SCHEDULING AND CHARACTERIZATION OF COMPUTER VISIONWORKLOADS ON HETEROGENEOUS SYSTEMS

DIKSHA MOOLCHANDANI

AMAR NATH AND SHASHI KHOSLA SCHOOL OF INFORMATION TECHNOLOGY

INDIAN INSTITUTE OF TECHNOLOGY DELHI

MARCH 2022

SCHEDULING AND CHARACTERIZATION OF COMPUTER VISIONWORKLOADS ON HETEROGENEOUS SYSTEMS

by

DIKSHA MOOLCHANDANI

Amar Nath and Shashi Khosla School of Information Technology

Submitted

in fulfilment of the requirements of the degree of Doctor of Philosophy

to the

INDIAN INSTITUTE OF TECHNOLOGY DELHI

MARCH 2022

Certificate

This is to certify that the thesis titled **SCHEDULING AND CHARACTERIZATION OF COMPUTER VISION WORKLOADS ON HETEROGENEOUS SYSTEMS** being submitted by **Ms. DIKSHA MOOLCHANDANI** for the award of **Doctor of Philosophy** in Information Technology is a record of bonafide work carried out by her under my guidance and supervision at the Amar Nath and Shashi Khosla School of Information Technology, Indian Institute of Technology Delhi. The work presented in this thesis has not been submitted elsewhere, either in part or full, for the award of any other degree or diploma.

Smruti Ranjan Sarangi
Professor
Dept. of Computer Science & Engg.
Indian Institute of Technology Delhi.

Anshul Kumar
Emeritus Professor
Dept. of Computer Science & Engg.
Indian Institute of Technology Delhi.

Acknowledgements

When I started counting my blessings, my whole life turned around.
- Willie Nelson
No duty is more urgent than giving thanks.
- James Allen
When eating fruit, remember the one who planted the tree.
- Vietnamese Proverb

My Ph.D. experience was a roller coaster ride with lots of ups and downs. Today when I look back, it was every bit worth it. It was a mind-blowing transformation of my personality, thought process, and nature. I believe there are more people instrumental in my growth than I remember. Nevertheless, I will make the best effort to acknowledge most of them.

Firstly, I am grateful to God. I had always believed that not even a leaf moves without His will, and whatever happens is always for our good (we always realize this a few years later). I experienced it during my Ph.D. when after a lot of difficult phases, I got **Prof. Smruti Sarangi** as my advisor. I thank him for taking me on from another lab. This thesis is a result of his firm belief in me. He chose not to give up on me and worked relentlessly to bring out the best in me. His invaluable pieces of advice not only shaped me as a researcher but also as a person. I thank him for guiding me when I was utterly lost and letting me explore when I was ready. He did his best to make me an independent researcher and made me identify my strengths. I thank him for making me understand that patience is the key to overcoming all the challenges. Now, when I look through my journey, I realize that His plan was perfect, and I could not have asked for a better advisor than Prof. Sarangi.

I also thank my co-advisor, **Prof. Anshul Kumar**. I took up a Ph.D. because he made me believe that I was capable of innovating. I thank him for believing in me even when I had not started research. I also thank **Prof. Kolin Paul** who was the first person in IIT to have believed in me and admitted me as a project associate in his lab, which enabled the cascading set of events that made me pursue a Ph.D. I thank my student research committee members: **Prof.**

Preeti R. Panda, Prof. M. Balakrishnan, and Prof. Shouribrata Chatterjee for providing valuable feedback on my research. I would also like to thank **Prof. José F. Martínez** for collaborating with me on my first paper and providing me the pointers to think critically. I thank **Prof. Manuel Mazzara** and **Prof. Ilya Afanasyev** for mentoring and hosting me during my internship at Innopolis University. I express my deepest gratitude to **Prof. Naveen Garg** and **Prof. Chetan Arora** for providing valuable feedback on my papers.

I express my heartfelt gratitude to Ms. Aditi, Ms. Vandana, Ms. Manju, Mr. Suresh, and Mr. Rajesh for their speedy assistance in all administrative matters. I started my research with a senior student, Rajshekar Kalayappan. I am thankful to him for teaching me a lot about computer architecture and architectural simulators. He had been instrumental in giving me a kick-start in research. I would also like to thank my best friend, Ashish Kumar. Despite his busy schedule, he always lent a listening ear to my problems and provided sound advice both professionally and personally. I thank him for his jokes and for always being there for me during my struggling graduate days. One of my very first friends in IIT was Sapna Bhardwaj. We first met by chance and then fortunately formed a team for the distributed computing course. I learned a lot from her during the course projects. Her sharp skills in learning any new computing language and debugging codes helped me in honing my skills. I am indebted to Sapna for being a great friend and teammate.

