

Bilddaten wirkungsvoll reduzieren

Konzentration auf das Wesentliche

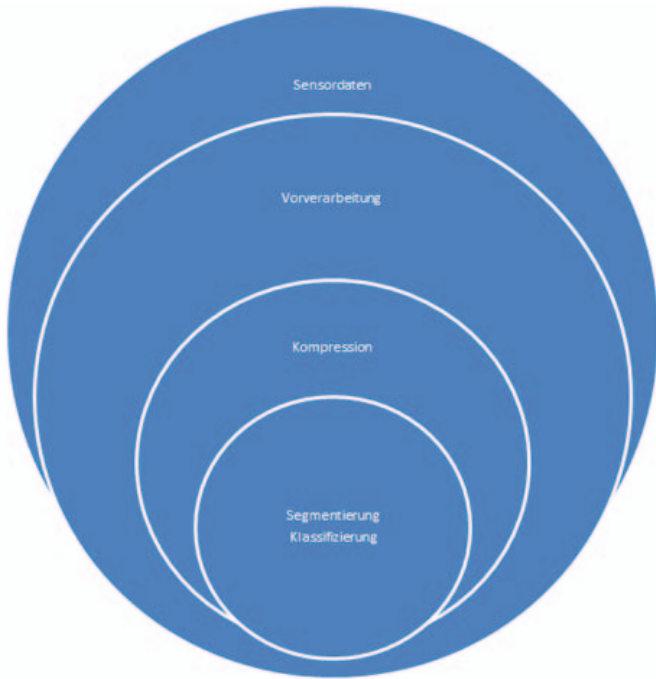


Bild 1: Datenreduktion entlang der Bildverarbeitungskette (alle Bilder: Silicon Software GmbH)

In der Bildverarbeitung ist der Trend zu immer kleineren Vision-Geräten vorgezeichnet, die immer größere Datenmengen immer schneller verarbeiten, etwa aufgrund gesteigerter Produktionsgeschwindigkeit. Welche Verfahren in der Bildverarbeitungskette zum Einsatz kommen,

um Daten zu strukturieren, komprimieren und letztendlich zu reduzieren, zeigt der Artikel anhand einiger Beispiele auf. Was schon immer eine der Hauptaufgaben der Bildverarbeitung war, nämlich relevante Informationen durch Datenreduktion zu extrahieren, erhält nun vor dem Hintergrund verfügbarer Komponenten, Bandbreiten und Speicherressourcen ungeahnte neue Möglichkeiten.

Informationsreduktion

Die Informationsreduktion beginnt lange vor der Ablage der Bilddaten im Rechner. Ein großer Teil der Bilddatenreduktion findet bereits durch passende Beleuchtung, Optik und Kameraelektronik statt. Nach Durchführung der Aufnahme gilt es, die durch den Sensor gelieferten digitalen Daten direkt vor Ort zu korrigieren, beispielsweise durch Neusortierung der übertragenen Sensorgeometrie, der sogenannten „Tap Geometry Sorting“. Dadurch kommen die Bilddaten für die weitere Verarbeitung in die richtige Reihenfolge, was deren Datenmenge jedoch erhöht.

Hier kommt, bevor die Daten das Aufnahmegerät verlassen, die Kür der Bildverarbeitung in Bezug auf Datenreduktion zum Einsatz: die Bildvorverarbeitung direkt in der Kamera

oder dem Sensor beziehungsweise auf den Prozessoren einer Grafikkarte und Framegrabbern. Bei diesem Verarbeitungsschritt werden zunächst Bildstörungen wie Helligkeit, Kontrast und Rauschen durch geeignete Algorithmen restauriert und ein Bild rekonstruiert, beispielsweise für die Computer-Tomografie. Anhand der Binarisierung des Bildes auf eine begrenzte Zahl an Graustufen (z.B. Reduzierung der Bittiefe von 256 auf 16 Graustufen) und Erstellen eines Kantenbildes wird in Verbindung mit der Blob-Analyse ein großer Teil der tatsächlich vorhandenen Information entfernt, um sich auf das Wesentliche zu konzentrieren und die spätere Bildauswertung zu erleichtern. Wird die Vorverarbeitung parallel zur Bildaufnahme durchgeführt, entlastet dies die CPU des Host-PCs.

