

Hardware-Abbildung eines videobasierten Verfahrens zur echtzeitfähigen Auswertung von Winkelhistogrammen auf eine modulare Coprozessor-Architektur

H. Flatt, A. Tarnowsky, H. Blume, and P. Pirsch

Leibniz Universität Hannover, Institut für Mikroelektronische Systeme, Appelstr. 4, 30167 Hannover, Deutschland

Zusammenfassung. Dieser Beitrag behandelt die Abbildung eines videobasierten Verfahrens zur echtzeitfähigen Auswertung von Winkelhistogrammen auf eine modulare Coprozessor-Architektur. Die Architektur besteht aus mehreren dedizierten Recheneinheiten zur parallelen Verarbeitung rechenintensiver Bildverarbeitungsverfahren und ist mit einem RISC-Prozessor verbunden. Eine konfigurierbare Architekturerweiterung um eine Recheneinheit zur Auswertung von Winkelhistogrammen von Objekten ermöglicht in Verbindung mit dem RISC eine echtzeitfähige Klassifikation. Je nach Konfiguration sind für die Architekturerweiterung auf einem Xilinx Virtex-5-FPGA zwischen 3300 und 12000 Lookup-Tables erforderlich. Bei einer Taktfrequenz von 100 MHz können unabhängig von der Bildauflösung pro Einzelbild in einem 25-Hz-Videodatenstrom bis zu 100 Objekte der Größe 256×256 Pixel analysiert werden.

Abstract. (English version) This paper presents the mapping of a video-based approach for real-time evaluation of angular histograms on a modular coprocessor architecture. The architecture comprises several dedicated processing elements for parallel processing of computation-intensive image processing tasks and is coupled with a RISC processor. A configurable architecture extension, especially a processing element for evaluating angular histograms of objects in conjunction with a RISC processor, provides a real-time classification. Depending on the configuration of the architecture extension, 3300 to 12000 look-up tables are required for a Xilinx Virtex-5 FPGA implementation. Running at a clock frequency of 100 MHz and independently of the image resolution per frame, 100 objects of size 256×256 pixels are analyzed in a 25 Hz video stream by the architecture.

1 Einleitung

In den vergangenen Jahren haben die Anforderungen an intelligente Bild- und Videoverarbeitungsverfahren kontinuierlich zugenommen. Die fortschreitende Technologie ermöglicht es, komplexe Algorithmen auf eingebettete Systeme abzubilden. Insbesondere an mobile Systeme werden hohe Anforderungen gestellt, wenn die Datenverarbeitung in Echtzeit erforderlich ist.

Ein vielseitiges Teilgebiet der Bildverarbeitung ist die Objekterkennung. Anwendungen existieren in verschiedensten Bereichen. So kann beispielsweise die Erkennung von Fahrbahnrandern im Kfz die Grundlage von Fahrerassistenzsystemen bilden (Schönfeld, 1994; Yu et al., 2008). Im Bereich der Luftbilder unterstützen automatische Verfahren das Auffinden besonderer Objekte wie z.B. Flughäfen (Liu et al., 2004), Minen (Messelink et al., 2002) oder Fahrzeugen (Greenberg et al., 2005). Das Bildmaterial stammt sowohl von optischen, als auch von thermischen und Radar-Sensoren.

Alle diese Anwendungsbeispiele basieren auf einer automatisierten Objekterkennung und -klassifikation. Die ihnen zugrunde liegenden Verfahren sind meist sehr rechenaufwändig und benötigen entweder sehr schnelle General Purpose Prozessoren, digitale Signalprozessoren oder dedizierte Hardware-Einheiten. Insbesondere in der mobilen Anwendung ist erstere Lösung aufgrund der hohen Leistungsaufnahme derartiger Prozessoren jedoch häufig nicht anwendbar. Da die Objekterkennung meist aus mehreren Algorithmen besteht, ist auch eine hohe Flexibilität notwendig, die eine reine Hardware-Lösung nicht immer bieten kann. Als geeignete Alternative bieten sich hybride RISC/Coprozessor-Systeme an. Der RISC-Prozessor bietet die erforderliche Flexibilität, während der Coprozessor spezielle dedizierte Recheneinheiten zur Prozessierung zeitkritischer Algorithmen enthält.



Correspondence to: H. Flatt
(flatt@ims.uni-hannover.de)

Die in Flatt et al. (2009a) dargestellte modulare Coprozessor-Architektur eignet sich für Objekterkennungsapplikationen, wie die Erkennung von Fahrzeugen auf geraden Autobahnabschnitten in Luftbildern. Der Coprozessor ermittelt Objektkandidaten basierend auf Basisoperationen der Bildverarbeitung und berechnet für diese Merkmale, wie Objektgröße, Position und Exzentrizität. Anhand dieser Merkmale entscheidet der RISC-Prozessor, ob es sich um ein Fahrzeug handelt oder nicht. Diese Merkmale allein sind nicht ausreichend für eine zuverlässige Fahrzeugerkennung. Speziell Fahrbahnmarkierungen und Schatten auf der Fahrbahnoberfläche stellen eine Herausforderung an das Objekterkennungsverfahren dar. Um die Erkennungsleistung zu optimieren, ist es erforderlich, aufwändigere Analyseverfahren zu verwenden, die aber lediglich von sehr komplexen Hardware-Architekturen in Echtzeit verarbeitet werden können. Für diese Aufgabe gibt es bereits eine Vielzahl unterschiedlicher Ansätze, die sich durch diverse Vor- und Nachteile bezüglich ihrer Flexibilität, Geschwindigkeit und ihres erforderlichen Hardware-Aufwandes auszeichnen.

Eine sehr hohe Erkennungsleistung bieten Verfahren, wie Scale Invariant Feature Transform (SIFT) (Lowe, 2004) und Histogram of Orientated Gradients (HOG) (Dalal et al., 2006), welche jedoch einen vergleichsweise hohen Rechenaufwand erfordern.

