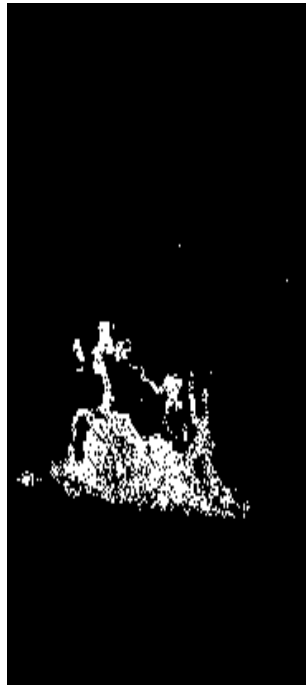


Bild 5.6: Simulierte Testbilder



(a) Verrauschtes Bild mit gauss'schem Rauschen



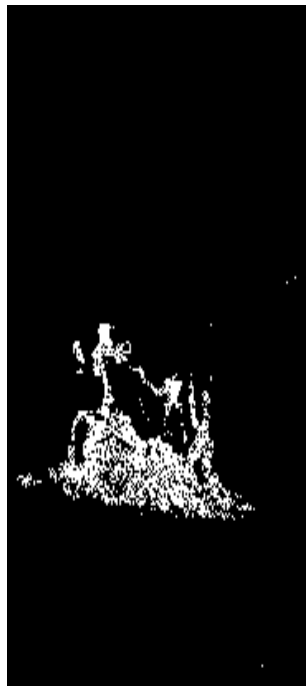
(b) Binäres Bild



(c) Ergebnis



(d) Verrauschtes Bild mit gleichverteiltem Rauschen



(e) Binäres Bild



(f) Ergebnis

Bild 5.7: Testbild mit Rauschen

5.1.2 Fazit

Durch die erzielten Ergebnisse wird festgestellt, dass die **bildbasierte Feuerdetektion** geeignet ist, wenn sich das UAV in einer Umgebung befindet, in der nur ein Waldbrand vorhanden ist.

Mit dem Test von den simulierten Bildern 5.6 wird bestätigt, dass das Verfahren in einer simulierten Umgebung wie **USARSim** sehr gute Ergebnisse liefern kann.

Ein Test mit verrauschten Bildern 5.7 hat gezeigt, dass das Verfahren bei der Verarbeitung von verrauschten Bildern optimale Ergebnisse liefern wird.

USARSim (*Unified System for Automation and Robot Simulation*) [drs09] ist eine simulierte interaktive Umgebung für Roboter und deren Sensoren [Hol07].

Ein UAV, mit einer eingebauten Kamera, kann während des Fluges Bilder aufnehmen. Mit der Hilfe des Detektionsverfahrens wird das UAV auf den aufgenommenen Bildern unabhängig der Wetterlage, der Bildauflösung und der Bildqualität Feuer erkennen können, wenn in den aufgenommenen Bildern Feuer vorhanden ist.

Sobald in den aufgenommenen Bildern jedoch andere Objekte mit feuerähnlichen Farben vorhanden sind, wird eine falsche Detektion erfolgen.

In den Abbildungen 5.3 und 5.4 ist erkennbar, dass das Verfahren anfällig wird, wenn die Objekte auf den aufgenommenen Bildern die Farben Rot, Gelb oder eine Mischung aus Rot-Grün haben.

Die Transformation von einem RGB-Bild in L-, a-, b-Bilder beträgt: 0.002 *Sekunde*

Für die gesamte Bearbeitung der Bilder benötigt das UAV eine CPU-Zeit von ungefähr 0.027 *Sekunde*.

5.2 Ergebnisse der videobasierten Feuererkennung

Dieser Abschnitt widmet sich dem Kapitel **Video basierte Feuererkennung**. Hier werden die Ergebnisse, die mit den Videos erzielt wurden, präsentiert. Das Feuer ist ein Element, das sehr instabil ist. Das bedeutet, dass sich das Feuerverhalten von Situation zu Situation ändern kann. Eine Begründung dafür liegt in dem Auslöser des Feuers.

Ist das Feuer im Video wegen einer Gasexplosion entstanden? Ist das Feuer durch das Anzünden eines Streichholzes entstanden? Oder ist das Feuer im Video ein Waldbrand? Diese Arbeit beschäftigt sich am meisten mit dem Waldbrand und es wurden mehrere Videos mit unterschiedlichen Inhalten getestet. Die meisten Videos wurden aus dem Internet heruntergeladen. Neun Videos, die Waldbrand enthalten, wurden von der Seite <http://signal.ee.bilkent.edu.tr/VisiFire> heruntergeladen.

Das Video 5.14(a) wurde von Frau Zhu Teng «tengzhu12@163.com» aus China zur Verfügung gestellt.

Für die Auswertung dieser Arbeit werden die erzielten Ergebnisse der Feuerdetektionsverfahren von ein paar Videos in Abbildung 5.8 gezeigt. Dabei werden die Ergebnisse von jedem einzelnen Schritt gezeigt.