Weitere Schritte der Bildverarbeitung dienen dazu, Objekte in Bildern zu identifizieren, analysieren und signifikante Objektmerkmale zu extrahieren, um abschließend eine Klassifizierung mit quantitativen und qualitativen Aussagen der Objekte, wie Größe und Farbe, zu treffen. Hier kommen abermals nur relevante Bildbestandteile zum Tragen. Nach der Segmentierung von Objekten werden deren Geo-

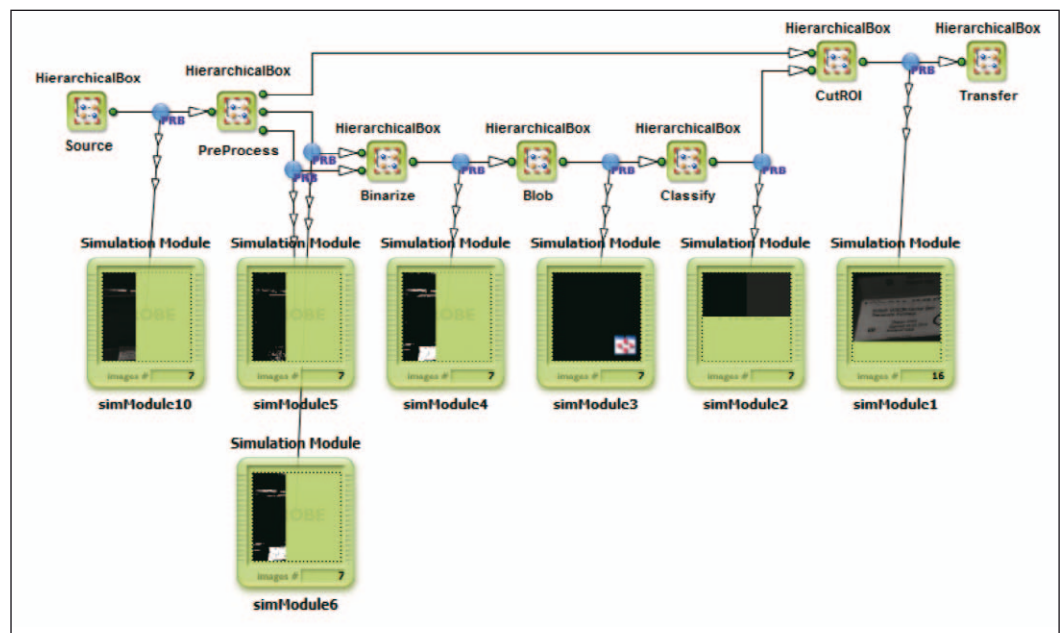
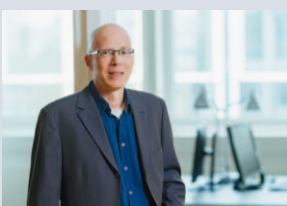


Bild 2: High-Level-Simulation in Echtzeit intelligent segmentierter ROIs

Autor:



Martin Cassel, Redakteur bei der Silicon Software GmbH

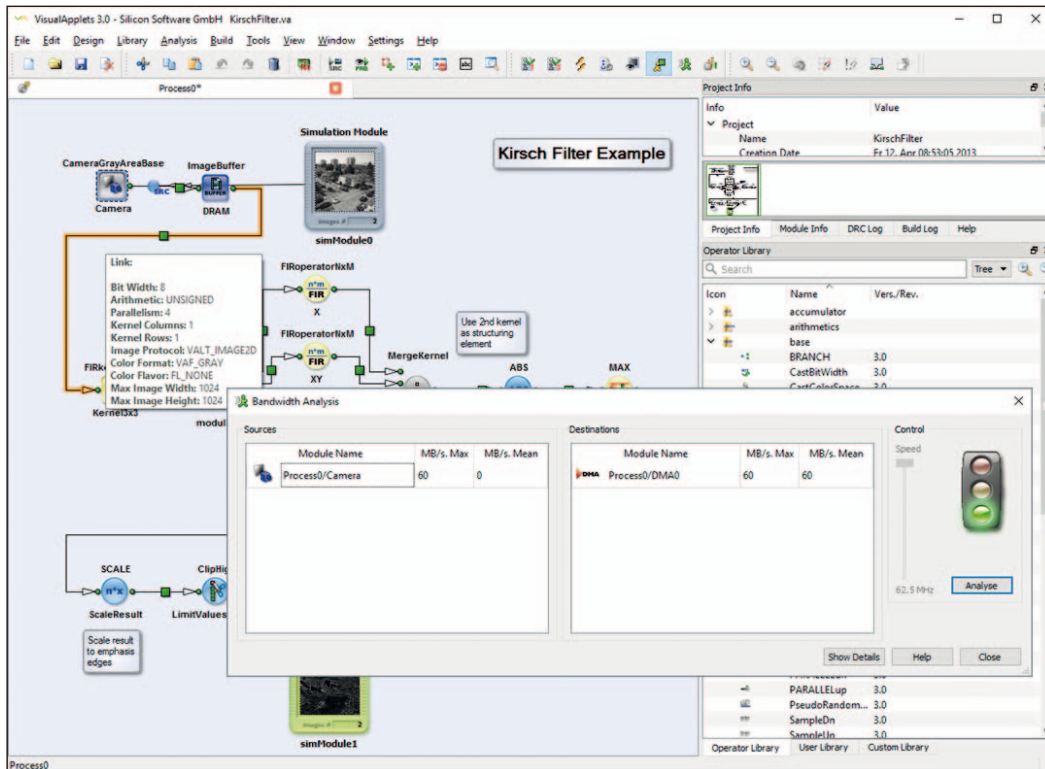


Bild 3: Bandbreitenanalyse im VisualApplets Design

metrien, Proportionen, Positionen, Strukturen und Muster wie auch deren Bewegung genau beschrieben. So lässt sich etwa die Geometrie über das Verhältnis von Konturlänge und Fläche oder eine Rundheit über die Hough-Transformation ermitteln und die Position über Lage einer Bounding Box oder dem Schwerpunkt (Center of Gravity) der Pixelfläche.

Reduktion durch smarte Vorverarbeitung

Durch die Blob-Analyse werden zusammenhängende Pixelbereiche als separate Formen und Objekte voneinander und vom Hintergrund getrennt. Dadurch findet ein großer Teil der Bildsegmentierung bereits während der Vorverarbeitung statt, noch bevor die Daten erstmals gespeichert werden. Darüber hinaus bietet die intelligente Auswahl eines Bildausschnitts per ROI (Region of Interest) die Möglichkeit, größere Bereiche des Bildes zu verwerfen und dadurch die Verarbeitungszeit für Algorithmen zu verkürzen – beispielsweise bei der Druckbildinspektion zur Detektion von Barcodes beim Kommissionieren von sich bewegenden Verpackungen über dynamische ROIs. Ein zusätzlicher Effekt kann hier eine Verringerung

der Rechneranzahl sein. Beim Einsatz mehrerer Kameras oder synchronisierter Aufnahmen kommt es darauf an, die Bilder sinnvoll miteinander zu verrechnen. Anwendungen wie High Dynamic Range (HDR) oder die 3D-Lasertriangulation reduzieren Daten bereits in der Vorverarbeitung wirkungsvoll. Besteht bei HDR die Datenreduktion durch Zusammenfassen mehrerer Helligkeitsstufen je Pixel aus mehreren Bildern zu einem einzigen Bild, können bei der Lasertriangulation aus Pixelbildern Höheninformationen zu 3D-Profilen zusammengefasst, die 3D-Profildaten errechnet und nur diese übertragen werden. Nur ein relevanter Punkt pro Sensorspalte der Laserlinie und nicht das Ausgangsbild ist hier entscheidend. Dies ermöglicht eine sehr hohe Bildwiederholrate, spart die Verarbeitung auf dem Host-Rechner und erlaubt einen kostengünstigen Systemaufbau beispielsweise mittels Anbindung über GigE Schnittstelle.