Die sogenannten Template-Matching-Verfahren weisen eine deutlich geringere algorithmische Komplexität auf, da sie das zu analysierende Objekt lediglich mit einer Anzahl von Referenzbildern vergleichen. Der Vergleichswert bildet ein Ähnlichkeitsmaß bezüglich zweier Bilder. Der Aufwand für eine Hardware-Implementierung ist gering (Flatt et al., 2007), jedoch ist eine große Menge an Vergleichsdaten notwendig, um eine akzeptable Erkennungsleistung zu erhalten. Daher ist auch hier ein hoher Gesamtrechenaufwand erforderlich.

Das Verfahren von Korn (1998) vereint einige Vorteile der oben genannten Verfahren. Obwohl algorithmische Komplexität und Hardware-Implementierungsaufwand gering sind, ist der Algorithmus, ähnlich dem SIFT-Verfahren, skalierungs- und rotationsinvariant. Der Ansatz des Verfahrens besteht darin, Richtungsinformationen innerhalb des Quellbildes statistisch zu analysieren und anhand von gegebenen Referenzdaten eine Klassifikation vorzunehmen. Dies ermöglicht beispielsweise in der Fahrzeugerkennung aus Luftbildern die Unterscheidung zwischen PKW, LKW und Störobjekten.

Dieser Beitrag zeigt die Hardware-Umsetzung eines auf dem Ansatz von Korn (1998) basierenden Verfahrens. Die Berechnungen erfolgen ausschließlich auf der Basis von Festpunktarithmetik. Der Vergleichsaufwand mit den Referenzbildern wird durch Vorwissen über die Objektform reduziert. Der Schwerpunkt dieses Beitrages liegt in der Hardware-Umsetzung mit dem Ziel, eine Vielzahl von Regions-of-Interest in Videosequenzen schritt haltend analysieren zu können.

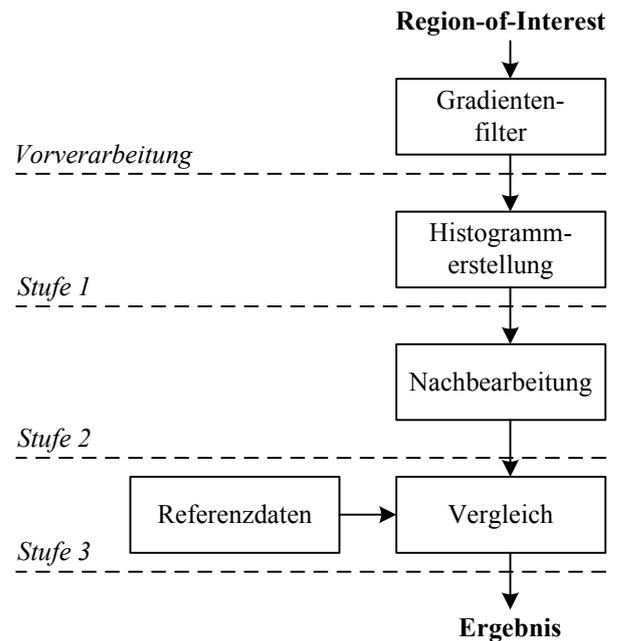


Abbildung 1. Ablauf des Verfahrens zur Winkelauswertung

Der Beitrag gliedert sich wie folgt: In Abschnitt 2 wird das Verfahren zur Auswertung von Winkelverteilungsräumen präsentiert. Abschnitt 3 behandelt Optimierungen des Verfahrens, welche eine effiziente Umsetzung in Hardware begünstigen. Abschnitt 4 zeigt die Hardware-Implementierung des Verfahrens. In Abschnitt 5 werden die Ergebnisse der Hardware-Implementierung diskutiert. Die Zusammenfassung des Beitrages erfolgt in Abschnitt 6.

2 Auswertung von Winkelhistogrammen

Das Verfahren nach Korn lässt sich, wie in Abb. 1 dargestellt, in drei Stufen unterteilen. Als Eingangsdaten des Verfahrens dienen ein horizontales und ein vertikales Grauwertgradientenbild sowie die Koordinaten einer Region-of-Interest (ROI) im Bild, welche im Rahmen der Vorverarbeitung berechnet werden.

Die ersten beiden Stufen dienen der Transformation der Eingangsdaten in den Merkmalsraum. Da die Daten der direkten Umwandlung (Stufe 1) jedoch meist mit Störungen behaftet sind, werden sie geglättet und normiert (Stufe 2). Nach diesen beiden Schritten können die Daten mit denen anderer Referenzbilder verglichen werden (Stufe 3). Die Referenzdaten sind ebenfalls durch Anwendung der ersten beiden Stufen im Voraus erstellt worden und in einer Listenstruktur gespeichert. Als Ergebnis werden der Index sowie der Vergleichswert des Referenzhistogramms mit dem niedrigsten Vergleichswert ausgegeben. Im Folgenden werden die einzelnen Verarbeitungsschritte detailliert dargestellt.

2.1 Gradientenfilter

Um aus den Grauwerten der ROI die für die Transformation nötigen Richtungsdaten extrahieren zu können, müssen zunächst die Gradientendaten ermittelt werden. Die Gradientenberechnung kann durch ein geeignetes Faltungsfiler angenähert werden. Dieses muss sowohl in vertikaler als auch horizontaler Richtung ausgeführt werden. Hier bietet sich im Wesentlichen ein optimiertes Sobelfilter an (Scharr, 2000). Da das Faltungsfiler eine Differenzierung annähert, bedeutet dies rückwirkend auf die zu analysierende ROI, dass globale Helligkeitsänderungen keinen Einfluss auf das Ergebnis des Verfahrens haben.

