Massive Parallelization of HOG-based Algorithms for Object Detection

Darius Malysiak

Die vorliegende Arbeit befasst sich mit der Frage, wie der HOG-Algorithmus auf massiv parallelen Architekturen beschleunigt werden kann. Sie führt das Konzept von sogenannten Hypersystemen ein, welche eine abstrakte Schnittstelle für tatsächliche Berechnungssysteme darstellen. Der Einsatz von Hypersystemen wird durch praktische Anwendungen motiviert; GPU-beschleunigtes Mean-Shift Clustering, verteiltes Training von kleinen neuronalen Netzen und verteilte Matrixmultiplikation. Auf der Basis dieses Ansatzes präsentiere ich ein Konzept zur Steigerung der Effizienz bestehender HOG Algorithmen; in Bezug auf Zuverlässigkeit und Recheneffizienz. Mein Konzept erlaubt es HOG Algorithmen effizient in heterogenen Rechensystemen zu verteilen. Des Weiteren präsentiere ich, ein für verteilte Berechnungen in heterogenen Cluster-Systemen ausgelegtes Software-Framework. Hierbei liegt ein besonderes Augenmerk auf Beowulf Clustern mit Multi-GPU-Knoten. Alle entwickelten Algorithmen werden gründlich analysiert; formal und durch praktische Auswertung.
Die Forschungsfragen, welche meine Forschung motivierten können in drei aufeinanderfolgende Fragen gegliedert werden. Meine anfängliche Motivation war es zu analysieren, wie der klassische HOG-Algorithmus in Bezug auf die Effizienz durch Parallelisierung verbessert werden kann. Dies führt zu der Frage, wie der lokal optimierte Ansatz für große Systemstrukturen verallgemeinert werden kann. Die dritte Frage war jene danach, wie diese Verallgemeinerung verwendet werden kann um komplexe Systeme zur Objekterkennung verbessern zu können. Insbesondere in Bezug auf bestehende HOG-basierte Architekturen.
Meine Arbeit schließe ich mit wichtigen Beobachtungen meiner Ansätze ab; Mit Hilfe eines massiv parallelisierten Mean-Shift Algorithmus zeige ich wie bereits GPU optimierte HOG Varianten formal verbessert werden können. Aus der generischen Formulierung des Algorithmus leite ich eine GPU-spezifische Variante ab. Diese adressiert das Problem der GPU Unter- und Überauslastung in Bezug auf das Datenvolumen. Der allgemeine und naive Ansatz zur Parallelisierung resultiert, im Vergleich zu CPU Rechenzeiten, in einer Verringerung der Berechnungszeit um den Faktor ~17. Mein Ansatz erhöht diesen Faktor auf ~176. Um solche Phänomene allgemein zu vermeiden entwickelte ich das Konzept von vPPUs, eine Meta-Ebene für bestehende SIMD-Architekturen.
Im Hinblick auf die zweite Forschungsfrage verallgemeinere ich den vPPU Ansatz für beliebige Berechnungssysteme, dies resultiert im Konzept der Hypersysteme. Wie zuvor motiviere ich dessen Verwendung mit einem anderen Beispiel aus der Praxis; Training von kleinen neuronalen Netzen auf massiv parallelen Architekturen. Ich zeige, dass mein Ansatz, im Vergleich zu kanonischen Ansätzen, eine Reduktion der Rechenzeit um den Faktor 2.37 bewirkt. Da die allgemeine Herausforderung beim Training neuronaler Netze hauptsächlich in der effizienten Matrixmultiplikation liegt, analysiere ich die klassische Matrix-Multiplikation im Lichte meines Ansatzes. Weiterhin diskutiere die Lastverteilung in Beowulf Cluster-Systemen mit einem speziellen Fokus auf Multi-GPU Unterstützung. Meine Analyse zeigt, dass die Effizienz der Matrixmultiplikation signifikant gesteigert werden kann indem intrinsische Hard- und Software Eigenschaften ausgenutzt werden.
Mit diesen Ergebnissen diskutiere ich den HOG-Algorithmus erneut. Ich stelle ein Konzept vor, welches so genannte Kachel-Bilder verwendet. Diesen liegt der gleiche Kerngedanke wie vPPUs zugrunde. Mein Ziel ist es die GPU-Auslastung zu erhöhen, ohne den bestehenden Algorithmus neu zu gestalten. Wie ich zeige, kann dies mit einer gleichzeitigen Reduktion der Berechnungszeit von bis zu 2.88 erreicht werden. Des Weiteren erläutere ich ein Konzept zur effizienten Verteilung der Objekterkennung in einem heterogenen Cluster-System. Ich identifiziere die Parameter für die Echtzeit-Grenzen und zeige, dass ein Ethernet-basiertes System sie leicht erfüllen können. Die Umsetzung des vorhandenen Algorithmus bleibt unverändert.
Eine sehr häufige Anwendung für die Objekterkennung ist die Videoüberwachung, in dem ein System viele parallele Videoströme verarbeiten muss. In diesem Kontext erläutere ich ebenfalls, wie ein vorhandenes HOG-basiertes System für diesen Anwendungsfall verbessert werden kann; mein entwickelter Ansatz macht es möglich, dass die Detektionsgüte erhöht und die Berechnungszeit verringert wird.