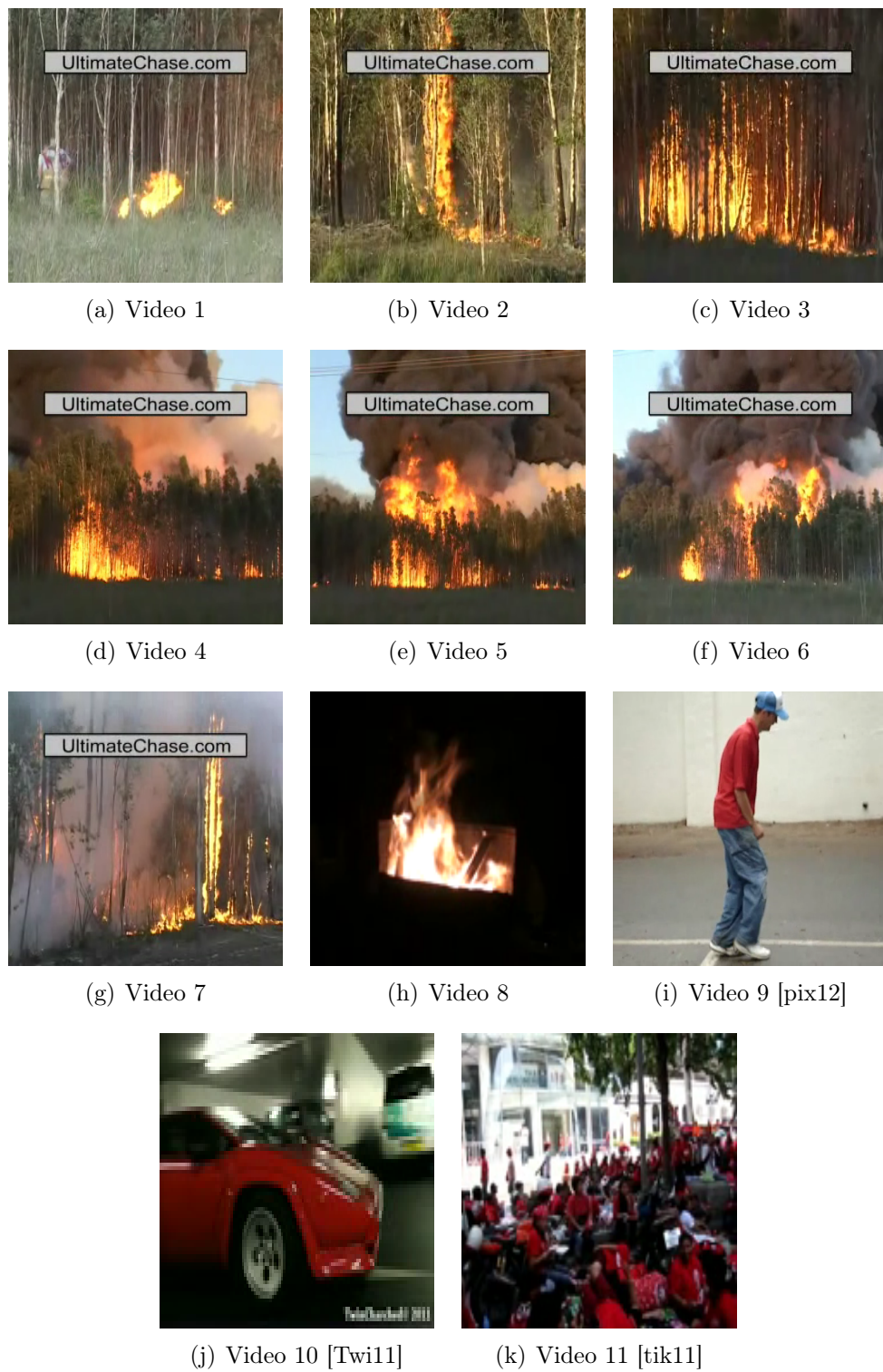


Bild 5.8: Testvideos mit Feuer und Feuerähnlichen Elementen

Das Verfahren **Detektion basierend auf Flammenkanten** besteht aus vier Schritten:

- Selektion von Feuer-Framekandidaten
- Selektion von Flammenregionen
- Kantendetektion
- Entscheidung

5.2.1 Test von Videos mit sicheren Feuerinhalten

Selektion von Feuer-Framekandidaten :

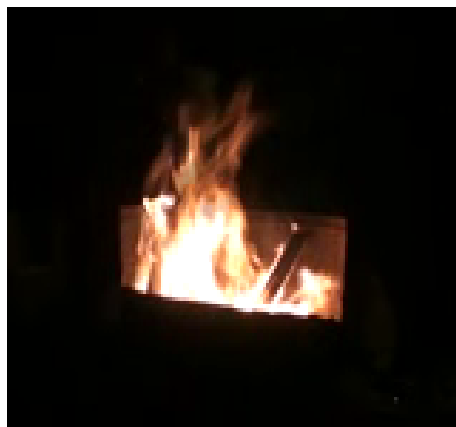
In dieser Etappe werden Feuer und feuerähnliche Elemente ausgewählt.



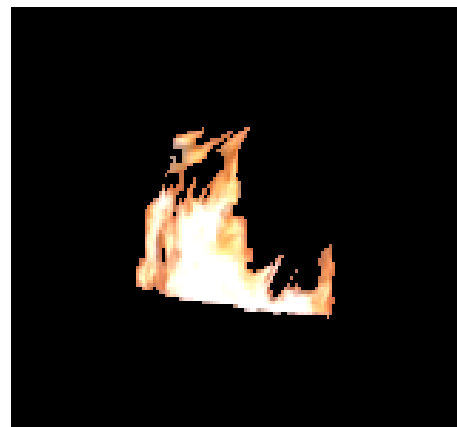
(a) Video 3



(b) Video 3 Framekandidaten



(c) Video 8



(d) Video 8 Framekandidaten

Bild 5.9: *Selektion von Feuer-Framekandidaten*

Hier werden alle Pixel, die die Gleichung (4.4) erfüllen, herausgefiltert. Es handelt sich

nicht nur um Pixel, die das Feuer tatsächlich repräsentieren. In der Abbildung 5.9(b) wird festgestellt, dass nur die Bäume des Videos 3 unterdrückt worden sind und die Pixel, die das Banner darstellen, wurden als Feuerpixel selektiert.

Die Abbildung 5.9(d) zeigt hingegen eine perfekte Selektion, weil das Video in dieser Sequenz, siehe Bild 5.9(c), keine weiteren Objekte als Feuer enthält.

Selektion von Flammenregionen :

In diesem Schritt wird entschieden, ob ein Pixel zu einer Feuerflamme gehört oder nicht.