2.2 Histogrammerstellung

Die in der Vorverarbeitung berechneten horizontalen und vertikalen Gradienten müssen zunächst in eine polare Darstellung, bestehend aus Betrag r und Winkel φ , überführt werden. Die Gradienten lassen sich wie folgt in einen Betrags- und Winkelwert r und φ überführen, der die Normale für diesen Punkt beschreibt (Jähne, 2005):

$$r = \sqrt{g_x^2 + g_y^2} \quad (1)$$

$$\tan \varphi = \frac{g_y}{g_x} \Leftrightarrow \varphi = \arctan \frac{g_y}{g_x} \quad (2)$$

Um alle vier Quadranten abzudecken, müssen zusätzlich die Vorzeichen von g_x und g_y bei der Winkelberechnung mit berücksichtigt werden. Abschließend müssen die berechneten Winkelwerte, welche einen Wertebereich von $-\pi$ bis π aufweisen, auf die Breite des Histogramms angepasst werden. Das Histogramm besteht aus n Einträgen, die jeweils einen Winkelbereich von $\frac{2\pi}{n}$ zusammenfassen. Die Histogrammadressen $\tilde{\varphi}$ berechnen sich entsprechend Gleichung (3).

$$\tilde{\varphi} = \left[\frac{n}{2\pi} \right] \quad (3)$$

Das ungeglättete Winkelhistogramm h_u berechnet sich für m Gradientenvektoren zu:

$$h_{u,i} = \sum_{j=0}^{m-1} r_j |(\tilde{\varphi}_j = i) \quad (4)$$

Da die Breite des Histogramms den Berechnungsaufwand für den Vergleichsschritt maßgeblich bestimmt, ist sie so klein wie möglich zu wählen. Für die Zielapplikation der Luftbild-basierten Fahrzeugerkennung wurde eine Breite von 128 Histogrammeinträgen gewählt, da diese die Erkennungsleistung des Verfahrens nur marginal beeinflusst (Tarnowsky, 2009).

2.3 Nachbearbeitung

Bevor der eigentliche Vergleich der Winkelverteilungsräume erfolgen kann, müssen die vorliegenden Daten zunächst aufbereitet werden. Das im vorherigen Schritt berechnete Winkelhistogramm ist meist mit Störungen behaftet. Aus diesem Grund werden die vorliegenden Daten durch ein geeignetes Filter geglättet.

Es bietet sich an, diese Glättung durch ein FIR-Filter zu realisieren, welches eine Tiefpassfilterung in Form eines Gauß-Filters abbildet. Eine konkrete Beschreibung des eindimensionalen Filterkerns f lässt sich an dieser Stelle noch nicht geben, da dieser von der Implementierung des Histogramms abhängt. Für das geglättete Histogramm h_g gilt dann:

$$h_g = h_u * f \quad (5)$$

Die nach der Glättung vorliegenden Daten haben einen Wertebereich, der sowohl von der Größe des Eingangsbildes, als auch von dessen Inhalt abhängt. Um eine Invarianz bezüglich der Skalierung der Eingangsdaten zu erhalten, werden die Werte des Histogramms in einem zusätzlichen Schritt normiert. Diese Invarianz ist insbesondere deswegen von Vorteil, da sie die Anzahl der benötigten Referenzdaten reduziert und für verschiedene Objektgrößen ein einziger Referenzdatensatz verwendet werden kann. Die Normierung erfolgt, indem sämtliche Werte des Histogramms durch den größten Wert dividiert werden.

2.4 Vergleich

Im dritten Verarbeitungsschritt wird das Winkelhistogramm des zu analysierenden Objektes mit denen anderer Referenzbilder verglichen. Diese wurden vorher durch Anwendung der ersten beiden Schritte erstellt (vgl. Abb. 1).

Ziel ist es, das Referenzhistogramm zu finden, welches die größte Ähnlichkeit zu dem aktuell ermittelten hat. Es existieren verschiedene Ähnlichkeitsmaße, die dieses leisten können. Hier soll das Vergleichsmaß Sum of Absolute Differences (SAD) verwendet werden, da es sich sehr gut für eine Hardware-Umsetzung eignet. Um eine Rotationsinvarianz zu ermöglichen, wird das zu vergleichende Histogramm um einen Winkelschritt rotiert und erneut verglichen. Dies wird solange wiederholt, bis sämtliche Winkelkombinationen verglichen wurden. Das Minimum über alle so ermittelten Differenzsummen bildet das Endergebnis des Algorithmus.

Der SAD-Vergleich des geglätteten Histogramms h_g mit einer einzelnen Referenz r_j wird wie folgt beschrieben:

$$d_{j,\min} = \min_{\varphi \in [0, n-1]} \sum_{i=0}^{n-1} |h_{g,(i+\varphi) \bmod n} - r_{j,i}| \quad (6)$$

Zusätzlich zum Minimalwert aller Vergleichsergebnisse d_j wird auch der Index j des gefundenen

Referenzhistogramms zurückgegeben. Da die Klassenzugehörigkeiten der Referenzhistogramme j a priori bekannt sind, kann mit Hilfe des Index j die Klassifikation des Objektes vorgenommen werden.

2.5 Bewertung des Verfahrens

Das Verfahren eignet sich vor allem für die zuverlässige Erkennung rechteckiger Formen (Tarnowsky, 2009). Das vorgestellte Verfahren unterliegt den folgenden natürlichen Grenzen:

Da bei der Transformation der Gradientendaten in den Winkelverteilungsraum sämtliche Positionsinformationen verloren gehen, ist es nicht möglich, zwischen Objekten zu unterscheiden, die eine ähnliche Winkelverteilung besitzen, optisch sich jedoch stark unterscheiden.

