

Online-Prozessor für ein Echtzeit-Bildverarbeitungssystem

Steffen Dolling, Rüdiger Ide, Gerhard Kutschke, Dirk Timmermann*

Universität Rostock, Fachbereich Elektrotechnik und Informationstechnik,
Institut für Angewandte Mikroelektronik und Datentechnik,
R.-Wagner-Str. 31, D-18119 Rostock,
email: dol@baltic.e-technik.uni-rostock.de

* Graphikon Gesellschaft für Bildverarbeitung und Computergraphik mbH,
Mandelstr. 16, D-10409 Berlin

Abstract

In dieser Arbeit wird ein Prozessor zur Analyse von Rasterbildern vorgestellt. Das implementierte Randlinien-Code-Verfahren (RLC-Verfahren) basiert auf einem inkrementellen Algorithmus, der die Zusammenhangsanalyse der im Bild enthaltenen Objekte sowie die Ermittlung sie beschreibender morphometrischer Merkmale und Nachbarschaftsbeziehungen in einem single-pass Verfahren ausführt. Hierbei wird das Rasterbild in eine Konturbeschreibung transformiert und die Gebietshierarchie der enthaltenen Objekte ermittelt. Parallel hierzu berechnet der Prozessor einen Satz der wichtigsten Objektmerkmale, wie z.B. Fläche und Umfang. Die ermittelten Daten werden über einen Pufferspeicher an einen Hostrechner übergeben. Das Konzept des Online-Prozessors, dessen einzelne Bestandteile als parametrierbare VHDL-Modelle beschrieben wurden, ist für die Analyse mehrwertiger Rasterbilder (Farbbildanalyse) ausgelegt. Die Simulation und Synthese in eine XILINX-FPGA-Technologie erfolgte dagegen für die Bearbeitung zweiwertiger Rasterbilder (Binärbildanalyse).

1. Einleitung

Anwendungsspezifische Bildverarbeitungsapplikationen, wie die Erkennung und Verfolgung von bewegten Objekten in Bildfolgen, erfordern echtzeitfähige Verarbeitungslösungen für die Ermittlung wichtiger Objektmerkmale sowie die Zusammenhangsanalyse. Konturfolge- bzw. Mehrpaßverfahren zeichnen sich durch einfache Verarbeitungsalgorithmen aus, benötigen aber einen Bildspeicher zur wahlfreien Bereitstellung der Bilddaten bzw. zur Speicherung von Zwischenergebnissen. Da diese Verfahren keine online-Verarbeitung der Rasterbilddaten gestatten, sind sie für die Ausführung von Echtzeit-Bildverarbeitungsaufgaben nicht geeignet. Zeilenkoinzidenzverfahren, die eine online-Zusammenhangsanalyse des Bildes durch den

Vergleich zweier aufeinander folgender Bildzeilen durchführen, ermöglichen die Entwicklung hierfür geeigneter Lösungen. Zeilenkoinzidenzverfahren auf der Basis lauflängencodierter Bilder (Run length code Verfahren) bieten Vorteile bei der Ermittlung statistischer Merkmale im Bild, haben aber entscheidende Nachteile bei der Ermittlung der Konturbeschreibung von Objekten. Das RLC-Verfahren setzt im Gegensatz hierzu direkt auf der Verarbeitung der Rasterbilddaten auf. Das Rasterbild wird in eine kompakte Form überführt, in dem jedes Zusammenhangsgebiet durch seine Randlinie beschrieben wird. Der inkrementelle Algorithmus realisiert Zusammenhangsanalyse, Kontur- und Merkmalsberechnung in einem single pass-Verfahren. Um eine weitere Geschwindigkeitsoptimierung bzw. eine Dezentralisierung des rechenintensiven Prozesses zu erreichen, wurde ein Hardwarekonzept für einen Online-Prozessor entwickelt. Dieser integriert eine spezielle Adaption des RLC-Verfahrens und ist auf mehrfarbige Bilder ausgerichtet. Die strukturierte Lösung besteht aus verschiedenen parametrierbaren Einzelmodulen, die als konfigurierbare VHDL-Modelle vorliegen. Zur Verifikation und Validierung wurde ein Prozessor für die Bearbeitung von Binärbildern in eine XILINX-FPGA-Technologie synthetisiert. Folgende Ausführungen beziehen sich damit auf diesen Fall.