Einfache FPGA-Programmierung

Die Vorverarbeitung direkt in Kamera und Sensor ist notwendig, wenn Kameraschnittstellen wie GigE oder die Rechnerschnittstelle nur begrenzte Datenmengen

übertragen. Sie ist außerdem zwingend für eingebettete Systeme einer Produktionskette, die mit geringerer Rechenleistung dazu in der Lage sein sollen, Daten wie Produkt-, Qualitäts- und Prozessinformationen in Echtzeit auszuwerten, zur weiteren Verwendung aufzubereiten und die Ergebnisse zurückzumelden. Wird die Vorverarbeitung hingegen mit leistungsfähigeren programmierbaren Framegrabbern durchgeführt, lassen sich aufwändige Berechnungen wie etwa Filter-Operationen und Farbraum-Umwandlungen beschleunigen und so die CPU entlasten. Sowohl bei eingebetteten Systemen wie auch bei Framegrabbern können Anwender beliebige Algorithmen und Applikationen direkt auf den FPGAs (Field Programmable Gate Arrays) der Geräte programmieren, beispielsweise mit einer einfach zu bedienenden grafischen Entwicklungsumgebung wie VisualApplets.

Die Entwicklungsumgebung

ermöglicht, anwendungsspezifische Bildvorverarbeitungsaufgaben direkt im Framegrabber oder Kamera-FPGA umzusetzen und erübrigt dessen aufwändige und teure VHDL-Programmierung durch Hardware-Spezia-

listen. Davon profitieren Kamera- und Sensorhersteller sowie Endanwender, die nicht über das entsprechende Know-how zur FPGA-Programmierung und Kameraentwicklung verfügen. Auf geänderte Applikationsanforderungen können sie flexibel reagieren, ohne dass eine komplett neue Kamera entwickelt werden muss. Für die Programmierung von Kamera-FPGAs mit VisualApplets wird ein dynamischer IP-Core im FPGA als Kompatibilitätsschicht mittels VisualApplets Embedded implementiert. Dies erlaubt es Herstellern von Bildverarbeitungsgeräten, nach der einmaligen Implementierung beliebig viele Anwendungen zu entwickeln, diese auf weitere Geräte zu übertragen und ihren Kunden die Entwicklung eigener Anwendungen zu eröffnen. Werden zusätzlich programmierbare Framegrabber eingesetzt, lassen sich weitere aufwändigere Vorverarbeitungen durchführen und Bild-daten zusätzlich reduzieren.

Effiziente Verarbeitung durch Datenkompression

Zur Bildvorverarbeitung gehört außerdem die schnelle Datenkompression für Bilddaten. Die Komprimierung hat neben der Datenreduzierung den Vorteil, mit den komprimierten Bilddaten weitere Verarbeitungsschritte durchzuführen und dadurch manche Bildverarbeitungsoperatoren effizient zu verwenden, beispielsweise bei der Run-length Codierung von Binärbildern. Bei der Kompression ist es anschließend möglich, per Dekomprimierung das Originalbild eindeutig und fehlerfrei zu rekonstruieren.

Bei der abschließenden Übertragung zum Auswertungsrechner sollten die Daten schon so weit reduziert sein, dass keine Bandbreiten- oder Speicherplatzprobleme entstehen. Die Eignung einzelner Verfahren entlang der Bildverarbeitungskette hängt stark vom jeweiligen Anwendungsfall und der anschließenden Weiterverarbeitung ab. Dementsprechend müssen die passenden Algorithmen für die Verarbeitung der Bilddatei sorgfältig ausgewählt und implementiert werden.

■ Silicon Software GmbH
<https://silicon.software>