Ein weiterer Nachteil des Verfahrens ist die Erkennung runder Formen. Da das resultierende Histogramm eines kreisförmigen Objektes theoretisch gleichverteilt ist, die Gradientenfilterung auf einem diskreten Gitter jedoch nicht exakt ist (Jähne, 2005), kann es kaum von einem z. B. stark verrauschten Bild unterschieden werden. Diese Einschränkung ist jedoch nicht gravierend, da in dem vorgesehenen Umfeld der Fahrzeugerkennung vorwiegend rechteckige Strukturen dominieren.

3 Optimierung des Verfahrens

Das Verfahren hat in der dargestellten Form Eigenschaften, die sich unvorteilhaft im Rahmen einer Hardware-Implementierung auswirken. Neben der noch nicht festgelegten Diskretisierung des Histogramms ist auch die Verwendung von Fließkommaarithmetik sehr aufwändig. Dieses schließt auch die bisherige Implementierung der Funktion $\arctan\left(\frac{g_y}{g_x}\right)$ mit ein. Die im Folgenden dargelegten Modifikationen sind die Voraussetzung der in Abschnitt 4 präsentierten echtzeitfähigen Hardware-Architektur.

3.1 Festkommadarstellung

Für die Gradientendaten, welche als Eingangswerte des Verfahrens dienen, wird zugunsten einer effizienten Hardware-Umsetzung eine Ganzzahldarstellung vorausgesetzt. Ziel ist es, die Inkrementierungslogik des Histogramms ebenfalls auf Ganzzahlen zu beschränken. Lediglich bei der Normierung der Histogrammeinträge muss auf eine Festkommadarstellung ausgewichen werden. Diese wurde derart gewählt, dass der Wertebereich des Histogramms auf die Werte 0 bis 255 abgebildet wird. Die Genauigkeit dieser 8-Bit-Darstellung zeigt sich als ausreichend, kann aber bei Bedarf auch erhöht werden (Tarnowsky, 2009).

3.2 Gradientenapproximation

Die in Gleichung (2) dargestellte Winkelberechnung, welche auf der arctan-Funktion basiert, soll auf 128 Histogrammeinträge abgebildet werden und erfordert daher nur eine geringe Genauigkeit. Wird die Symmetrie bei der Winkelberechnung berücksichtigt, ist eine Approximation der Funktion $\arctan\left(\frac{g_y}{g_x}\right)$ für den ersten Quadranten ausreichend. Somit besteht der Wertebereich der approximierten Funktion aus nur 5 Bit. Die arctan-Funktion kann auf einen festen Definitionsbereich reduziert werden, indem g_y und g_x durch eine geeignete Zweierpotenz dividiert und gerundet werden. Resultierend kann die Winkelberechnung als kompakte Funktionsstabelle realisiert werden.

Die in Gleichung (1) dargestellte Betragsfunktion würde in Hardware aufwändige Multiplikations- und Wurzelberechnungen erfordern. Daher wird die Betragsfunktion durch die L1-Norm wie folgt approximiert:

$$\tilde{r} = |g_x| + |g_y| \quad (7)$$

3.3 Histogrammvergleiche

Ausgehend von Abschnitt 2.4 müssen für jedes Referenzhistogramm alle 128 möglichen Verschiebungswerte analysiert und das Minimum über deren Differenzsummen ermittelt werden, damit die Rotationsinvarianz gewährleistet ist. Dieser Ansatz erfordert somit $128 \cdot 128 = 16384$ Subtraktionen mit anschließender Akkumulation (SAD-Operationen). Bei Betrachtung der durchschnittlichen Winkelverteilungen rechteckiger Objekte zeigt sich, dass diese mindestens zwei ausgeprägte Maxima aufweisen. Diese repräsentieren die beiden Kanten entlang der Längsachse. Sind die Positionen der absoluten Maxima der zu untersuchenden Histogramme bekannt, so kann der Vergleichsaufwand reduziert werden, indem der Verschiebungswert einmalig derart gewählt wird, dass die Maxima beider Histogramme aufeinander fallen.

Um den durch diese Vereinfachung entstandenen Fehler zu verringern, kann die Minimumsdifferenzsuche ausgehend von dieser initialen Position ausgedehnt werden. über eine Konstante, hier als Vergleichsfenster bezeichnet, kann der zu untersuchende Bereich festgelegt werden. Da die Winkelverteilungen der Fahrzeuge ein zweites Maximum aufweisen, welches um 180° respektive 64 Histogrammschritte gegenüber dem ersten Maximum verschoben ist, muss der gesamte Vergleich für dieses Maximum wiederholt werden, um den Fehler weiter zu minimieren.

Der Berechnungsaufwand wird durch diese Vereinfachung stark reduziert. Anstelle der ursprünglichen 16384 SAD-Operationen sind bei einem Vergleichsfenster von $[-1, 1]$ nur noch $128 \cdot 6 = 768$ SAD-Operationen notwendig. Selbst bei einem derart kleinen Vergleichsbereich wird die Erkennungsleistung kaum beeinträchtigt (Tarnowsky, 2009). Der wesentliche Nachteil dieser Vorgehensweise besteht jedoch

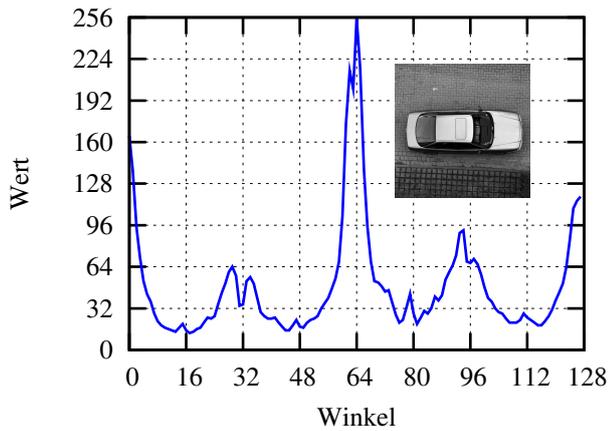


Abbildung 2. Geglättetes und normiertes Winkelhistogramm eines Fahrzeuges.

darin, dass der Algorithmus einen Teil seiner Flexibilität einbüßt: Objekte mit anderer Winkelverteilungscharakteristik, z. B. wenn sie keine deutlichen Maxima aufweisen, können unter Umständen nicht mehr zuverlässig detektiert werden. Abbildung 2 zeigt exemplarisch das geglättete Winkelhistogramm eines Fahrzeuges, welches der obigen Winkelverteilungscharakteristik entspricht.