I found many more great friends in Priyanka Singla, Hameedah Sultan, Chrystle Lobo, Sandeep Kumar, Omais Shafi, Ismi Abidi, Shubhankar Singh, Janib Bashir, Nivedita Shrivastava, Shailja Pandey, Garima Modi, Chahat Bansal, and Ankita Raj. I am thankful to Priyanka for the thoughtful discussions that we had during our coffee breaks and for attending interesting courses with me. I am grateful to Hameedah for being ever-ready to talk to me during the pandemic and providing valuable pieces of advice at different points in my Ph.D. I thank Sandeep for motivating me throughout, from our first collaboration till I got a job, and making me believe in myself. I am grateful to Nivedita for being a great friend, teammate and staying awake with me countless nights so that we could finish the project even when her part was complete. I thank Janib and Shubhankar for having insightful discussions with me and guiding me about the plans after Ph.D. I am grateful to Shailja, Omais, Ismi, and Garima for keeping a regular tab on me during the pandemic. I thank Garima for taking care of me and bringing me homemade lunch so that I don't have to go to mess in the scorching sun. I was blessed to have Chahat, Ankita, and Chrystle in my life for cracking me up every time we met.

Apart from my lab friends, I found a great friend in my roommate, Dilpreet Kaur. We met in a computer architecture course in the first semester and decided to stay together from thereon. She is the best roommate I could have asked for, always lively and cheerful, kind-hearted, and

always ready to help me out in any situation. I would also like to thank all my interns for believing in me and providing me with learning opportunities. I especially thank Sudhanshu and Kishore for helping me with the parts of the papers. I also thank my friends from Innopolis University: Geesara Prathap, Swati Megha, Vivek Kumar, Jasna Jankovic, Maria Naumcheva, Inna Baskakova, and Andrey Sadovykh. They made my internship a memorable experience.

I would also like to thank my cousins in Delhi: Pavan, Preeti, Krunal, and Vraj. Their liveliness re-energized me. I thank DST for funding my research and IRD for releasing my stipend on time. This list is incomplete without mentioning the security guards of the Himadri hostel, Kailash hostel, and SIT building. All thanks to their efforts that I was able to walk through the campus without fear of dogs. I express my heartfelt thanks to the warden, caretaker, and mess staff for catering to my requests. I also thank the cleaning staff of SIT building and Himadri hostel for maintaining proper hygiene (Cleanliness is next to Godliness).

No words will ever be enough to thank my parents: Dr. Kishore Moolchandani and Mrs. Sonia Moolchandani, for their immense support and understanding throughout my life, especially through my graduate years. I can never compensate for their endless sacrifices in this lifetime. I thank them for instilling in me the value of education, time, discipline, perseverance, and hard work. I also thank my sister, Gunja Moolchandani. All through these years, she had been my box of happiness, lucky charm, my mood booster, my stressbuster, and my biggest inspiration. I cannot thank her enough for the sacrifices she made during the pandemic to make my life easier.

Diksha Moolchandani

Abstract

The coming decade will be the era of smart, intelligent systems; many of these will have computer vision algorithms at their core. Such systems include self-driving cars, UAVs, robots, mobile nodes with edge computing support, and AR/VR systems. Hence, accelerating computer vision workloads is a very important problem as of today and has begun to attract a considerable amount of attention in both academia and industry. For building such systems we need to focus on both the hardware and software aspects.

As of 2021, the community appears to have significantly migrated towards CNN (convolutional neural network) based algorithms that run on GPUs and custom accelerators including FPGAs. A significant proportion of state-of-the-art systems comprises highly optimized CNN-based algorithms running on bespoke accelerators and GPUs. This trend is justified because such design choices have proven their mettle against traditional algorithms that primarily relied on multicore processors for parallelization.

However, we argue that this trend is showing signs of saturation, does not hold all the time, and may not prevail in the future. We systematically analyze, deconstruct, and in some cases debunk such unipolar notions in this thesis. We, instead, offer a more nuanced argument and envision a system that is an amalgam of all kinds of hardware platforms and software algorithms that together provide the best performance or power efficiency – as evinced in our experiments. We first start with showing that for traditional vision algorithms, GPUs are not the best choice. We show that as the number of concurrent applications (conventional vision algorithms) running on GPUs increases, the performance drops due to increased contention at the shared resources. In contrast, the performance of a bag of these concurrent applications on multicore processors scales well with an increase in the number of concurrent applications due to well-developed contention management policies in multicores and the inherent nature of conventional vision algorithms. This work is extended to create an accurate predictor that predicts the performance (91% accuracy) and power (96% accuracy) of a set of concurrently running vision applications on a GPU given their CPU execution characteristics and the execution time/power of individual

workloads on a GPU.