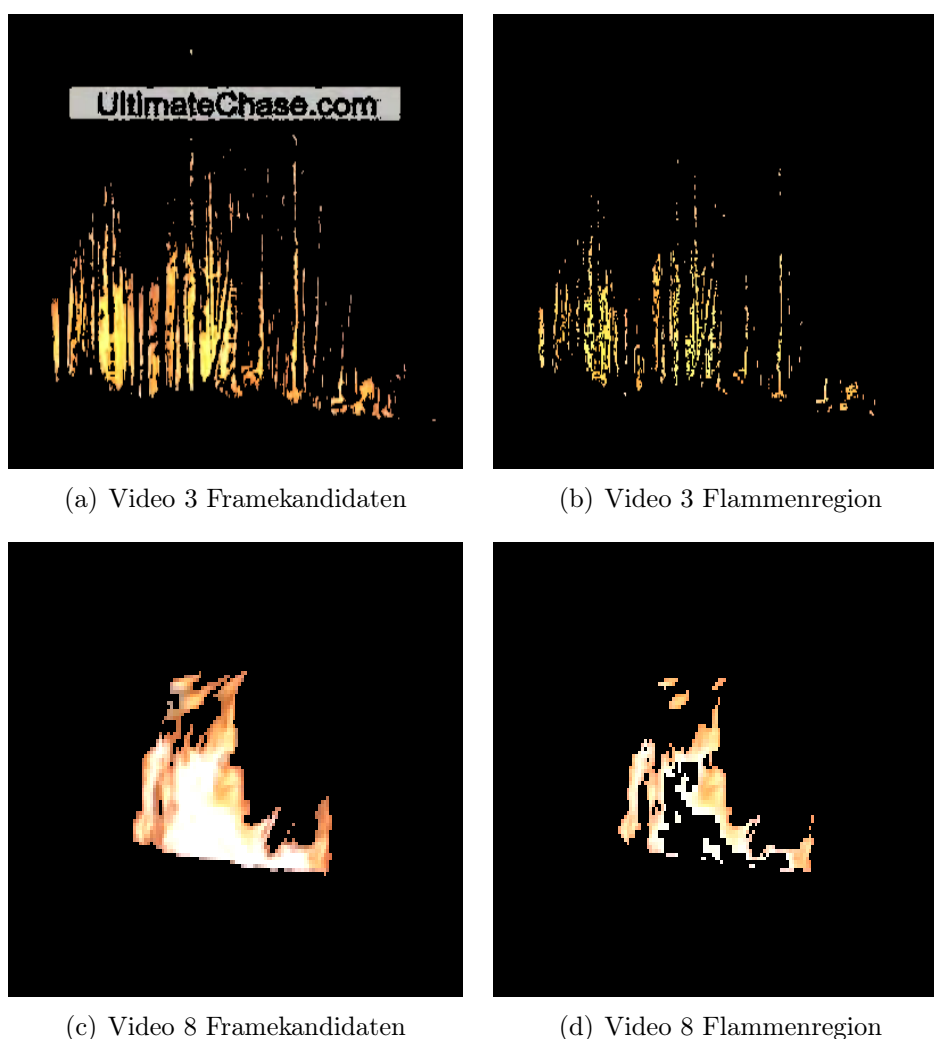


Bild 5.10: Selektion von Flammenregionen

Nach der Ausführung dieses Schrittes sind alle Regionen vorhanden, die das Feuer enthalten. Das Bild 5.10(a) enthält ein Banner und nach der Ausführung der Gleichung (4.5) verschwindet das Banner; siehe Bild 5.10(b).

Manche Methoden geben sich mit dem hier erreichten Ergebnis zufrieden, dennoch wird in dieser Arbeit ein Schritt weiter gegangen und eine Kantendetektion der Flammenre-

gionen wird durchgeführt.

Kantendetektion :

Die Detektion der Kanten wird mit der Hilfe des Canny-Verfahrens gemacht.

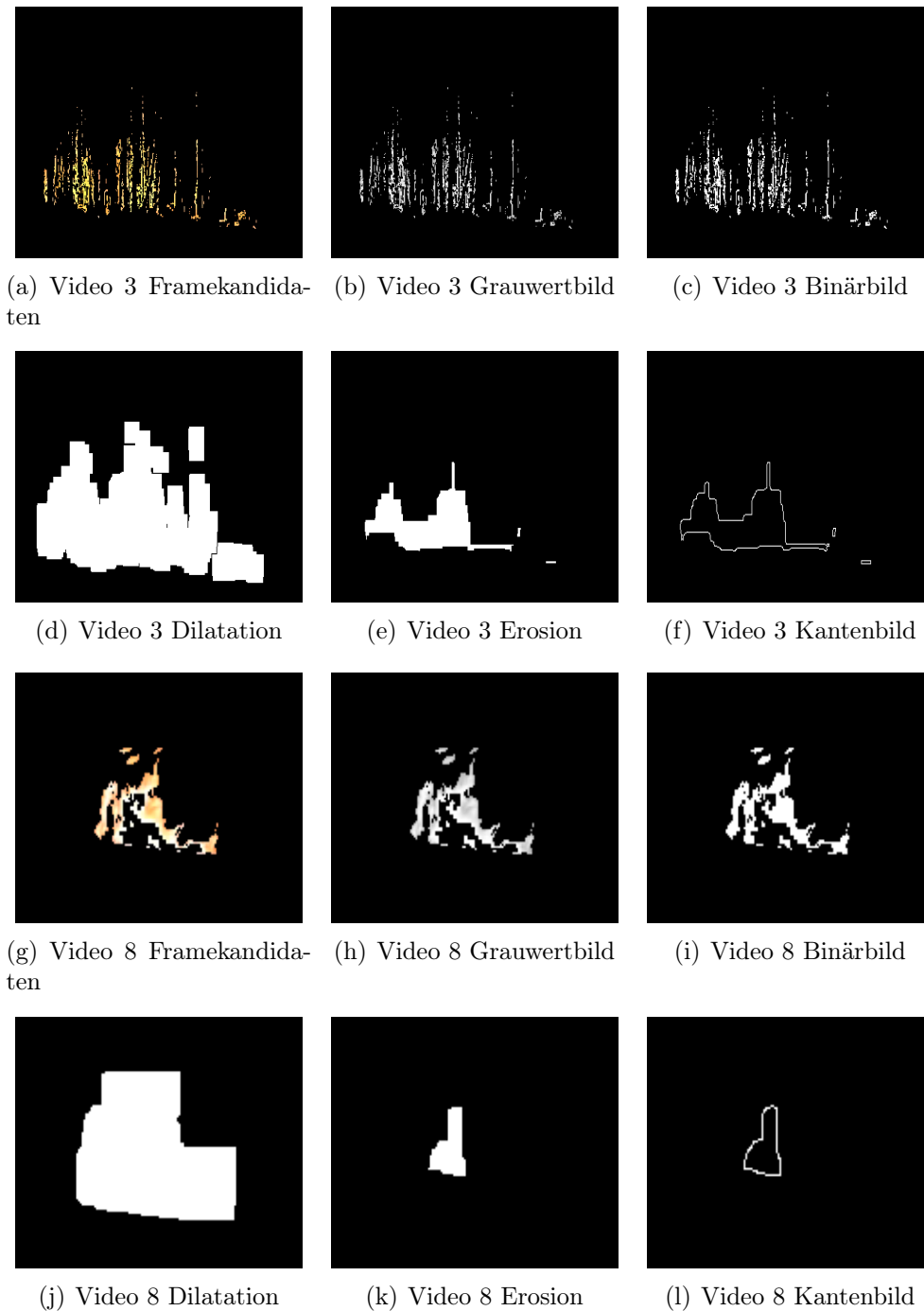


Bild 5.11: *Kantendetektion*

Dieser Schritt ist sehr wichtig für unser Verfahren, weil anhand der detektierten Kanten in Bild 5.11(f) die Entscheidung getroffen wird, ob Feuer in einem Video präsent ist.

