Entwicklung von Embedded Vision Systemen

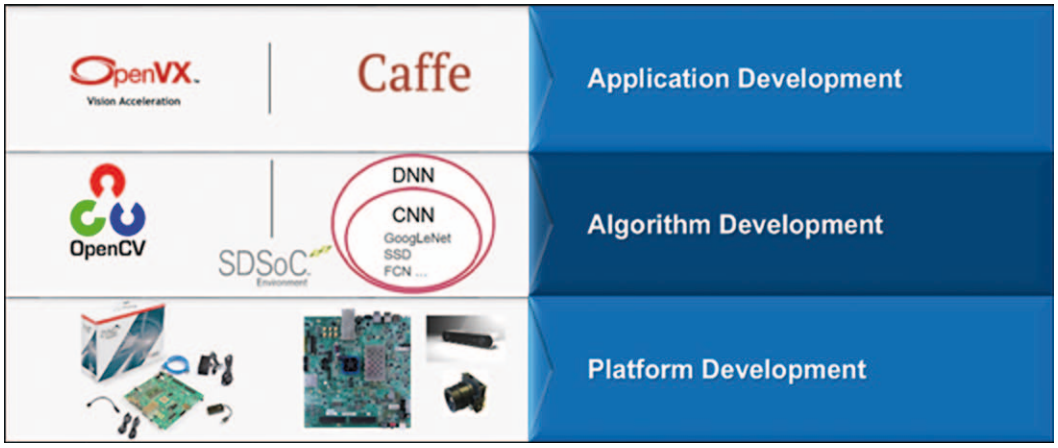

Bild 1: Der reVISION Stack

Embedded Vision Systeme finden sich heute in zahlreichen Industrien und Applikationen, von ADAS und digital gestützter Robotik bis zur medizinischen Bildgebung und Augmented Reality. Die breite Akzeptanz von Embedded Vision über viele Marktsegmente hinweg involviert tief greifende Änderungen in der Effizienz der Produktion und der Geschäftsmodelle. In den meisten dieser Anwendungen ist die Pipeline zur Bildbearbeitung sehr ähnlich aufgebaut. Diese Downstream Pipeline umfasst Funktionalitäten wie die Schnittstellen für Sensor/Kamera, sowie die Umsetzung des Bildes in geeignete Formate zur weiteren Bearbeitung.

Gebräuchliche Algorithmen in diesem Downstream Processing sind Farbrekonstruktion (Bayer-Filter) und Farbraum-Konversion, nebst Algorithmen zur Rauschunterdrückung. Es sind diese applikationsspezifischen Algorithmen, in denen die Unterschiede zwischen den genannten Anwendungen sichtbar werden. Deren Implementierung muss der Entwickler also einen signifikanten Zeit- und Arbeitsaufwand widmen.

Oft sind die für diese Applikationen erforderlichen Algorithmen recht komplex in ihrer Implementierung, weil sie Verfahren wie Objekterkennung und Klassifizierung verwenden, daneben spezifische Filterfunktionen und Berechnungsmethoden. Mehr und mehr werden diese Applikations-bezogenen Algorithmen auf der Basis von Open-Source Frameworks wie OpenCV und Caffe entwickelt. Der Einsatz dieser Open-Source Frameworks erlaubt dem Embedded-Vision-Entwickler die volle Konzentration auf seine eigentliche Arbeit, nämlich die Implementierung des geeigneten Algorithmus. Die Verwendung von vordefinierten Funktionen und der durch sie bereitgestellten IP umgeht die Notwendigkeit, jedes Mal von Grund auf neu zu beginnen. Das führt zu einer wesentlichen Abkürzung der Entwicklungszeit. Abhängig von der vorliegenden Applikation besteht die Herausforderung für den Entwickler jedoch nicht nur darin, wie er die dafür vorgesehenen Algorithmen angemessen implementiert. Der Embedded-Vision-Entwickler muss auch die Anforderungen an die Applikation und deren Einsatzumgebung berücksichtigen und dabei die gängigen Trends der Marktentwicklung im Auge behalten.

Autoren:

Nick Ni und Adam Taylor

Herausforderungen und Trends

Zu diesen Herausforderungen und Trends zählt auch die Ausführung der Bildbearbeitung und Entscheidungsfindung auf dem technologisch neuesten Stand, da Embedded-Vision-Applikationen zunehmend autonom und ohne Unterstützung aus der Cloud operieren müssen. Ein Beispiel dafür ist die Vision-gestützte Robotik. Sie basiert auf der Verarbeitung von Informationen aus Sensoren und leitet daraus die entsprechenden Aktionen zur Navigation innerhalb ihrer funktionalen Umgebung ab.