This work establishes multicore CPUs as legitimate systems for executing conventional vision applications. Next, we generalize this idea and consider a system with diverse computing elements that include small cores, large cores, possibly GPUs and accelerators, and then propose a scheduling algorithm. Given the large number of choices and attendant trade-offs, existing approaches find it very hard to compute optimal thread execution schedules for such systems. We developed a fast game theory-based technique that gives us near-optimal schedules very quickly without sacrificing fairness. The approach yields a performance benefit of 17% over the nearest competing algorithm.

Subsequently, we survey and characterize the CNN-based accelerators for vision systems. We found that the energy efficiency of the accelerators designed for CNNs relies on the amount of data reuse, the sparsity in the data, and the ineffectuality in the bits used for representing the data. Then, we choose the best CNN-based stereo algorithm and the best traditional computational geometry-based algorithm and compare their depth estimation accuracies head-to-head for a self-driving task. The results are *similar*, there is no clear winner. However, if we start considering inclement weather such as rain or fog, then the results are mixed. In roughly a third of the cases, the CNN-based solution is much better and the reverse holds true for roughly another third of the cases. For the rest, the accuracies are similar. We thus design an ensemble predictor that predicts the best algorithm for a given scene. This helps us maximize the overall accuracy and reduce the error in depth estimation substantially. We achieve up to an 18% (6.25% average) improvement in accuracy over the best state-of-the-art algorithm with a negligible area overhead (0.003 mm^2) while respecting all real-time constraints. Our algorithm also yields a confidence of prediction that can be utilized by higher decision-making layers. Such experiments prove that CNNs are not always the best; they have robustness issues and don't generalize very well as compared to their counterparts that use computational geometry too.

By this point we establish that in a real system we need both CPUs as well as GPUs and accelerators; we also need CNN-based and conventional computational geometry-based algorithms to create an accurate and robust vision system. Next, we consider other knobs that characterize cyber-physical systems such as UAVs, which use vision algorithms for their navigation or as standalone applications. Here, apart from the right choice of the computer vision algorithm, there are many more parameters that are important such as collision-free path planning, power efficiency, and flight duration. We show that when we include all these parameters with the vision system, optimizing the full set of parameters for the entire system becomes a very complex problem. It is large, multi-dimensional, non-convex, and not amenable to solution in real-time. We use game theory again and replace this intractable optimization problem with a simpler

yet approximate game theory equivalent: a set of players with different payoffs and strategies. Now, a game theory solver (Gambit [1]) can very quickly ($10-100\times$ faster than solving the optimization problem) obtain approximate yet good-quality solutions to our optimization problem.

संक्षेप

आने वाला दशक स्मार्ट, इंटेलिजेंट सिस्टम का युग होगा; इनमें से कई के मूल में कंप्यूटर विज़न एल्गोरिदम होंगे। इस तरह के सिस्टम में सेल्फ-ड्राइविंग कार, यूएवी, रोबोट, एज कंप्यूटिंग सपोर्ट वाले मोबाइल नोड्स और एआर / वीआर सिस्टम शामिल हैं। इसलिए, कंप्यूटर विज़न वर्कलोड में तेजी लाना आज की एक बहुत ही महत्वपूर्ण समस्या है और इसने अकादमिक और उद्योग दोनों में काफी ध्यान आकर्षित करना शुरू कर दिया है। ऐसी प्रणालियों के निर्माण के लिए हमें हार्डवेयर और सॉफ्टवेयर दोनों पहलुओं पर ध्यान देने की आवश्यकता है।

२०२१ तक, समुदाय काफी हद तक सीएनएन (कोंवोलुशनल न्यूरल नेटवर्क) आधारित एल्गोरिदम की ओर माइग्रेट हो गया है जो जीपीयू और कस्टम त्वरक जैसे एफपीजीए पर चलते हैं। अत्याधुनिक प्रणालियों के एक महत्वपूर्ण अनुपात में अत्यधिक अनुकूलित सीएनएन-आधारित एल्गोरिदम शामिल हैं जो बीस्पोक त्वरक और जीपीयू पर चल रहे हैं। यह प्रवृत्ति उचित है क्योंकि इस तरह के डिजाइन विकल्पों ने पारंपरिक एल्गोरिदम के खिलाफ अपनी योग्यता साबित कर दी है जो मुख्य रूप से समानांतरकरण के लिए मल्टीकोर प्रोसेसर पर निर्भर थे।