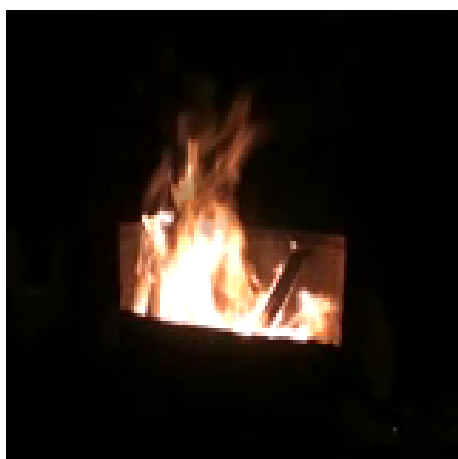
Bei dem Canny-Verfahren handelt es sich um ein Verfahren, das eines der besten Ergebnisse in dem Bereich der Kantenbearbeitung liefert.

Die Abbildung 5.17 zeigt, wie der Inhalt von einem Frame nach der Detektion der Feuerregion bis zum Kantenbild verändert wird.

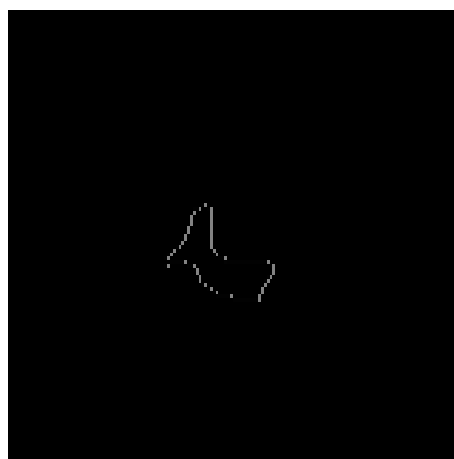
Entscheidung :

Nun werden die Alarmbilder angezeigt. Ein Alarmbild in dieser Arbeit ist das Kantenbild, das angezeigt wird, wenn eine der drei Entscheidungen 4.3.4.1, 4.3.4.2 und 4.3.4.3 der Detektionsverfahren eingetroffen ist.

Durch die Ergebnisse lässt sich sehen, dass das Feuer in dem gleichen Video unterschiedliches Verhalten aufweisen kann. Während in Video 3 nur zwei der drei Entscheidungen eintreffen, treffen bei Video 8 alle drei Entscheidungen ein.



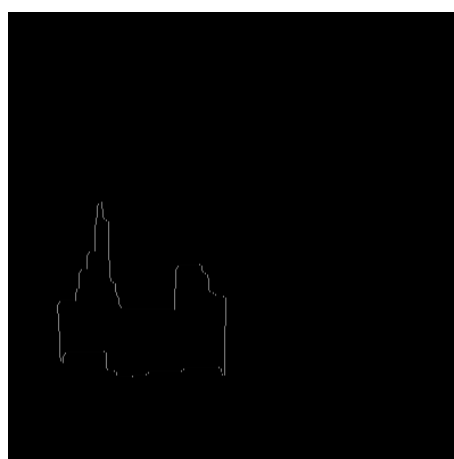
(a) Video 8



(b) Erstes Entscheidungsalarmbild für Video 8

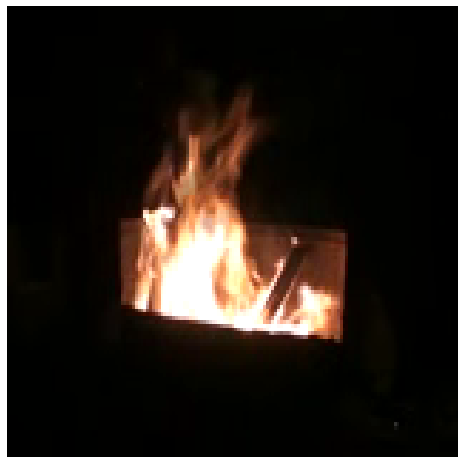


(c) Video 3



(d) Erstes Entscheidungsalarmbild für Video 3

Bild 5.12: Detektion, nachdem die erste Entscheidungsbedingung erfüllt wurde.



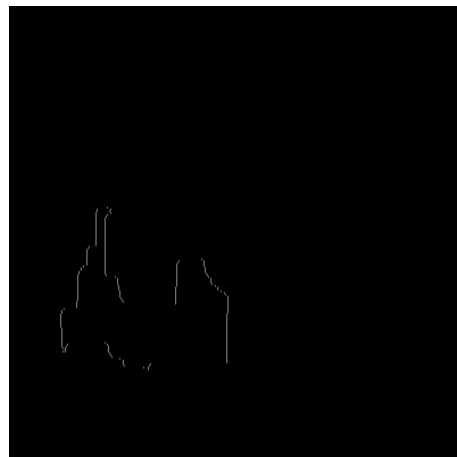
(a) Video 8



(b) Zweites Entscheidungsalarmbild für Video 8

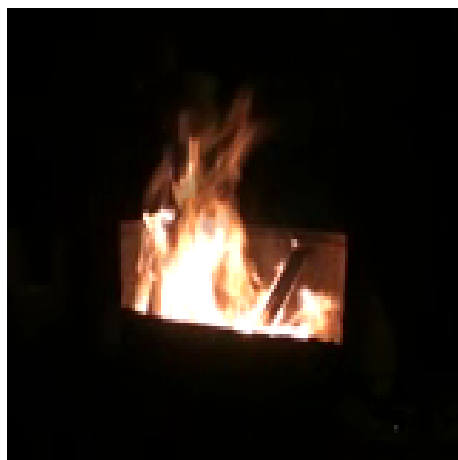


(c) Video 3

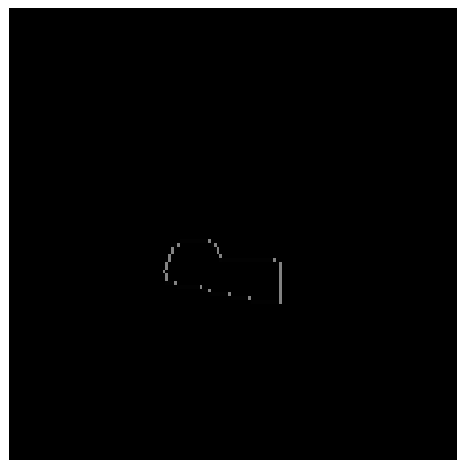


(d) Zweites Entscheidungsalarmbild für Video 3

Bild 5.13: Detektion, nachdem die zweite Entscheidungsbedingung erfüllt wurde.