2. Das RLC-Verfahren

Das Rand-Linien-Code-Verfahren ist ein inkrementelles Verfahren zur Codierung von Pixelbildern. Während der zeilenweisen Abtastung des Pixelbildes werden stets die Konturen der gleichfarbigen Gebiete gebildet, die somit durch ihre Randlinie beschrieben werden. Dabei werden die Konturen der Objekte als eckig angenommen und somit läßt sich eine Kontur als Folge von horizontalen und vertikalen Linien zwischen je zwei Ecken auffassen. Die Randlinie wird in einer kompakten Form als sogenannter RLC-String abgespeichert, wobei die Kontur eines gleichfarbigen Gebiets als Vektormenge ausgedrückt wird. Zu jedem Gebiet werden gleichzeitig wesentliche Merkmale, die eine schnelle Klassifizierung erlauben, errechnet. Außerdem wird die Gebietshierarchie ermittelt. Somit ist nach der Abarbeitung der letzten Bildzeile die Transformation in eine symbolische Beschreibung komplett, womit sich dieser Algorithmus für Video-Echtzeitanwendungen anbietet. Der Rechenzeitbedarf für diesen Algorithmus wird dabei im wesentlichen durch die Anzahl der Ecken im Bild bestimmt. Zur Analyse des Bildes werden der linken oberen Ecke des linken oberen Pixels die Koordinaten (0,0) des Koordinatensystems zugeordnet, d.h. die Koordinaten liegen an den Berührungspunkten der Pixel. Auf der Basis eines 2x2 Fensters, in dessen Zentrum der Berührungspunkt vier benachbarter Pixel liegt, werden die möglichen Ecken beschrieben (siehe Abb. 1).

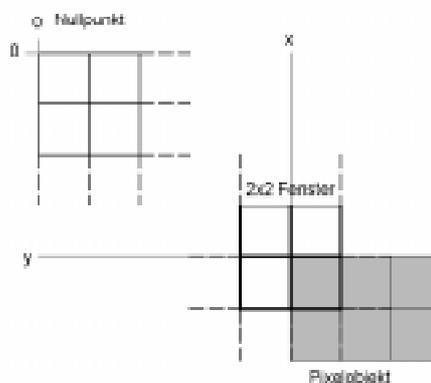


Abb. 1: Eckendefinition

Von den 16 Möglichkeiten, die im binären Fall durch ein solches 2x2 Feld beschrieben werden können, bleiben acht Fälle übrig, die je eine Ecke darstellen. Hinzu kommen zwei Muster, welche man als sogenannte Doppellecken bezeichnet. Diese werden als aneinanderstoßende Ecken zweier verschiedener schwarzer Objekte gewertet (siehe Tab. 1).

Muster im 2x2 Fenster	Bedeutung
 	gleichfarbige Fläche
   	senkrechte oder waagerechte Kante
       	gültiger Eckencode
 	Doppellecken

Tab. 1: Muster im 2x2 Fenster eines Binärbildes

Je zwei Ecken in einer Zeile oder Spalte bilden ein Paar und damit den Anfang und das Ende eines horizontalen bzw. vertikalen Kantenabschnitts (siehe Tab. 2).

Eckenpaar	Bedeutung
 	Eckenpaar eröffnet neue Teilkontur; mögliches weißes Gebiet
 	Eckenpaar eröffnet neue Teilkontur; mögliches schwarzes Gebiet
 	Eckenpaar versetzt eine vertikale Kante nach links
 	Eckenpaar versetzt eine vertikale Kante nach links
 	Eckenpaar versetzt eine vertikale Kante nach rechts
 	Eckenpaar versetzt eine vertikale Kante nach rechts
 	Eckenpaar schließt eine Kontur (weißes Gebiet) oder verschmilzt zwei Konturen
 	Eckenpaar schließt eine Kontur (schwarzes Gebiet) oder verschmilzt zwei Konturen

Tab. 2: Eckenpaare im Binärbild

Während der Konturermittlung werden aus den Eckenpaaren Konturelemente gebildet. Dabei werden die ein Objekt beschreibenden und damit zusammengehörigen Elemente in der Rei-

henfolge ihres Auftretens miteinander in einer Konturelementliste verkettet. Somit ergibt sich für jedes Objekt ein geschlossener Umlauf. Da durch die Eckenpaare nur horizontale Kanten darstellbar sind, müssen die entstehenden offenen vertikalen Kanten pro Spalte zwischengespeichert werden. Eine Verkettung der entsprechenden Konturelemente erfolgt erst mit der Bearbeitung der unteren zugehörigen Ecke. Ebenso können auf Grund des inkrementellen Verfahrens „Pseudokonturen“, wie z.B. Einschnitte in einzelne Objekte, auftreten. Diese können erst erkannt werden, wenn eine horizontale Kante zwei Konturen verbindet, was zur Folge hat, daß diese miteinander verschmolzen werden. Deshalb werden bis zu diesem Zeitpunkt Einschnitte als eigenständige Objekte behandelt (siehe Abb. 2).