हालांकि, हम तर्क देते हैं कि यह प्रवृत्ति संतृप्ति के संकेत दिखा रही है, हर समय नहीं रहती है, और भविष्य में प्रबल नहीं हो सकती है। हम इस थीसिस में व्यवस्थित रूप से विश्लेषण, विघटन, और कुछ मामलों में ऐसी एकध्रुवीय धारणाओं को खारिज करते हैं। इसके बजाय, हम एक अधिक सूक्ष्म तर्क प्रस्तुत करते हैं और एक ऐसी प्रणाली की कल्पना करते हैं जो सभी प्रकार के हार्डवेयर प्लेटफ़ॉर्म और सॉफ़्टवेयर एल्गोरिदम का एक मिश्रण है जो एक साथ सर्वोत्तम प्रदर्शन या शक्ति दक्षता प्रदान करते हैं - जैसा कि हमारे प्रयोगों में दिखाया गया है। हम सबसे पहले यह दिखाना शुरू करते हैं कि पारंपरिक विज़न एल्गोरिदम के लिए, जीपीयू सबसे अच्छा विकल्प नहीं है। हम दिखाते हैं कि जैसे-जैसे जीपीयू पर चलने वाले समवर्ती अनुप्रयोगों (पारंपरिक दृष्टि एल्गोरिदम) की संख्या बढ़ती है, साझा संसाधनों के बढ़ते विवाद के कारण प्रदर्शन गिर जाता है। इसके विपरीत, मल्टीकोर प्रोसेसर पर इन समवर्ती अनुप्रयोगों के एक बैग का प्रदर्शन समवर्ती अनुप्रयोगों की संख्या में वृद्धि के साथ अच्छी तरह से बढ़ता है।इसका कारण मल्टीकोर में अच्छी तरह से विकसित विवाद प्रबंधन नीतियाँ और पारंपरिक विज़न एल्गोरिदम की अंतर्निहित प्रकृति है। इस कार्य में हम एक सटीक भविष्यवक्ता बनाते हैं जो एक जीपीयू पर समवर्ती रूप से चल रहे विज़न अनुप्रयोगों के एक सेट के प्रदर्शन (९१% सटीकता) और शक्ति (९६% सटीकता) की भविष्यवाणी करता है, जो कि उनके सी पी यू निष्पादन विशेषताओं और व्यक्तिगत कार्यभार के जीपीयू पर निष्पादन समय पर आधारित है।

यह कार्य पारंपरिक विज़न अनुप्रयोगों को क्रियान्वित करने के लिए वैध सिस्टम के रूप में मल्टीकोर सीपीयू को स्थापित करता है। इसके बाद, हम इस विचार को सामान्य बनाते हैं और विविध कंप्यूटिंग तत्वों वाली एक प्रणाली पर विचार करते हैं जिसमें छोटे कोर, बड़े कोर, संभवतः जीपीयू और त्वरक शामिल होते हैं, और फिर एक शेड्यूलिंग एल्गोरिदम का प्रस्ताव करते हैं। बड़ी संख्या में विकल्पों और परिचर ट्रेड-ऑफ को देखते हुए, मौजूदा दृष्टिकोणों को ऐसी प्रणालियों के लिए इष्टतम थ्रेड निष्पादन शेड्यूल की गणना करना बहुत कठिन लगता है। हमने एक तेज़ गेम थ्योरी-आधारित तकनीक विकसित की है जो हमें निष्पक्षता का त्याग किए बिना बहुत जल्दी-इष्टतम कार्यक्रम देती है। हमारा दृष्टिकोण १७% से अधिक का प्रदर्शन लाभ देता है निकटतम प्रतिस्पर्धी एल्गोरिथ्म पर।