(a) Video 8



(b) Drittes Entscheidungsalarmbild für Video 8

Bild 5.14: Detektion, nachdem die dritte Entscheidungsbedingung erfüllt wurde.

Die Abbildungen 5.12, 5.13, 5.14 zeigen die Ergebnisbilder. Die Anwendung von drei Entscheidungen zur Feuerdetektion verleiht dem Feuerdetektionsverfahren eine sehr gute Effizienz, weil zu unterschiedlichen Feuerverhalten eine Detektion des Feuers ermöglicht wird.

5.2.2 Test von Videos ohne Feuerinhalte

Es kann vorkommen, dass ein Video Objekte enthält, die sich wie Feuer verhalten. Solche Objekte erfüllen in manchen Sequenzen Feuerfarben-Bedingungen. Sobald diese Objekte Feuer-Kriterien erfüllen, werden sie von dem Detektionsverfahren ebenfalls als Feuer wahrgenommen.

In der Abbildung 5.15 sehen wir einen Mann mit rotem T-Shirt beim Tanzen und in dem Video 10 sehen wir ein rotes Auto in einem Tunnel. Nach der Selektion von Pixeln, die möglicherweise Feuerpixel sind, erkennt man, dass das Ergebnisbild nicht völlig dunkel ist. Das Ergebnisbild sollte jedoch dunkel sein, weil in den Videos kein Feuer enthalten ist.



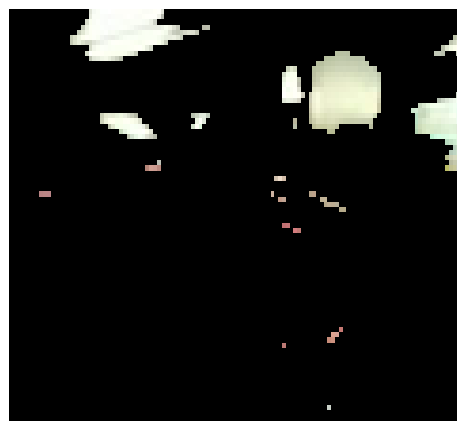
(a) Video 9



(b) Video 9 Framekandidaten



(c) Video 10

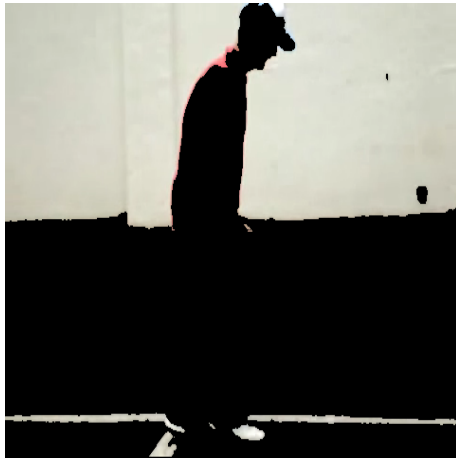


(d) Video 10 Framekandidaten

Bild 5.15: Selektion von Feuer-Framekandidaten in Video ohne Feuer

Nach der Selektion von Flammenregionen ist in dem Bild 5.16(b) zu sehen, dass es

komplett dunkel ist. Dies bedeutet, dass es kein Pixel gibt, das die Bedingung zur Flammenregion erfüllt. In dem Bild 5.16(d) wird erkannt, dass einige Pixel als Feuerregion erkannt worden sind.



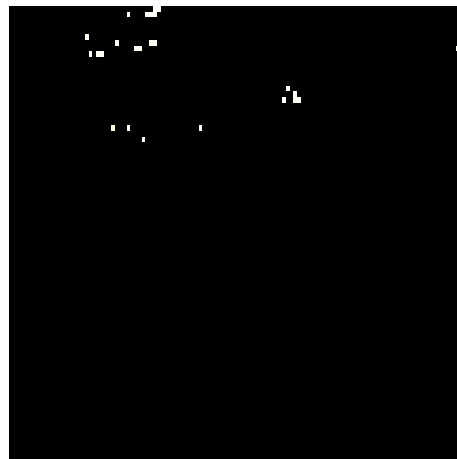
(a) Video 9 Framekandidaten



(b) Video 9 Flammenregion

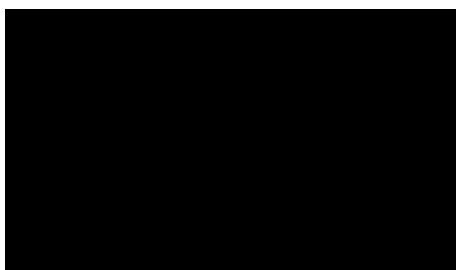


(c) Video 10 Framekandidaten

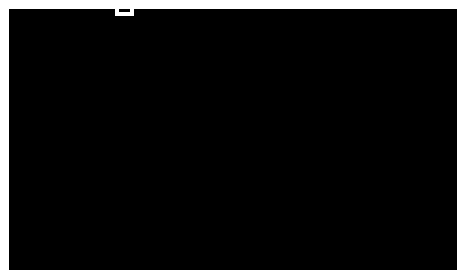


(d) Video 10 Flammenregion

Bild 5.16: *Selektion von Flammenregionen in Videos ohne Feuer*



(a) Video 9 Kantenbild



(b) Video 10 Kantenbild

Bild 5.17: *Kantendetektion*

Alle Pixel, die zu der Feuerregion durchgelassen wurden, werden weiter verarbeitet, damit eine Kantendetektion folgen kann. In der Abbildung 5.17 ist zu sehen, dass entweder

kein Kantenbild vorhanden ist oder das Kantenbild ist so unbedeutend, dass es keiner der drei Entscheidungen zur Feuererkennung entsprechen wird.