Viele Applikationen implementieren außerdem eine Sensorfusion, mit der Zusammenführung mehrerer verschiedener Sensor-Modalitäten zur verbesserten Wahrnehmung ihrer Umgebung und zur Unterstützung der Entscheidungsfindung. Alles das bedingt höhere Anforderungen an die Verarbeitung. Mit der schnellen Evolution der Sensoren und der Algorithmen zur Bildbearbeitung müssen sich die Systeme jederzeit nach den geltenden Vorgaben der Produkt-Roadmap aktualisieren lassen. Die wachsende Bedeutung von autonomen und entfernten Applikationen bringt außerdem die Herausforderungen der Effizienz in der Leistungsaufnahme und der Sicherheit, um jeden Versuch nicht autorisierter Eingriffe und Modifizierungen zu unterbinden.

Implementierungssysteme

Um dieser Problemlage gerecht zu werden, nutzen die Entwickler vorteilhaft das Xilinx All Programmable System on Chip (SoC) und das Multi Processor System on Chip (MPSoC) aus den Zynq-7000- and Zynq UltraScale+ MPSoC-Familien zur Implementierung ihrer Lösungen. Diese Bausteine enthalten High-Performance Prozessoren, die eng mit programmierbarer Logik gekoppelt sind. Damit können die Embedded-Vision Entwickler ihre Lösungen optimieren.

Der Einsatz der Zynq SoCs oder der Zynq UltraScale+ MPSoCs ermöglichen dem Entwickler auch die Ausschöpfung aller Vorteile der Any-to-Any Konnektivität, die mit dem Einsatz der programmierbaren Logik einhergeht. In der programmmierbaren Logik kann man auch die Pipeline(s) zur Bildbearbeitung implementieren, was wegen der parallelen Verarbeitung einen erheblichen Zuwachs an Performance realisiert. Der Einsatz der programmierbaren Logik erhöht die System-Performance, die Konnektivität und die Performance pro Watt bezogen auf die Leistungsaufnahme und liefert damit eine insgesamt effizientere Systemlösung.

Die Prozessorkerne lassen sich auch für die höheren Funktionalitäten der Applikation nutzen, etwa für die Entscheidungsfindung auf der Basis der vorliegenden Information und der Kommunikation zwischen den Systemen und der Cloud.

Sicherheitsanforderungen

Um die Sicherheitsanforderungen zu adressieren, wie sie für autonome und entfernte Applikationen typisch sind, bieten beide Bausteinfamilien eine sichere Umgebung mit verschlüsseltem sicherem Booten und der ARM Trust Zone Technologie innerhalb des Prozessors, und der Fähigkeit zur Implementierung von Anti-Tamper Funktionalität. Der Einsatz der Zynq-7000- und Zynq UltraScale+ MPSoC-Bausteine bietet dem Embedded-Vision-Entwickler einen signifikanten Spielraum zur Berücksichtigung der genannten Herausforderungen und Trends. Diese Fähigkeiten erfordern ein geeignetes Ökosystem der Entwicklung, das es dem Entwickler erlaubt, nicht nur die Vorteile der Bausteine zu nutzen, sondern weiterhin die gebräuchlichen Frameworks einzusetzen. An diesem Punkt setzt der reVISION Stack an.

Der reVISION Stack

Der reVISION Stack wurde entwickelt, um dem Embedded-Vision Entwickler die Lösung der vier zentralen Herausforderungen zu ermöglichen, wie sie im Bereich der Embedded-Vision Anwendungen gegeben sind. Diese Herausforderungen lassen sich wie folgt zusammenfassen: Ansprechempfindlichkeit, Rekonfigurierbarkeit, Konnektivität und Software-Definition.