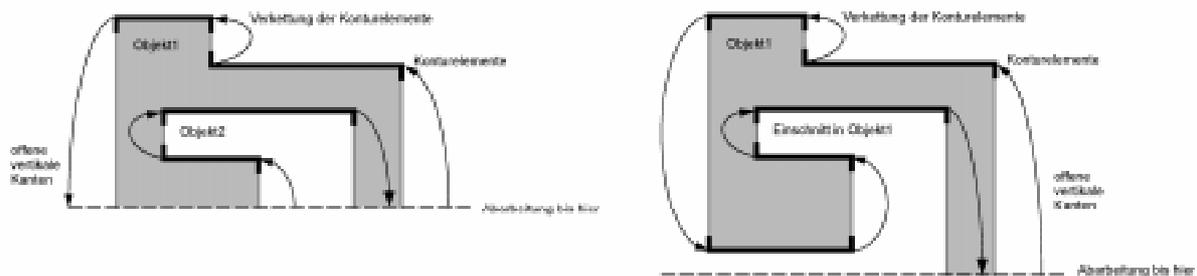


Abb. 2: Verkettung und Verschmelzung von Konturen

Für jedes durch die Konturermittlung erkannte gleichfarbige Gebiet - das sogenannte RLC-Objekt - wird bei dessen Öffnung ein Datensatz erstellt. Dieser enthält alle für das Objekt repräsentativen Pointer, wie z.B. die Zeiger auf die Eintrittspunkte in die Konturelementliste, und Merkmale, wie z.B. die Farbe des Objekts (siehe Abb. 3). Der RLC-Datensatz wird online während der Konturermittlung aktualisiert.

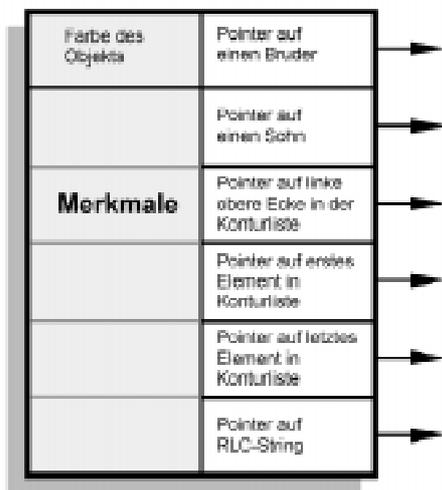


Abb. 3: Datensatz eines RLC-Objekts

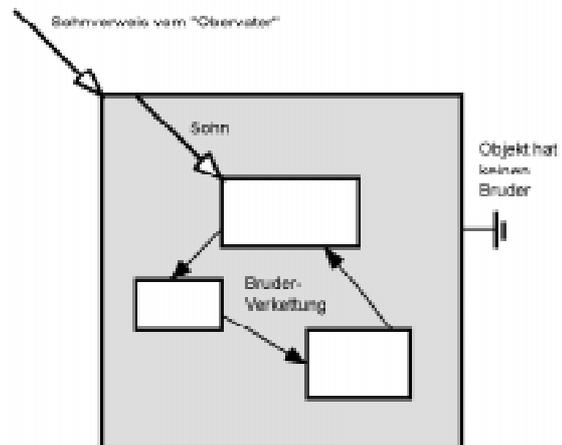


Abb. 4: Bruder-Sohn-Verkettung

Im Datensatz eines RLC-Objekts sind ebenfalls Informationen über die Gebietshierarchie enthalten, über die Beziehungen zwischen gleich- oder verschiedenfarbigen Gebieten hergestellt werden können. In diesem Zusammenhang werden „Väter“, „Söhne“ und „Brüder“ unterschieden (siehe Abb. 4). Ein „Sohn“ ist ein Gebiet, das durch eine andersfarbige Kontur umschlossen wird, die somit den „Vater“ darstellt. Als „Brüder“ werden gleichfarbige Objekte,

die gemeinsam in einem andersfarbigen Gebiet liegen, bezeichnet. Dabei wird der Rand des gesamten Bildes als Kontur des ersten RLC-Objekts aufgefaßt, das damit der „Obervater“ für alle weiteren Gebiete ist. Die Aktualisierung der Gebietshierarchie erfolgt jeweils nach dem Schließen einer Kontur durch eine Betrachtung der offenen vertikalen rechten oder linken Kanten neben dem geschlossenen Objekt. Dabei werden ebenfalls die oben erwähnten Konturverschmelzungen eingeschlossen. In Abbildung 5 ist ein Beispiel mit 20 Objekten (einschließlich Pseudoobjekte) und deren Gebietshierarchie dargestellt.