5.2.3 Gesamte Auswertung

Die Tabelle 5.1 enthält die getesteten Videos und deren Beschreibung.

Videos	Beschreibung
Video 1	Zwei Männer mit Feuer
Video 2	Waldbrand 1
Video 3	Waldbrand 2
Video 4	Waldbrand 3
Video 5	Waldbrand 4
Video 6	Waldbrand 5
Video 7	Waldbrand 7
Video 8	Frau mit feuerähnlichen Farben hinter Feuer
Video 9	Rotes Auto im Tunnel
Video 10	Rotes Auto auf offener Straße
Video 11	Tanzende Frau mit feuerähnlichen Elementen
Video 12	Tanzender Mann mit feuerähnlichem T-Shirt

Tabelle 5.1: Beschreibung von Testvideos

In der nächsten Tabelle (5.2) sind die wichtigsten Informationen zur Bewertung enthalten. Video 1 wurde während 6 Runden beobachtet. Dies bedeutet, dass 120 Frames beobachtet wurden, da 20 Frames pro Entscheidung analysiert werden. Der Eintrag *Alarm* definiert, wie oft das Alarmbild während einer Beobachtung ausgelöst wird. Das Alarmbild wird frühestens nach zwei Beobachtungen angezeigt.

Mit *Warnung* ist die Summe von Entscheidungen gemeint, die während einer gesamten Videoanalyse eingetroffen ist.

In den Spalten *Entscheidung 1*, *Entscheidung 2*, *Entscheidung 3* bezeichnet jeder Eintrag, wie oft eine bestimmte Entscheidung während einer Videoanalyse eingetroffen wurde.

In dem Video 1 werden nur 2 Warnungen und 2 Alarmen ausgelöst, obwohl das Feuer vorhanden ist. Dies lässt sich erstens durch die große Entfernung zwischen der Kamera und dem Feuer und zweitens durch die sehr schwache Brennstärke des Feuers erklären. In den Videos 2 bis 7 sind Waldbrände vorhanden. Die Detektion des Feuers in den Fällen ist perfekt. Die Detektion im Video 8 ist ebenfalls perfekt.

Die Videos 9 bis 12 sind Videos, die kein Feuer enthalten, aber die Farben von den darin enthaltenen Objekten sind feuerähnlich. In Video 9 werden drei Warnungen gegeben, aber kein Alarmbild ausgelöst. In Video 12 wird nach zwei Beobachtungen ein Alarmbild angezeigt, weil die Kamera sich bewegt hat und der tanzende Mann während der zweiten Beobachtung kurzfristig verdeckt wurde, sodass die Anzahl von Pixeln deutlich reduziert wurde. Die Videos 10 und 12 werden optimal analysiert.

Videos	Runden	Alarm	Warnung	Entscheidung 1	Entscheidung 2	Entscheidung 3
Video 1	6	2	2	0	1	1
Video 2	6	5	6	3	3	0
Video 3	5	4	5	3	2	0
Video 4	5	4	5	5	0	0
Video 5	6	5	6	5	1	0
Video 6	5	4	5	3	2	0
Video 7	5	4	5	1	4	0
Video 8	35	34	35	16	8	11
Video 9	105	3	3	3	0	0
Video 10	187	0	0	0	0	0
Video 11	23	0	0	0	0	0
Video 12	21	1	1	1	0	0

Tabelle 5.2: Leistung des Verfahrens auf Testvideos

5.2.4 Fazit

Auf Basis der mittels videobasierter Feuererkennung erlangten Ergebnisse kann behauptet werden, dass das hier gewählte und modifizierte Verfahren in einem UAV verwendet werden kann. Die Quote der falschen Detektionen befindet sich in der Tabelle 5.3. In der Tabelle wird ausgewertet, wie oft nach der Analyse eines Videos das Feuer nicht detektiert wird (Falsch Negativ) bzw. ein feuerähnliches Element als Feuer detektiert wird (Falsch Positiv).

Videos	Falsch Negativ	Falsch Positiv
Video 1	60.0%	0.0%
Video 2	0.0%	0.0%
Video 3	0.0%	0.0%
Video 4	0.0%	0.0%
Video 5	0.0%	0.0%
Video 6	0.0%	0.0%
Video 7	0.0%	0.0%
Video 8	0.0%	0.0%
Video 9	0.0%	0.0%
Video 10	0.0%	0.0%
Video 11	0.0%	0.0%
Video 12	0.0%	5.0%

Tabelle 5.3: Falsch Negativ und Falsch Positiv

Nachdem das UAV eine Flugroute zugewiesen bekommen hat, kann es mit Hilfe der Kamera Videos aufnehmen. Das Verfahren, das implementiert wurde, ermöglicht dem UAV eine Bearbeitung des aufgenommenen Videos in Echtzeit. Die erhaltenen Bildergebnisse zeigen, dass die durchgeführte Feuerdetektion innerhalb des UAVs immer sehr gut, effizient und optimal sein wird; und dies unabhängig vom Ort, in dem sich das UAV befindet.

Kapitel 6

Zusammenfassung und Ausblick

Abschließend wird noch einmal die zentralen Arbeitsschritte und Ergebnisse der vorliegenden Arbeit zusammengefasst. Zudem werden in einem Ausblick Anknüpfungspunkte für künftige Forschungsvorhaben und Ideen für eine Erweiterung dieser Arbeit benannt.

6.1 Zusammenfassung

Das Ziel, eine Methode zu implementieren, die einem UAV eine interne Bildverarbeitung zur Waldbranderkennung ermöglicht, ist erreicht.

In dem Kapitel **Grundlagen der Bildverarbeitung** sind wichtige Informationen gewonnen worden, um die später eingeführten Bildverarbeitungsverfahren, die von einem UAV ausgeführt werden, zu verstehen. In diesem Kapitel wurde eine generelle Definition des Bildes gegeben. Ebenso beschäftigt sich das Kapitel mit den Farbmodellen, zu denen ein Bild gehören kann. Für diese Arbeit waren tiefe Kenntnisse zu den Themen RGB-Farbmodell und Lab-Farbmodell erforderlich. Die digitale Bildverarbeitung wurde dazu verwendet, um Feuer in Videos und Bildern zu detektieren. Die Techniken der Segmentierung wurden ebenfalls in diesem Kapitel erläutert, weil mit diesen Feuerregionen von den anderen Bildregionen in Bildern und Videos unterschieden und herausgefiltert werden können. Eine Definition von Kanten wurde gegeben und deren unterschiedliche Formen erläutert. Die Kanten haben sich später als sehr wichtiger Bestandteil dieser Arbeit herausgestellt. Deshalb war es notwendig, die Verfahren zu erwähnen, die man zur Kantendetektion einsetzen kann. Im letzten Teil des Kapitels wird erklärt, welche Elemente notwendig sind, damit ein UAV autonom eine interne Bildverarbeitung durchführen kann.