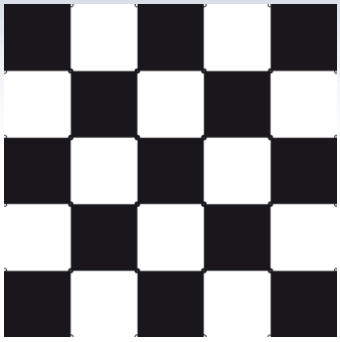

Bild 2: Beschleunigte OpenCV Harris Corner Detektion

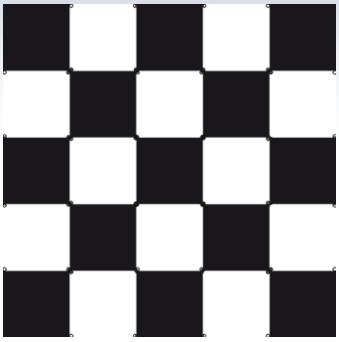

Bild 3: Traditionelle OpenCV-Implementierung

Im Hinblick auf diese vier wichtigen Trendsetter kombiniert der reVISION Stack eine Vielzahl von Ressourcen zur Plattform-, Applikations- und Algorithmus-Entwicklung. Dazu ist der Stack, wie Bild 1 zeigt, in drei unterschiedliche Ebenen gegliedert:

1. Platform Layer

Dies ist die niedrigste Ebene des Stack, auf der die anderen Ebenen aufsetzen. Sie ermöglicht sowohl eine Hardware-Definition für die Konfiguration des Zynq-7000 / Zynq UltraScale+ MPSoC, als auch die Software-Definition über ein spezifisch angepasstes Betriebssystem zur Unterstützung der Hardware-Definition. Die Hardware-Definition kann die Konfiguration eines Entwicklungs-Boards oder auch eines produktionsreifen Boards, etwa eines System-on-Module, definieren. Über diese Hardware-Definition werden auch der Sensor und die Systemschnittstellen definiert. Die Hardware-Plattform wird unter Einsatz von Vivado HLX erfasst. Sie kann IP-Blocks von Xilinx und auch von Drittanbietern nutzen, zusammen mit dem Einsatz der High-Level-Synthese zur Erstellung von Spezial-IP. Dieser Layer stellt auch Software-Treiber für IP-Module und, falls erforderlich, auch eine aktualisierte PetaLinux Konfiguration bereit, um die Software-definierte Umgebung auf der höheren Ebene zu unterstützen.

2. Mittlere Ebene des Stack

Sie wird als Algorithmus-Layer bezeichnet. Die Entwicklung auf dieser Ebene findet innerhalb des Eclipse-basierten SDSoC Environment statt. Das SDSoC stellt einen Compiler zur Systemoptimierung bereit, der die Entwicklung unter Einsatz einer Software-definierten Umgebung ermöglicht. Kritisch ist hier, bei der Entwicklung der geforderten Software-Algorithmen, dass die Identifizierung und Beseitigung von Performance-Engpässen durch die Beschleunigung der Funktionen in der programmierbaren Logik möglich ist. Für den Benutzer verläuft dieser Prozess absolut reibungslos, weil er die High-Level-Synthese mit einem Konnektivitäts-Framework kombiniert, um eine Funktion von der Software-Verarbeitung zur Implementierung in der programmierbaren Logik zu verschieben. Auf dieser Ebene wird OpenCV eingesetzt, um die Algorithmen zur Bildbearbeitung in der vorgesehenen Applikation zu implementieren. Zur Reduzierung der im Bildbearbeitungs-Algorithmus identifizierten Engpässe bietet reVISION eine Vielzahl von beschleunigungsfähigen OpenCV-Funktionen. Auf dieser Ebene findet sich auch die Unterstützung der gebräuchlichsten Bibliotheken für neurale Netzwerke wie AlexNet, GoogLeNet, SqueezeNet, SSD und FCN.

3. Oberer Layer

Der abschließende obere Layer ist die Ebene der Applikationsentwicklung. Hier finden die High-Level-Frameworks wie Caffe- und OpenVX-Anwendung, zur Vervollständigung der Applikation und zur Implementierung der Funktionalität der Entscheidungsfindung. Die Applikationen werden auf dieser Ebene unter Einsatz einer Eclipse-basierten Umgebung entwickelt, mit Ausrichtung auf die Prozessorkerne im Zynq-7000/Zynq UltraScale+ MPSoC.

Die vom reVISION Stack bereit gestellten Fähigkeiten liefern alle notwendigen Elemente zur Erstellung von High-Performance Imaging-Applikationen in einer Vielzahl von Anwendungen im Industrial Internet of Things, Vision-gestützter Robotik und anderem mehr.