4 Hardware-Architektur

Für die Hardware-Umsetzung ist aufgrund der Datenabhängigkeiten ein sequentieller Ablauf der drei Teilschritte Histogrammerzeugung, Histogrammglättung und Histogrammvergleich erforderlich. Im Folgenden soll die Implementierung der Teilschritte und deren Integration in eine dedizierte Recheneinheit vorgestellt werden.

4.1 Histogrammerzeugung

Die Implementierung des Verfahrens approximiert zunächst Gradientenwinkel und -beträge der zu untersuchenden ROI. Dazu werden die horizontalen und vertikalen Komponenten der Gradientendaten aus dem externen Speicher gelesen. Die in Abschnitt 3.2 dargestellte Approximation der Winkelberechnung basiert auf einem on-chip SRAM-Speicher. Für die anschließende Histogrammberechnung wird ein on-chip Dual-Port-SRAM-Speicher verwendet, so dass pro Takt ein Histogrammwert parallel gelesen und geschrieben werden kann. Dieser Ansatz ermöglicht bereits die Prozessierung eines Gradientenvektors $\begin{pmatrix} g_x \\ g_y \end{pmatrix}$ pro Takt.

Zur Steigerung des Durchsatzes können mehrere Gradientenvektoren parallel verarbeitet und der Histogrammberechnung zugeführt werden. Dazu wird die Funktionstabelle zur Winkelberechnung mehrfach implementiert. Jeweils ein Dual-Port-Speicher ermöglicht es, pro Takt die Winkel

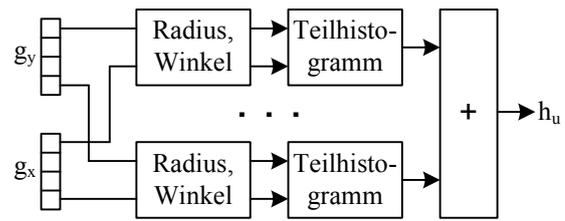


Abbildung 3. Architektur zur parallelen Histogrammberechnung.

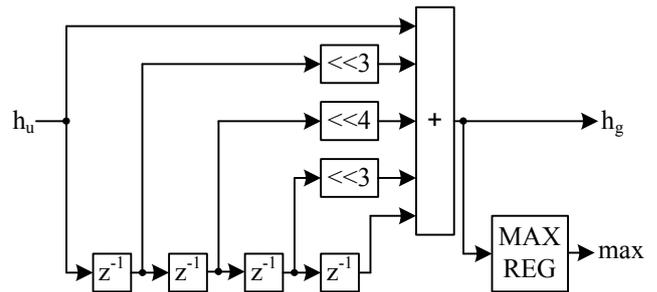


Abbildung 4. Architektur zur Histogrammglättung.

zweier Gradientenvektoren parallel zu berechnen. Zusätzlich müssen mehrere Teilhistogrammspeicher, entsprechend dem Grad der Parallelisierung, zur Verfügung stehen. In der nachfolgenden Stufe werden die Teilhistogramme parallel ausgelesen und deren Werte addiert. Bei maximaler Parallelität, welche durch die Datenzufuhr des eingesetzten 128-Bit-Systembusses begrenzt wird, können vier Gradientenvektoren mit jeweils 16 Bit pro Komponente gleichzeitig der Histogrammberechnung zugeführt werden. Das resultierende Blockschaltbild ist in Abb. 3 dargestellt.

4.2 Histogrammglättung

Die Glättung des Histogramms wird durch ein eindimensionales FIR-Filter realisiert. Die Impulsantwort des Filters wurde derart gewählt, dass sie weitestgehend einer Gauß-Filterung mit einer Varianz von $\sigma^2=0,72$ entspricht und sich die Koeffizienten durch Zweierpotenzen ausdrücken lassen. Diese Wahl der Impulsantwort ermöglicht somit eine sehr kostengünstige Umsetzung, da auf Hardware-Multiplizierer vollständig verzichtet werden kann.

Für die nachfolgende Normierung der Daten wird zusätzlich der Maximalwert der geglätteten Daten und dessen Position in einem Register gespeichert. Die Architektur der Histogrammglättung ist in Abb. 4 verdeutlicht.

4.3 Histogrammvergleich

Im nächsten Schritt erfolgt der Vergleich des geglätteten und normierten Winkelhistogramms mit mehreren

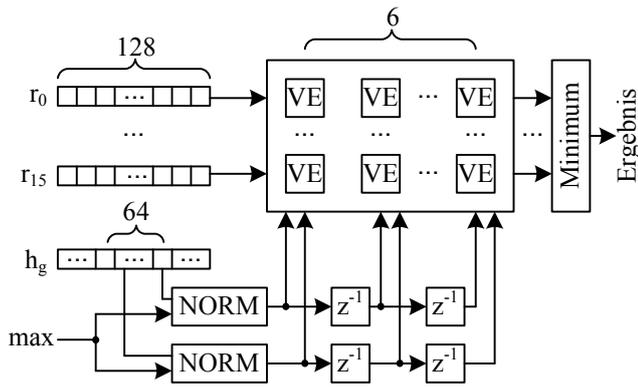


Abbildung 5. Architektur zum parallelen Histogrammvergleich.