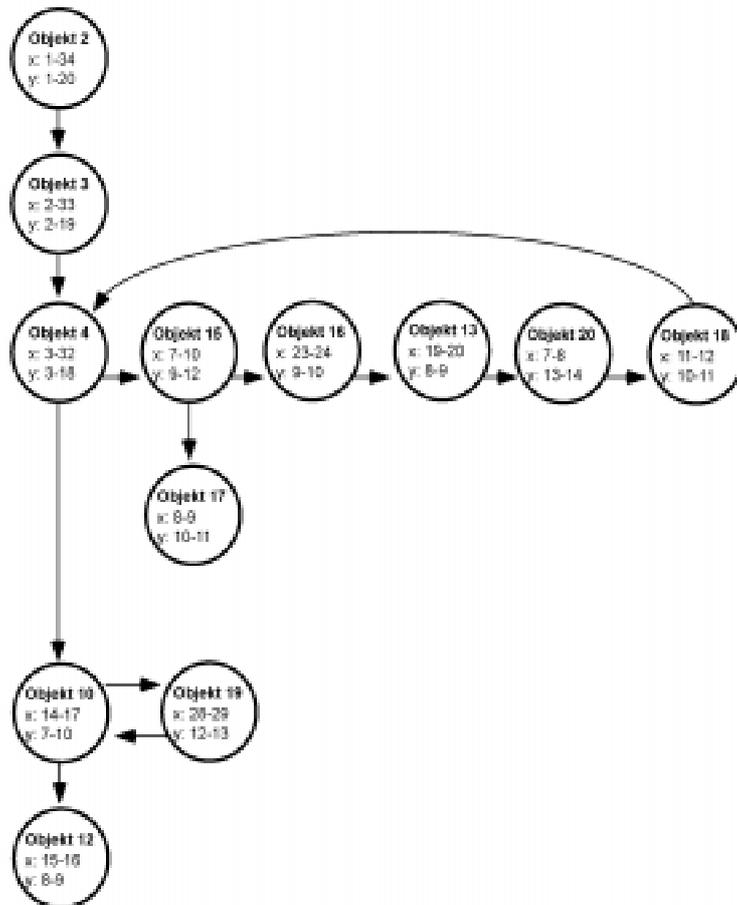
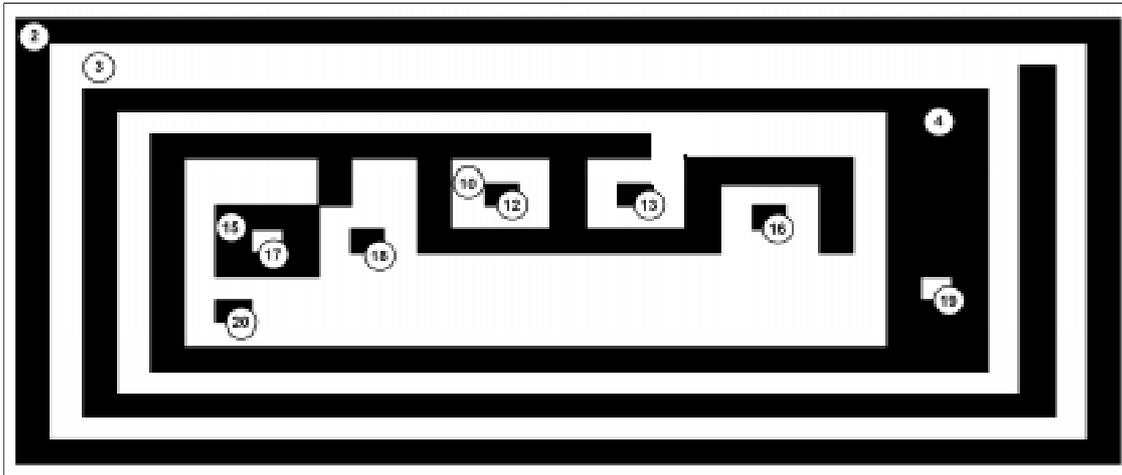


Abb. 5: Binärbild mit 20 Objekten und deren Gebietshierarchie

3. Der Online-Prozessor

Der unter Video-Echtzeitbedingungen arbeitende Online-Prozessor erfasst gleichfarbige Konturen in einem Bild, bestimmt grundlegende Merkmale dieser Objekte und ermittelt deren hierarchische Beziehungen. Der Prozessor erhält dabei aus einem vorgeschalteten Modul einen aus einer Pixeldarstellung generierten Strom von Ecken, deren Abfolge den auftretenden Ecken bei einer zeilenweisen Abtastung des Bildes entspricht. In der Reihenfolge ihres Eintreffens werden diese Eckencodes in einer speziell auf die Hardwarebelange zugeschnittenen Variante des RLC-Verfahrens [3] verarbeitet. Die ermittelten RLC-Datensätze werden in einem externen RAM zwischengespeichert und sind dort für einen Host zur weiteren Verarbeitung verfügbar (siehe Abb. 6). Dieser soll aber lediglich zeitunkritische Aufgaben wahrnehmen.