Der Kapitel 3 beschäftigte sich mit einem Gassensor und die Hitzedetektion als Hilfsmitteln für die Feuerdetektion. Weil das UAV mit einer Kamera ausgestattet ist, wurde in diesem Kapitel nach einem Weg gesucht, der dem UAV ermöglicht in Bildern Feuer zu entdecken. Dafür wurde das Verfahren "**Fast and Efficient Method for Fire**

Detection Using Image Processing" von *Turgay Celik* [Cel10] verwendet. Mit diesem Verfahren kann das UAV aufgenommene Bilder, die meistens in RGB-Farbmodell sind, in Lab-Farbmodell konvertieren. Nach der Konvertierung werden die Bilder in L-Kanalbild, a-Kanalbild und b-Kanalbild umgewandelt, danach findet eine Binarisierung statt und zum Schluss wird die Feuerregion mit Hilfe der Segmentierung von den anderen Regionen selektiert. Die Feuerregionen sind dann in einem Ergebnisbild dargestellt. Dieses Bild wird in den gerade genannten Schritten verarbeitet und am Ende wird das detektierte Feuer angezeigt.

Eine weitere Möglichkeit mit der Kamera besteht darin, Feuer in einem laufenden Video zu detektieren. Im vierten Kapitel war die erste Idee, bewegte Objekte zu isolieren und nach einer bestimmten Farbanalyse zu entscheiden, ob man Feuer entdeckt hat oder nicht. Diese erste Idee hat sich als gut erwiesen, aber eine Fehldetektion lag häufig vor. Die zweite Idee basiert auf dem Verfahren **Early Fire Detection Based on Flame Contour in Video** [ZYW⁺10]. Hier wird entschieden, ob Feuer vorhanden ist, nachdem man die Kanten der detektierten Flammen analysiert hat. In dem Verfahren werden zuerst alle Pixel selektiert, die ein bestimmtes Feuerverhalten aufweisen, dann werden Flammenregionen gesucht und alle Regionen, die kein Feuer enthalten, werden unterdrückt. Um bestimmen zu können, ob es sich tatsächlich um Feuer handelt, werden mit Hilfe des Canny-Verfahrens deren Kanten detektiert. Die Kanten werden zuletzt analysiert, um zu entscheiden, ob das Feuer detektiert wurde.

In dem Abschnitt (4.4) wurde ein Vorschlag zur Simulation der Feuerdetektion in einer simulierten Umgebung erläutert.

Die Ergebnisse und die Auswertungen der auf Bildern und Videos basierenden Verfahren zur Feuerdetektion im fünften Kapitel haben gezeigt, dass man diese Verfahren sehr gut auf einem UAV einsetzen kann.

6.2 Ausblick

Bei der Durchführung dieser Arbeit wurde deutlich, dass die Qualität der Bildverarbeitungsergebnisse von dem Verfahren abhängt, aber auch von der Stabilität der Kamera und dem Abstand zwischen Kamera und Zielobjekten. Ebenso wurde festgestellt, dass Feuer unterschiedliche Verhaltensweisen aufzeigen kann.

Es kann vorkommen, dass ein bestimmtes Feuer trotz der Analyse der folgenden vier Eigenschaften (**Fläche, Umfang, Rundheit und Fluktuation**) nicht entdeckt wird, weil die Fläche und der Umfang des Feuers während einer bestimmten Zeit unverändert bleiben.

Basierend auf diesen Erkenntnissen, könnte eine Erweiterung meiner Arbeit sich mit folgenden Aspekten beschäftigen:

- Weitere Analyse des Feuers, um mehrere Eigenschaften zu erhalten, die zur Feuerdetektionsentscheidung verwendet werden können

-
- Einsatz von mehr als einer Kamera, um die Detektionswahrscheinlichkeit zu erhöhen. Dafür ist die Implementierung eines Verfahrens nötig, welches mehrere UAVs zur Feuerdetektion verwendet
 - Feuerdetektion in 3D-Videos

Literaturverzeichnis

- [AA99] B.W. Albers und A.K. Agrawal, “Schlieren analysis of an oscillating gas-jet diffusion flame”, *Combustio and flame*, Bd. 119, Nr. 1-2, S. 84–94, 1999.
- [Abm94] Wolfgang Abmayr, *Einführung in die digitale Bildverarbeitung*, B.G. Teubner Stuttgart, 1994.
- [Ans12] Stefan Anshl, “Schwelbrand”, <http://www.ansahl.com/Hab-und-Gut/Hausratversicherung/Hausrat-Lexikon/Schwelbrand.html>, Januar 2012.
- [Bae09] Baeckmann, “Lena”, <http://beackmann.de/img/lena.jpg>, November 2009.
- [Bov05] A.C. Bovik, *Handbook of image and video processing*, Academic Press, 2005.
- [Bru02] Ralf Bruder, “Digitale Bildverarbeitung (Image/Picture Processing)”, *Techn. Ber.*, Informatik, 2002.
- [Cel10] T. Celik, “Fast and Efficient Method for Fire Detection Using Image Processing”, *ETRI Journal*, Bd. 32, Nr. 6, December 2010.
- [Chr10] Christianvette, “Feuer”, <files.christianvette.de/fh/blender/feuer.png>, Oktober 2010.
- [Col05] ColorBasics, “tristimulis”, www.colorbasics.com/Tristimulus/, 2005.
- [Com09] Bionic Commando, “Geschichte des Feuers in PC-Spielen”, <http://www.pcgameshardware.de/aid,679054/Feuer-frei-Die-Geschichte-des-Feuers-in-PC-Spielen-2011er-Update-Spiele/Wissen/bildergalerie/?iid=1169948>, August 2009.
- [CtBC05] Tropical Savannas CRC und the Bushfire CRC, “Surface Fire”, learnline.cdu.edu.au/units/sbi263/resources/surface_fire.JPG, März 2005.
- [CWC04] T.H. Chen, P.H. Wu und Y.C. Chiou, “An early fire-detection method based on image processing”, *International Conference on Image Processing*, Bd. 3, Nr. 1707–1710, 2004.