इसके बाद, हम दृष्टि प्रणालियों के लिए सीएनएन-आधारित त्वरक का सर्वेक्षण और लक्षण वर्णन करते हैं। हमने पाया कि सीएनएन के लिए डिज़ाइन किए गए त्वरक की ऊर्जा दक्षता निर्भर करती है निम्नलिखित बातों पर – डेटा के पुन: उपयोग की मात्रा, डेटा में विरलता, और डेटा का प्रतिनिधित्व करने के लिए उपयोग किए जाने वाले बिट्स में अप्रभावीता। फिर, हम सबसे अच्छा सीएनएन-आधारित स्टीरियो एल्गोरिदम और सबसे अच्छा पारंपरिक कम्प्यूटेशनल ज्यामिति-आधारित एल्गोरिदम चुनते हैं और सेल्फ-ड्राइविंग कार्य के लिए उनकी गहराई अनुमान सटीकता की तुलना करते हैं। परिणाम समान हैं, कोई स्पष्ट विजेता नहीं है। हालांकि, अगर हम बारिश या कोहरे जैसे खराब मौसम पर विचार करना शुरू करते हैं, तो परिणाम मिले-जुले होते हैं। मोटे तौर पर एक तिहाई मामलों में सीएनएन-आधारित समाधान बहुत बेहतर होता है और लगभग एक तिहाई मामलों के लिए रिवर्स सही होता है। बाकी के लिए, सटीकता समान हैं। इस प्रकार हम एक समूह भविष्यवक्ता डिजाइन करते हैं जो किसी दिए गए दृश्य के लिए सर्वश्रेष्ठ एल्गोरिथ्म की भविष्यवाणी करता है। यह हमें समग्र सटीकता को अधिकतम करने में मदद करता है और गहराई से आकलन में त्रुटि को काफी हद तक कम करता है। हम सभी वास्तविक समय की बाधाओं का सम्मान करते हुए एक नगण्य क्षेत्र ओवरहेड (०.००३ मिमी^२) के साथ सर्वश्रेष्ठ अत्याधुनिक एल्गोरिदम पर सटीकता में १८% (६.२५% औसत) सुधार प्राप्त करते हैं। हमारा एल्गोरिदम भविष्यवाणी का विश्वास भी पैदा करता है जिसका उपयोग उच्च निर्णय लेने वाली परतों द्वारा किया जा सकता है। ऐसे प्रयोग साबित करते हैं कि सीएनएन हमेशा सर्वश्रेष्ठ नहीं होते हैं; उनके पास मजबूती के मुद्दे हैं और वे अपने समकक्षों की तुलना में बहुत अच्छी तरह से सामान्यीकरण नहीं करते हैं जो कम्प्यूटेशनल ज्यामिति का भी उपयोग करते हैं।

इस बिंदु तक हम यह स्थापित करते हैं कि एक वास्तविक प्रणाली में हमें सीपीयू के साथ-साथ जीपीयू और त्वरक दोनों की आवश्यकता होती है; हमें सीएनएन-आधारित और पारंपरिक कम्प्यूटेशनल ज्यामिति-आधारित एल्गोरिदम की भी आवश्यकता है एक सटीक और मजबूत विज़न प्रणाली बनाने के लिए। इसके बाद, हम अन्य नॉब्स पर विचार करते हैं जो यूएवी जैसे साइबर-भौतिक प्रणालियों की विशेषता रखते हैं, जो अपने नेविगेशन के लिए विज़न एल्गोरिदम का उपयोग करते हैं। कंप्यूटर विज़न एल्गोरिथम के सही विकल्प के अलावा, कई और पैरामीटर हैं जो महत्वपूर्ण हैं जैसे कि टकराव मुक्त पथ योजना, बिजली दक्षता और उड़ान अवधि। हम दिखाते हैं कि जब हम इन सभी मापदंडों को विज़न सिस्टम के साथ शामिल करते हैं, तो पूरे सिस्टम के लिए मापदंडों के पूर्ण सेट को अनुकूलित करना एक बहुत ही जटिल समस्या बन जाती है। यह कई जगहों पर बड़ा, बहु-आयामी, गैर-उत्तल है, और वास्तविक समय में समाधान के लिए उत्तरदायी नहीं है। हम फिर से गेम थ्योरी का उपयोग करते हैं और इस कठिन अनुकूलन समस्या को एक सरल लेकिन अनुमानित गेम थ्योरी समकक्ष के साथ प्रतिस्थापित करते हैं: विभिन्न भुगतान और रणनीतियों वाले खिलाड़ियों का एक सेट। अब, एक गेम थ्योरी सॉल्वर बहुत जल्दी (ऑप्टिमाइज़ेशन समस्या को हल करने की तुलना में १०-१००× तेज़) हमारी ऑप्टिमाइज़ेशन समस्या के अनुमानित लेकिन अच्छी गुणवत्ता वाले समाधान प्राप्त कर सकते हैं।

Contents

List of Figures

List of Tables