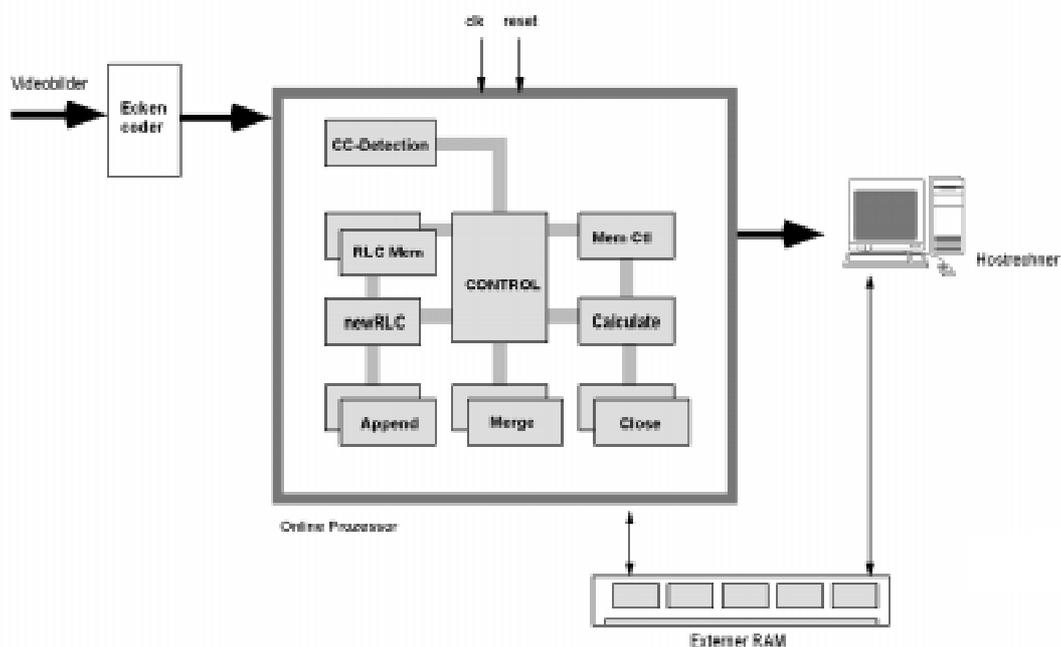


Abb. 6: Online-Prozessor und Umgebung

Der Prozessor erzeugt bzw. bearbeitet online im wesentlichen folgende Datenstrukturen:

a) RLC-Objektdatensatz

Für jedes Objekt wird ein RLC-Objektdatensatz erzeugt, der alle notwendigen Informationen über das entsprechende Gebiet enthält. Das sind zum einen die Merkmale, wie Objektfarbe, umschließendes Rechteck usw.. Außerdem erfolgen über die Pointer die Zugriffe auf die anderen bildbeschreibenden Datenstrukturen (Konturliste, Markierliste, Vaterliste).

b) Konturelement

Dieser Datensatz beschreibt die aus den Eckenpaaren gebildeten Kantenabschnitte. Entsprechend sind hier Daten über den Eckenpaartyp, die Lage oder die Länge des Konturelements abgelegt. Die einzelnen Elemente sind in einer Konturliste zusammengefaßt. Durch einen Zeiger auf den jeweils nächsten Datensatz ergibt sich die Zusammengehörigkeit mehrerer Konturelemente und damit die Randlinie eines Objektes.

c) Markierleistenelement

Die Markierleiste verwaltet die Informationen über die offenen senkrechten Kanten in einem Bild. Dazu existiert für jede mögliche Spalte ein Datensatz (Markierleistenelement), der einen Zeiger auf das zu der offenen Kante gehörige RLC-Objekt sowie dessen Farbe enthält. Die Farbinformation ist dabei zur Unterscheidung zwischen „Vater“ und „Bruder“ bei der Bestimmung der Gebietshierarchie notwendig.

d) Vaterlistenelement

Die Vaterliste enthält für jedes RLC-Objekt einen Eintrag (Vaterlistenelement). Mit den entsprechenden Zeigern auf den nächsten bzw. den vorhergehenden „Bruder“ bzw. auf einen „Sohn“ wird die Bestimmung der Gebietshierarchie eines Bildes gewährleistet.

Entsprechend den beschriebenen Datenstrukturen und der Funktionalität des RLC-Verfahrens wurde die in Abbildung 7 dargestellte Struktur für den Online-Prozessor entworfen.