- [drs09] drsteveb, "USARSim", <http://sourceforge.net/projects/usarsim>, 2009.
- [FB03] Dieter Kiesenbauer Fabian Bohnert, *Seminar Simulation und Bildanalyse mit Java*, Diplomarbeit, Universität Ulm, Juli 2003.
- [Hec02] Andreas Heck, "Farbmodell", www.beta45.de/farbcodes/theorie/farbmodelle.html, Januar 2002.
- [Hei06] Dominik Heider, *Drohnen im zivilen und militärischen Einsatz*, 2006.
- [Hol07] Dennis Holzhäuser, *Erweiterte Integration von USARSim in die Robbie-Architektur*, Diplomarbeit, Universität Koblenz . Landau, 2007.
- [HSL⁺93] G. Healey, D. Slater, T. Lin, B. Drda und A.D. Goedeke, "A System for Real-Time Fire Detection", *Proc. of IEEE Computer Vision and Pattern Recognition Conference*, S. 605–606, 1993.
- [Hub11] Fabian Hubert, "Drohne im Polizeiansatz", *Frankfurter Allgemeiner ZEITUNG FÜR DEUTSCHLAND*, 2011.
- [Iro12] Irotek, "Color conversion math and formulas", <http://www.easyrgb.com/index.php?X=MATH&H=07#text7>, Januar 2012.
- [Jim12] Jimbo, "Sonnenuntergang", <http://www.spanien-bilder.com/img672.htm>, Januar 2012.
- [Jog05] Markus Jogmin, *Kantenerkennung in medizinischen Bildern*, 2005.
- [Kid12] Science Kids, "Forest Fire", <http://www.sciencekids.co.nz/pictures/disasters/forestfire.html>, 2012.
- [KL11a] Hans Lackner Klaus Lipinski, "image processing", www.wissen.info, August 2011.
- [KL11b] Hans Lackner Klaus Lipinski, "Video in Bildverarbeitung", www.itwissen.info/definition/lexikon/Bildverarbeitung-image-processing.html, August 2011.
- [Kos04] Jana Kosecka, "Vision-based Landinf of an Unmanned Air Vehicle", *Techn. Ber.*, Department of Computer Science, 2004.
- [Kre09] Holger Kreissl, "INFORMATIK TUTORIALS", www.kreissl.info/bildererkennung.php, Oktober 2009.
- [LK03] Z. Liu und A.K. Kim, "Review of recent developments in fire detection in technologies", *Journal of Fire Protection Engineering*, Bd. 13, Nr. 2, S. 129–151, 2003.
- [LOPR97] T. Lehmann, W. Oberschelp, E. Pelikan und R. Repges, *Bildverarbeitung fuer die Medizin*, Springer Berlin, 1997.

- [Mea94] Brian J. Meacham, "International developments in Fire sensor Technology", *Journal of Fire Protection Engineering*, Bd. 6, Nr. 2, S. 89, 1994.
- [Neu09] B Neumann, "Grundlagen Bildverarbeitung", www3.fh-swf.de/fbin/neumann_b/download/Grundlagen_Bildverarbeitung_Kap_9_Vers_2.pdf, April 2009.
- [pix12] pixieMoonGurl, "Dancing guy", <http://www.youtube.com/watch?v=ef-0GLGQoPc>, Januar 2012.
- [P.P09] P.P.A, "HolyRomanEmpire", blog-imgs-34-origin.fc2.com/h/o/1/holyromanempire/anno-feuer.png, Oktober 2009.
- [PRB12] Marco L. Zehner Prof. Ralf Bill, "Farbbild", <http://www.geoinformatik.uni-rostock.de/einzel.asp?ID=653337650>, 2012.
- [Rei08] Ute Reinelt, "Farbe in Lab", www.ute-reinelt.de/bildbearbeitung.htm, Oktober 2008.
- [Sav11] Peter Saveliev, "Math. Images. Software", <http://inperc.com/wiki/index.php?title=Area>, Oktober 2011.
- [Sec12] "Schnelle Feuererkennung", www.securiton.de/system/document/attachments/de/252/original/Referenzbericht_Markusbiert-Tunnel-Luxemburg_web.pdf, Januar 2012.
- [Ste93] R. Steinbrecher, *Bildverarbeitung in der Praxis*, R. Oldenburg Verlag, 1993.
- [Str06] Carsten Strampe, "Bildverarbeitung in Echtzeit", *PRAXIS PROFILINE*, November 2006.
- [TDGC05] B.U. Toreyin, Y. Dedeoglu, U. Gudukbay und A.E. Cetin, "Computer vision based method for real-time fire and flame detection", *Pattern Recognition Letters*, Bd. 27, Nr. 1, S. 49–58, 2005.
- [the09] theprogrammer12, "OpenCV mit Qt nutzen", www.qtforum.de/forum/viewtopic.php?t=9721, August 2009.
- [tik11] tik4u007, "red-shirt-bangkok", http://www.youtube.com/watch?v=sfy_3sBPmos&feature=related, 2011.
- [TKK10] Z. Teng, J.H. Kim und D.J. Kang, "Fire detection based on hidden Markov models", *International Journal of Control, Automation and Systems*, Bd. 8, Nr. 4, S. 822–830, 2010.
- [Twi11] TwinCharched, "lamborghini", <http://www.youtube.com/watch?v=qDAJch5swpg&feature=fvsrc>, 2011.
- [WW09] WSB-Webservice, "Terrasse", www.fewo-buchholz.de/_pics/93/gemuettliches_Beisammensein_am_Feuer.png, Oktober 2009.

- [YX11] W. Yuanbin und M. Xianmin, “Fire Detection Based on Image Processing in Coal Mine”, in *Internet Computing & Information Service (ICICIS)*, S. 431–434, IEEE, 2011.
- [ZYW⁺10] X.L. Zhou, F.X. Yu, Y.C. Wen, Z.M. Lu und G.H. Song, “Early fire detection based on flame contours in video”, *Information Technology Journal*, Bd. 9, Nr. 5, S. 899–908, 2010.