Beschleunigung von OpenCV

Einer der interessantesten Aspekte des reVISION Stack ist die Möglichkeit zur Beschleunigung einer breiten Vielfalt von OpenCV-Funktionen innerhalb des Layers zur Algorithmus-Entwicklung. In diesem Layer lassen sich die beschleunigungsfähigen OpenCV-Funktionen in vier Haupt-Kategorien gruppieren:

1. Berechnung – Umfasst Funktionen wie die absolute Differenz zwischen zwei Frames, Pixel-bezogene Operationen wie Addition, Subtraktion und Multiplikation), sowie Gradient- und Integral-Operationen.
2. Eingangssignalverarbeitung – Bietet Unterstützung für Bittiefen-Konversion, Kanal-Operationen, Histogramm-Egalisierung, Remapping und Resizing.
3. Filterung – Unterstützt eine Fülle von Funktionen wie Sobel, kundenspezifische Faltung und Gauss-Filter.
4. Andere – Stellt eine Vielzahl weiterer Funktionen bereit, wie Canny/Fast/Harris-Kantendetektion, Schwellwertbildung und SVM- und HoG-Classifier.

Diese Funktionen kann der Entwickler zur Erstellung einer algorithmischen Pipeline in der programmierbaren Logik des gewählten Bausteins verwenden. Die Fähigkeit zu dieser Art der Logik-Implementierung trägt wesentlich zur höheren Performance der implementierten Algorithmen bei.

Natürlich kann man diese beschleunigungsfähigen OpenCV-Bibliotheken, da sie Software-definiert sind und die High-Level Synthese unterstützen, auch innerhalb des Vivado HLS-Tools einsetzen. Das ermöglicht die Erstellung und Nutzung von IP-Modulen auf dem Plattform-Layer, wenn die Hardware-Definition festgelegt ist.

Erkennung von Ecken

Ein in OpenCV oft eingesetzter Algorithmus ist die Implementierung der Harris Corner Detektion zur Erkennung von Ecken in einem Bild. Im reVISION Stack gibt es eine vordefinierte Funktion für die Harris Corner Detektion. Vergleicht man die Performance der mit reVISION beschleunigten Harris Corner Detektion mit einer normalen OpenCV-Implementierung, zeigen beide ein identisches Verhalten (Bilder 2 und 3). Allerdings erzielt man durch den Einsatz der per reVISION in die PL beschleunigten Harris Corner Funktion einen bedeutenden Gewinn an System-Performance. Damit ergibt sich eine Lösung mit wesentlich besserer Ansprechempfindlichkeit und Leistungseffizienz.

Performance optimieren

Im reVISION Stack kann der Entwickler, falls er die Beschleunigung von OpenCV-Funktionen anstrebt, sein Design innerhalb der programmierbaren Logik in Bezug auf die Ressourcen-Nutzung und Performance optimieren. Die dazu am häufigsten eingesetzte Methode basiert auf der Zahl der Pixel, die in jedem Taktzyklus verarbeitet werden. Für die meisten beschleunigten Funktionen kann man zwischen der Verarbeitung eines einzelnen Pixels oder von acht Pixeln wählen. Je mehr Pixel pro Takt verarbeitet werden, desto besser ist die geforderte Ressourcen-Nutzung bei geringerer Verarbeitungszeit. Die Verarbeitung nur eines Pixels pro Takt führt zu einer reduzierten Ausnutzung der Ressourcen, was sich in erhöhter Latenz bemerkbar macht. Diese Selektion der Anzahl der zu verarbeitenden Pixel pro Takt wird über den Funktionsaufruf konfiguriert. Das ist eine sehr einfache Methode zur geforderten Optimierung des Designs.

Nach dieser Optimierung der Performance seines Designs durch die beschleunigungsfähigen OpenCV-Bibliotheken kann der Embedded-Vision-Entwickler die höheren Ebenen der Applikation festlegen, indem er die vom Algorithmus- und Applikations-Layer des Stack bereitgestellten Fähigkeiten einsetzt.

Schlussfolgerung

Der Einsatz der All Programmable Zynq-7000- und der Zynq UltraScale+ MPSoC-Bausteine in Embedded-Vision Applikationen bringt zahlreiche Vorteile in Bezug auf die Flexibilität, der Performance, Security/Safety und der leistungseffizienten Verarbeitung. Die Entwicklung von Applikationen im reVISION Stack erlaubt die Verwendung von vielen gebräuchlichen Industrie-Standard Frameworks. Das ergibt kürzere Entwicklungszeiten und führt zur schnelleren Marktreife der Produkte.

■ *Xilinx, Ing.*
www.xilinx.com