Das Gesamtverhalten des Bildverarbeitungs-Prozessors wird durch das Zusammenwirken verschiedener unabhängiger hierarchisch angeordneter Module erreicht:

a) Eckenpaarbestimmung (CC-Detection)

Die Eckenpaardetektion bildet aus zwei hintereinander einlaufenden Eckencodes, die der Prozessor von der vorgeschalteten Eckenerkennung einliest, ein gültiges Eckenpaar. Dabei werden die beiden Doppelseiten in zwei einzelne Ecken aufgelöst. Ebenso wird hier ein Bild-Ende-Eckenpaar-Code zur Kennzeichnung eines Bildabschlusses erzeugt. Die CC-Detection arbeitet autonom zur Steuereinheit des Prozessors.

b) Steuereinheit (Control)

Die Steuereinheit als zentrales Element der gesamten Struktur ist durch die Abarbeitung von zwei unabhängigen internen Prozessen gekennzeichnet. Einerseits wird hier die Koordination der meisten anderen Module und damit des gesamten adaptierten RLC-Algorithmus ausgeführt. Entsprechend ist eine komplexe Zustandsmaschine zur Generierung bzw. Überwachung der Signale des Steuerbusses implementiert. Andererseits wird hier die Überprüfung sämtlicher auftretender Fehlermeldungen innerhalb des Prozessors realisiert. Die Behandlung umfaßt die Ausgabe eines Fehlercodes und das Rücksetzen der Verarbeitungseinheiten. Daraufhin wird eine durch die Peripherie auszuführende Synchronisation des einlaufenden Eckenstromes auf einen Bildbeginn erwartet.

c) neues Objekt anlegen (NewRLC)

Dieses Modul erzeugt aus einem eröffnenden Eckenpaar eines neues RLC-Objekt. Dabei erfolgt das Setzen der entsprechenden Pointer auf die Konturliste, eine Aktualisierung der Merkmale sowie eine Änderung der Markierungsleiste und der Vaterliste.

d) Kontur erweitern (Append)

Der Append-Block führt eine Konturerweiterung nach rechts oder links durch und wird entsprechend bei vier der möglichen Eckenpaare aktiviert. Das Ergebnis ist die Verschiebung einer offenen vertikalen Kante nach rechts oder links und die damit verbundene Aktualisierung der Datensätze.

e) Konturen verschmelzen (Merge)

Das Merge-Element wird bei Eckenpaaren aufgerufen, die ein Schließen einer Kontur oder die eine Konturverschmelzung hervorrufen können. Durch eine Überprüfung der in der Markierungsleiste gekennzeichneten Objekte mit den offenen Kanten wird festgestellt, welcher der genannten beiden Fälle vorliegt. Zeigen beide Pointer auf dasselbe

RLC-Objekt, wird über die Steuereinheit das Close-Modul aktiviert. Andernfalls werden die entsprechenden Gebiete verschmolzen, wofür in Abbildung 8 mögliche Varianten dargestellt sind.

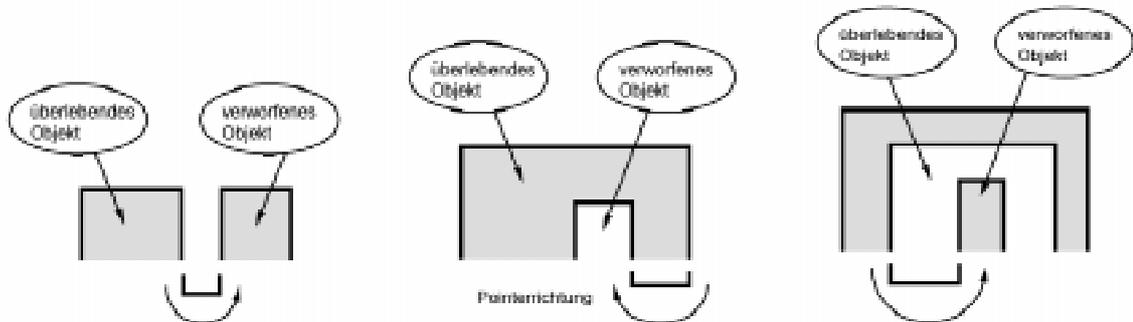


Abb. 8: Beispiele möglicher Verschmelzungen

f) Kontur schließen (Close)

Das Close-Modul wird durch die Steuereinheit aktiviert, wenn die Merge-Einheit zwei zu verbindende vertikale Kanten als zu einem Objekt gehörend identifiziert hat. In der Folge wird das gleichfarbige Zusammenhanggebiet geschlossen sowie die Bruder-Sohn-Verkettungen der Struktur aktualisiert.

g) Merkmale berechnen (Calculate)