Referenzhistogrammen. Wie in Abschnitt 3.3 beschrieben, muss das Auslesen des Winkelhistogramms an sechs verschiedenen Positionen beginnen. Der Dual-Port-Histogrammspeicher liefert pro Takt zwei Datenwörter beginnend an der um 1 dekrementierten Position des Maximums und der um 64 Werte verschobenen Position. Beide Werte werden daraufhin in einem Normierungsschritt durch Multiplikation mit 255 und Division durch den Maximalwert des Histogramms normiert. Nachfolgende Verzögerungsregister ermöglichen es, parallel jeweils ein Datenwort für alle sechs erforderlichen Vergleichsberechnungen bereitzustellen. Für jeden dieser Vergleiche existiert eine Vergleichseinheit, welche die in Gleichung (6) dargestellte SAD-Berechnung vornimmt. Zusätzliche Parallelität ergibt sich durch den nebenläufigen Vergleich aller Varianten des Winkelhistogramms mit mehreren Referenzhistogrammen. Für jedes zusätzliche Referenzhistogramm sind jeweils sechs weitere Vergleichseinheiten erforderlich. Bei einem 128-Bit-Systembus und einer Histogrammwortbreite von 8 Bit lassen sich ohne Wartezyklen maximal 16 Referenzhistogramme parallel vergleichen. In diesem Fall werden 96 SAD-Berechnungen parallel pro Takt vorgenommen. Nach erfolgter SAD-Berechnung werden der Vergleichswert und der Index des Referenzhistogramms mit dem minimalen Vergleichswert ermittelt und als Ergebnis ausgegeben. Die resultierende Architektur wird in Abb. 5 präsentiert.

4.4 Integration

Die Module Histogrammerstellung und Histogrammglättung kommunizieren über einen internen SRAM-basierten Dual-Port-Histogrammspeicher. Der externe Datenaustausch mit dem System-on-Chip erfolgt dabei über einen 128-Bit-Systembus (Flatt et al., 2007). Die resultierende Architektur ist in Abb. 6 dargestellt. Das Master-Interface dient dem selbstständigen Laden der Gradientendaten und Referenzhistogramme aus dem externen Systemspeicher. Der Referenzpuffer speichert einen Referenzdatensatz, so dass dieser

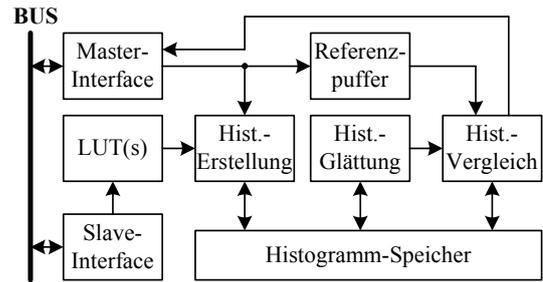


Abbildung 6. Hardware-Architektur zur Histogramm-basierten Winkelauswertung.

bei der Analyse weiterer ROIs nicht wiederholt aus dem externen Speicher geladen werden muss. Das Slave-Interface dient der Konfiguration der Architektur, dem Starten der Verarbeitung und dem einmaligen Hochladen der $\arctan\left(\frac{g_y}{g_x}\right)$ -Funktionstabelle nach dem Einschalten.

Die Anzahl parallel prozessierter Pixel für die Gradientenwinkelberechnung und Histogrammerstellung (Gradientenparallelität) sowie die Anzahl der parallel verarbeiteten Referenzhistogramme (Referenzparallelität) sind per Parameter vor der Schaltungssynthese konfigurierbar. Somit kann der maximale Datendurchsatz flexibel an die Echtzeitanforderungen angepasst werden.

5 Ergebnisse

Zur Evaluation wurde die VHDL-Implementierung der Hardware-Architektur zur Histogramm-basierten Winkelauswertung in die modulare Coprozessor-Architektur von Flatt et al. (2007, 2008, 2009a) integriert, die in Abb. 7 dargestellt ist. Ein weiterer zentraler Bestandteil der Coprozessor-Architektur ist eine Recheneinheit zur Gradientenfilterung (2D-FIR-Filter) (Flatt et al., 2009a). Beide Einheiten sind am gemeinsamen 128-Bit-Multi-Layer-Systembus angeschlossen. Für den Datenaustausch stehen interne und externe Speicher zur Verfügung. Die Interpretation der Vergleichsergebnisse ist Aufgabe des RISC-Prozessors, der über eine optimierte Steuerungsschnittstelle mit dem Coprozessor gekoppelt ist (Flatt et al., 2009b). Der Coprozessor wurde anschließend auf das Xilinx Virtex-5-FPGA-basierte ASIC-Emulationssystem CHIPit Iridium Edition V5 abgebildet (Synopsys, 2009) und emuliert. Die Emulationsfrequenz beträgt 100 MHz.

Wie in Abschn. 4 dargestellt, lassen sich die Gradienten- und Referenz-Parallelität konfigurieren und bestimmen maßgeblich die Anzahl der benötigten Lookup-Tables (LUTs) im FPGA. Abb. 8 zeigt die Anzahl der LUTs für verschiedene Konfigurationen der Architektur. Zusätzlich enthält die Architektur je nach Konfiguration 28 bis 56 KB-Block-RAM.

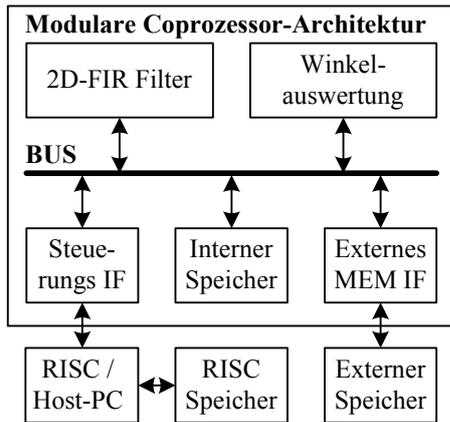


Abbildung 7. Modulare Coprozessor-Architektur.