Die Merkmalsberechnung wird nach jeder Änderung der Kontur eines RLC-Objektes durchgeführt und bezieht sich z.B. auf das umschließende Rechteck eines Gebietes.

h) Bussysteme

Neben dem Steuerbus sind vier Busse (RLC-Bus, Konturelement-Bus, Markierleisten-Bus, Vaterlisten-Bus) zum Datentransfer zwischen den einzelnen Modulen bzw. dem Prozessor und dem externen Speicher implementiert, deren physische Struktur an die zu verarbeitenden Datensätze angepaßt ist. Die Koordination der Buszugriffe erfolgt über spezielle Multiplexer.

i) Speicherverwaltung

Für eine flexible Speicherverwaltung besitzt der Online-Prozessor ein Memory-Control-Interface (RLC-Memory-Control, CEL-Memory-Control) für den Anschluß externer RAM's, in denen die während der Bildanalyse anfallenden Daten abgelegt werden.

Die einzelnen Module des Online-Prozessors sind sowohl als VHDL-Verhaltensbeschreibungen auf der algorithmischen Ebene als auch als synthesefähige VHDL-Modelle auf der Register-Transfer-Ebene entwickelt worden. Der gesamte Entwurf wurde auf eine hohe Modularität sowie die Parametrierbarkeit der Datensätze und der einzelnen internen Baugruppen ausgerichtet. Zur Verifikation und Validierung wurde eine entsprechende Umgebung mit Simulationsskripten und Testbench-Files, in die verschiedene Testdateien für zahlreiche unterschiedliche Bildszenarios eingebunden werden können, entwickelt. Somit ist das Bildverarbeitungselement durch eine erneute Ausführung des VHDL-Top-Down-Entwicklungsprozesses schnell auf spezielle Anforderungsprofile adaptierbar. Zur Erhöhung der Durchsatzrate des Prozessors könnten z.B. einzelne Module parallel implementiert werden. Ebenso ist die Berechnung weiterer Merkmale oder eine teilweise Serialisierung der Datenübertragung in die externen Speicher möglich. Somit kann ein geeignetes Verhältnis zwischen notwendiger Chipfläche, der Geschwindigkeit und der Eingabe/Ausgabe-Bandbreite erreicht werden.

Eine derzeit in eine XILINX-FPGA-Technologie synthetisierte Struktur geht von 500 RLC-Objekten, 2000 Konturelementen, einer Bildgröße von 2048x2048 Pixeln und einem voll parallel ausgeführten Speicher-Interface aus. Hinsichtlich des Flächenbedarfs werden derzeit 3791 konfigurierbare Blöcke (CLB's) benötigt. Entsprechend der Gesamtanzahl der CLB's ist das Design z.B. in vier XILINX-FPGA's vom Typ XC-4036EX mit einer Bausteinauslastung von 75 % implementierbar.

Der dargelegte Flächenbedarf läßt sich durch eine schon oben erwähnte Serialisierung der Datenströme minimieren. Dies ist auf eine Analyse der Synthesergebnisse zurückzuführen, die zeigte, daß der größte Teil der Ressourcen für interne Register aufgewendet wird. Allerdings wäre eine derartige Optimierung mit einer Verringerung der Verarbeitungsgeschwindigkeit verbunden.

Durch umfangreiche Simulationsläufe konnte für die derzeitige Konfiguration bei einer Taktfrequenz von 20 MHz eine Verarbeitungsdauer von durchschnittlich 21 Takten pro einlaufender Ecke ermittelt werden. Für das in Abbildung 6 dargestellte Binärbild ergab sich z.B. eine Laufzeit von 80,6 Mikrosekunden.

4. Zusammenfassung

In der vorliegenden Arbeit wurde eine Hardwareentwicklung für ein adaptiertes RLC-Verfahren zur Video-Echtzeitverarbeitung gezeigt. Die dabei entstandene VHDL-Beschreibung des Online-Prozessors weist eine modulare und auf verschiedene Anforderungen hin konfigurierbare Struktur auf. Die Synthese einer Musterapplikation zeigte, daß die entwickelte Lösung in einer FPGA-Struktur realisierbar ist. Für eine weitere Minituarisierung ist eine Implementierung in einen ASIC möglich. Somit werden geschwindigkeitsoptimale und dezentrale Anwendungen des Online-Prozessors in der Bildverarbeitung möglich.

Literatur

- [1] E. Mandler, M. Oberländer, „Ein single-pass Algorithmus für die schnelle Konturcodierung von Binärbildern“, 12. DAGM-Symposium, 1990
- [2] S. Eppler, „Randlinien-Code-Verfahren“, Diplomarbeit TU Braunschweig, 1992
- [3] G. Kutschke, „Beschreibung eines Online-Prozessors für EFA2“, interner Bericht, Graphikon GmbH, 1995
- [4] E. Mandler, M. Oberländer, „One-pass Encoding of Connected Components in Multi-Valued Images“, Proc. Of the 10th ICPR, Vol. II pp. 64-69, 1990
- [5] R. Ide, „VHDL-Synthese eines Prozessors für ein Echtzeit-Farbbild-Verarbeitungssystem“, Studienarbeit, Universität Rostock, 1995
- [6] R. Ide, „FPGA- Synthese eines Prozessors für ein Echtzeit-Farbbild-Verarbeitungssystem“, Diplomarbeit, Universität Rostock, 1996

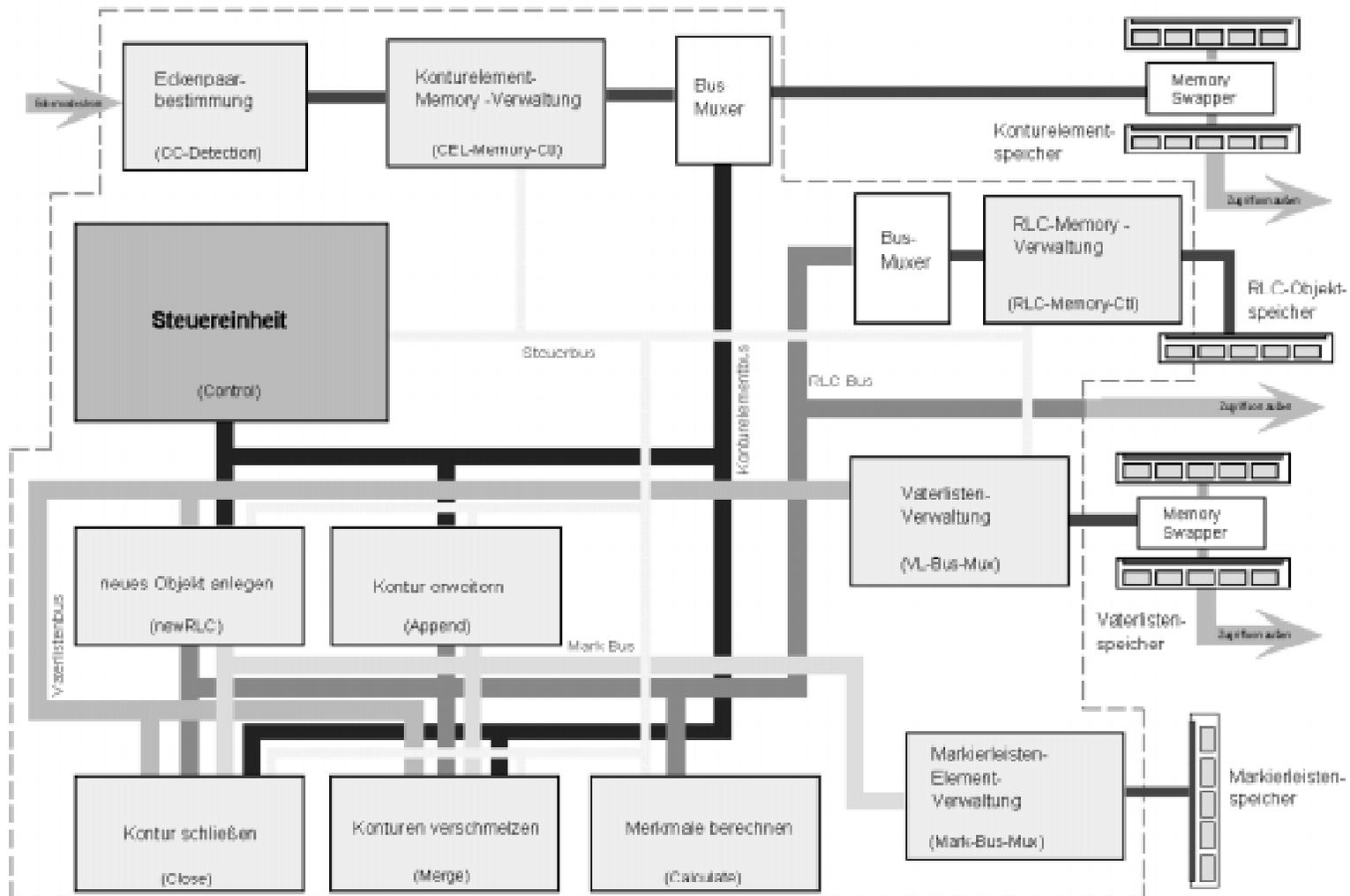


Abb. 7: Struktur des Online-Prozessors