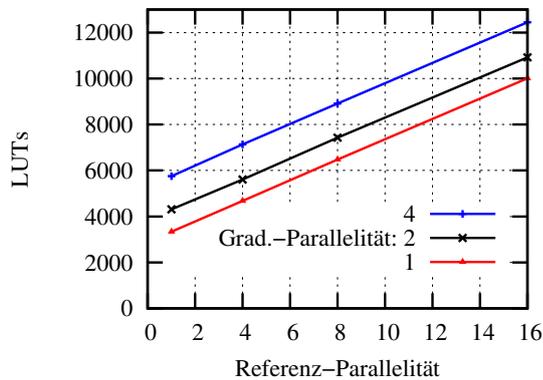


Abbildung 8. Synthesergebnisse, Variation von Referenz-Parallelität und Gradienten-Parallelität.

Die Taktfrequenz nach Platzierung und Verdrahtung beträgt 133 MHz.

Für eine praktische Anwendung der Hardware-Architektur wird angenommen, dass mit einem Referenzdatensatz eine Vielzahl von Objekten analysiert wird und dieser nur selten geändert wird. Somit ist es ausreichend, den Durchsatz ausschließlich in Abhängigkeit der ROI-Größe zu ermitteln. Der eigentliche Test erfolgt derart, dass je 100 Messungen für jede ROI-Größe zwischen 8×8 und 256×256 Pixel gemittelt wird. Da die Verarbeitungszeit des Verfahrens sowohl unabhängig von der Bildgröße als auch dem Bildinhalt ist, wurden Ausschnitte aus einem einzigen Testbild der Größe 768×576 verwendet. Die Anzahl parallel verarbeiteter Gradienten wurde ebenfalls variiert, da sie einen signifikanten Einfluss auf die gesamte Verarbeitungszeit des Verfahrens hat. Es wurden vier Referenzhistogramme verwendet, die von der Hardware-Architektur parallel analysiert werden können. Die Referenz- und Bilddaten sind im externen DDR-RAM des Coprozessors abgespeichert.

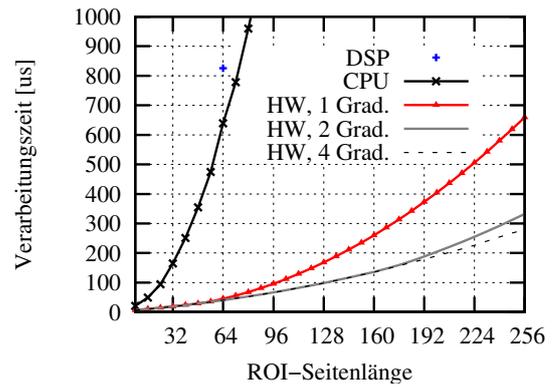


Abbildung 9. Vergleich der Verarbeitungszeit zwischen Hardware- und Software-Implementierung für quadratische ROIs.

Um die Verarbeitungszeit der Hardware-Architektur vergleichen zu können, wurde das Verfahren als Referenz-C-Code auf einen für mobilen Einsatz optimierten 1,6 GHz Intel Atom N270 General Purpose Prozessor abgebildet und evaluiert. Für einen weiteren Vergleich bei einer Objektgröße von 64×64 Pixeln wurde der C-Code auf einen 430 MHz Texas Instruments TMS320C64xTM DSP abgebildet und mit Intrinsic-Funktionen optimiert. Abbildung 9 zeigt die Verarbeitungszeiten der Hardware-Architektur für verschiedene Konfigurationen der Gradienten-Parallelität, der Intel Atom CPU und des DSPs.

Als Resultat ergibt sich sowohl für die CPU als auch für die Hardware-Architektur eine quadratische Abhängigkeit zwischen Seitenlänge und Verarbeitungszeit. Diese begründet sich insbesondere darin, dass die Anzahl der Objektpixel, aus denen das Winkelhistogramm ermittelt wird, quadratisch mit der Seitenlänge steigt.

Es wird deutlich, dass die Software-Implementierungen auf der CPU und dem DSP nur für sehr kleine Objekte den Anforderungen an eine echtzeitfähige Verarbeitung genügen. Soll die Intel Atom CPU in einem 25-Hz-Videodatenstrom beispielsweise 100 Objekte pro Bild analysieren, so entspricht dieses einer maximalen Objektgröße von 48×48 Pixeln. Die Hardware-Architektur kann bei 1-facher Gradienten-Parallelität bei gleicher Anforderung Objekte der Größe 192×192 Pixel analysieren, bzw. 256×256 bei 2-facher oder 4-facher Gradienten-Parallelität. Als weiteres Ergebnis zeigt sich, dass die Verarbeitungszeit mit 4-facher Gradienten-Parallelität nur für große Objekte signifikant reduziert werden kann. Dies begründet sich dadurch, dass die Gradientendaten als Region-of-Interest aus dem externen DDR-RAM Speicher blockweise gelesen werden müssen. Beim Lesen kleiner ROIs sind viele irreguläre Speicherzugriffe erforderlich, die eine Reduktion der Speicherbandbreite bewirken.

Der Histogrammvergleich muss in mehreren Durchläufen erfolgen, falls die konfigurierte Referenz-Parallelität kleiner

ist als die Anzahl der zu untersuchenden Referenzhistogramme. In diesem Fall erhöht sich die Verarbeitungszeit pro Durchlauf um 1,4 μ s.

6 Diskussion

Die echtzeitfähige Erkennung von Objekten anhand einfacher Merkmale wie Größe, Fläche, Position und Exzentrizität ist häufig nicht ausreichend zuverlässig. Als Lösungsansatz wurde das Objekterkennungsverfahren nach Korn vorgestellt, welches auf einer Histogrammauswertung der Gradientenverteilung basiert. Es ist insbesondere robust gegenüber Skalierung und Rotation der Objekte. Es wurde gezeigt, wie das Verfahren effizient in eine dedizierte Hardware-Architektur umgesetzt werden kann, die vollständig auf aufwändige Fließkomma-Operationen verzichtet. Gegenüber einer Software-Implementierung führt eine konfigurierbare Datenparallelität zu einer signifikanten Steigerung der Rechenleistung. Die Architektur benötigt auf einem Xilinx Virtex-5-FPGA je nach Konfiguration zwischen 3300 und 12000 LUTs sowie 28 bis 56 KB Block-RAM. Somit ist die Architektur in der Lage unabhängig von der Bildauflösung bei einer Taktfrequenz von nur 100 MHz in einem 25-Hz-Videodatenstrom bis zu 100 Objekte der Größe 256×256 Pixel in Echtzeit zu analysieren. Verglichen mit einem mobilen General Purpose Prozessor und einem digitalen Signalprozessor erzielt die Hardware-Architektur somit eine um ein bis zwei Größenordnungen bessere Durchsatzrate.

Discussion (English version)

The real-time object detection using simple features such as size, area, position, and eccentricity of objects is often not sufficiently reliable. As a solution, the object recognition approach of Korn was introduced, which is based on a histogram analysis of gradient distribution. It is particularly robust against scaling and rotation of objects. It was shown how the approach can be efficiently implemented as a dedicated hardware architecture, which completely avoids expensive floating-point operations. Compared to a software implementation, the configurable data parallelism leads to a significant increase in computing performance. Depending on the configuration, the architecture requires 3300 to 12 000 LUTs and 28 to 56 KB Block RAM on a Xilinx Virtex-5 FPGA. Running at a clock frequency of 100 MHz and independently of the image resolution per frame, 100 objects of size 256×256 pixels are analyzed in a 25 Hz video stream by the architecture. Finally, the data throughput of the hardware architecture is up to two orders of magnitude better compared to a mobile general-purpose processor and a digital signal processor.

Literatur

- Dalal, N., Triggs, B., and Schmid, C.: Human detection using oriented histograms of flow and appearance, in: European Conference on Computer Vision, 7–13, Springer, 2006.
- Flatt, H., Hesselbarth, S., Flügel, S., and Pirsch, P.: A Modular Coprocessor Architecture for Embedded Real-Time Image and Video Signal Processing, in: Embedded Computer Systems: Architectures, Modeling, and Simulation, edited by: Vassiliadis, S., 4599, 241–250, Springer, 2007.
- Flatt, H., Blume, S., Hesselbarth, S., Schünemann, T., and Pirsch, P.: A Parallel Hardware Architecture for Connected Component Labeling Based on Fast Label Merging, in: 19th IEEE International Conference on Application-specific Systems, Architectures and Processors, ASAP, 144–149, IEEE, 2008.
- Flatt, H., Blume, S., Tarnowsky, A., Blume, H., and Pirsch, P.: Echtzeitfähige Abbildung eines videobasierten Objekterkennungsalgorithmus auf eine modulare Coprozessor-Architektur, in: ITG Fachtagung für Elektronische Medien Systeme, Technologien, Anwendungen, 13. Dortmunder Fernsehseminar, VDE-Verlag, 2009a.
- Flatt, H., Schmäddecke, I., Kärigel, M., Blume, H., and Pirsch, P.: Hardware-Based Synchronization Framework for Heterogeneous RISC/Coprocessor Architectures, in: Proceedings of the Intl. Conference on Embedded Computer Systems: Architectures, Modeling and Simulation (IC-SAMOS 2009), 2009b.
- Greenberg, S., Rotman, S. R., Guterman, H., Zilberman, S., and Gens, A.: Region-of-interest-based algorithm for automatic target detection in infrared images, *Optical Engineering*, 44, 77 002, 2005.
- Jähne, B.: *Digitale Bildverarbeitung*, Springer, 6 edn., 2005.
- Korn, A.: Verfahren zur Bildauswertung, EP0394959, 1998.
- Liu, D., He, L., and Carin, L.: Airport detection in large aerial optical imagery, in: Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on, 5, pp. V-761–4, 2004.
- Lowe, D. G.: Distinctive Image Features from Scale-Invariant Key-points, *International Journal of Computer Vision*, 60, 91–110, 2004.
- Messelink, W., Schutte, K., Vossepoel, A., Cremer, F., Schavemaker, J., and den Breejen, E.: Feature-based detection of landmines in infrared images, in: Detection and Remediation Technologies for Mines and Minelike Targets VII, edited by: Broach, J., Harmon, R., and Dobeck, G., vol. 4742 of *Proceedings SPIE*, 108–119, 2002.
- Scharr, H.: *Optimal Operators in Digital Image Processing*, Dissertation, Ruprecht-Karls-Universität Heidelberg, 2000.
- Schönfeld, J.: *Kompakte Implementierung konturorientierter Bildverarbeitungssysteme mit VLSI-Bausteinen*, Dissertation, Universität Hannover, 1994.
- Synopsys: *CHIPit Iridium Edition*, <http://www.synopsys.com>, 2009.
- Tarnowsky, A.: *Konzeption und VHDL-Implementierung einer Hardware-Architektur zur vergleichsbasierten Auswertung von Winkelverteilungsräumen*, Studienarbeit, Leibniz Universität Hannover, Institut für Mikroelektronische Systeme, 2009.
- Yu, B., Zhang, W., and Cai, Y.: A Lane Departure Warning System Based on Machine Vision, in: Computational Intelligence and Industrial Application, 2008. PACIIA '08. Pacific-Asia Workshop on, 1, 197–201, 2008.