MOKSLAS – LIETUVOS ATEITIS
SCIENCE – FUTURE OF LITHUANIA
ISSN 2029-2341 print / ISSN 2029-2252 online
http://www.mla.vgtu.lt

ELEKTRONIKA IR ELEKTROTECHNIKA
ELECTRONICS AND ELECTRICAL ENGINEERING

2014 6(2): 129–133

Vaizdų technologijos T 111
Image Technologies T 111

http://dx.doi.org/10.3846/mla.2014.16

CPU IR GPU (CUDA) PALYGINIMAS VYKDANT ŠABLONŲ ATITIKTIES ALGORITMĄ

Evaldas Borcovas[1], Gintautas Daunys[2]

Šiaulių universitetas
El. paštas: [1]evaldas.borcovas@gmail.com; [2]g.daunys@tf.su.lt

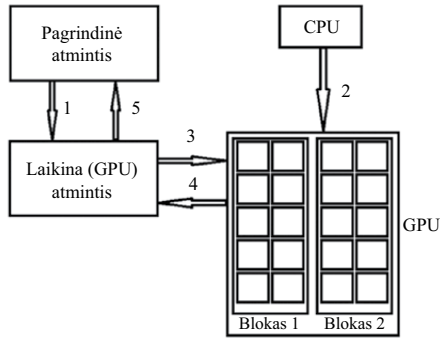
Santrauka. Vaizdų apdorojimas, kompiuterinė rega ir kiti sudėtingi algoritmai, apdorojantys optinę informaciją, naudoja didelius skaičiavimo išteklius. Dažnai šiuos algoritmus reikia realizuoti realiuoju laiku. Šį uždavinį išspręsti naudojant tik vieno CPU (angl. *Central processing unit*) pajėgumus yra sudėtinga. nVidia pasiūlyta CUDA (angl. *Compute unified device architecture*) technologija leidžia panaudoti GPU (angl. *Graphic processing unit*) išteklius. Tyrimui atlikti buvo pasirinkti du skirtingi CPU: Intel Pentium Dual-Core T4500 ir Intel Core I5 2500K, bei GPU: nVidia GeForce GT320M ir NVidia GeForce 560. Tyrime buvo panaudotos vaizdų apdorojimo bibliotekos: OpenCV 2.1 ir OpenCV 2.4. Tyrimui buvo pasirinktas šablonų atitikties algoritmas. Algoritmui realizuoti reikalingas analizuojamas vaizdas ir ieškomo objekto vaizdo šablonas. Tyrimo metu buvo keičiamas vaizdo ir šablono dydis bei stebima, kaip tai veikia algoritmo vykdymo trukmę ir vykdomų operacijų skaičių per sekundę. Iš gautų rezultatų galima teigti, kad apdorojant didelį duomenų kiekį GPU realizuoja algoritmą iki 24 kartų greičiau nei tik CPU. Dirbant su nedideliu duomenų kiekiu, skirtumas tarp CPU ir GPU yra minimalus. Lyginant skaičiavimus dviejuose GPU, pastebėta, kad skaičiavimų sparta yra tiesiogiai proporcinga GPU turimų branduolių kiekiui. Mūsų tyrimo atveju spartesniame GPU jų buvo 16 kartų daugiau, tad ir skaičiavimai vyko 16 kartų sparčiau.

Reikšminiai žodžiai: vaizdų apdorojimas, bendrosios paskirties GPU, šablonų atitiktis, CUDA technologija.

Įvadas

Siekiant įgyvendinti vaizdo apdorojimo, kompiuterinės regos ar kitus sudėtingus optinės informacijos apdorojimo algoritmus realiuoju laiku, reikalingi dideli skaičiavimo ištekliai. Naudojant vien tik CPU tai padaryti sudėtinga, o kartais neįmanoma. Siekiant padidinti algoritmų spartą, galima taikyti lygiagrečiojo skaičiavimo metodą. Norint pasiekti maksimalų skaičiavimų greitį, reikėtų naudoti GPU nVidia pasiūlytą technologiją CUDA. Ši technologija pritaikyta daugumai programavimo kalbų ir veikia su C, C++, FORTRAN, Python ir kt.

CUDA technologija leidžia kurti programas, kurioms vykdyti naudojami GPU ištekliai (1 pav.). Pirmiausia informacija turi būti perduota iš pagrindinės atminties į laikinąją, kuri naudojama atliekant skaičiavimus su GPU. CPU valdo visus procesus, vykstančius GPU, kuris turi tam tikrą branduolių skaičių, priklausantį nuo vaizdo plokštės modelio. Siekiant atlikti sudėtingesnius skaičiavimus, branduoliai gali būti sujunti į blokus, taip padidinant jų našumą. Pritaikydamas GPU tam tikram algoritmui, vartotojas gali keisti branduolių ir blokų konfigūraciją. Kiekvienas blokas gali atlikti skaičiavimus nepriklausomai nuo kitų blokų, t. y. atlikti skaičiavimus tuo pat metu. Taigi būdu didelis



1 pav. Informacijos apdorojimas GPU (CUDA) technologija
Fig. 1. Processing flow in CUDA technology

kiekis paprastų skaičiavimų gali būti atliekamas sparčiau nei tik naudojant CPU išteklius. Atlikus visus skaičiavimus, informacija grąžinama į pagrindinę atmintį ir skaičiavimai gali būti tęsiami, išjungiant GPU išteklius ar paliekant juos aktyvius (Fung, Mann 2008).

GPU privalumas tas, kad galima atlikti gausybę lygiagrečiųjų skaičiavimų, ko negali atlikti CPU. Blokų ir branduolių konfigūraciją galima adaptuoti pagal algoritmo sudėtingumą ar net keisti atskirų algoritmo blokų konfigūraciją.

GPU pagrindinis trūkumas tas, kad taktinis branduolių dažnis yra nedidelis, palyginti su CPU taktiniu dažniu. Informacijos perdavimas iš pagrindinės atminties į laikinąją ir atvirkščiai dėl papildomos informacijos perdavimo lėtina algoritmą.

Šio tyrimo tikslas – nustatyti, kaip GPU naudojimas gali paspartinti vaizdo apdorojamų algoritmų vykdymą. Tyrime buvo naudoti du skirtingo našumo GPU procesoriai. Juose buvo vykdomas šablonų atitikties algoritmas.

Panašūs tyrimai

Yamasaki *et al.* (2012) teigia, kad GPU naudojimas vaizdams apdoroti padidina algoritmų našumą iki 14 kartų, lyginant su technologija, kai naudojamas tik CPU. Autoriai tyrė vaizdų filtravimą ir jų pasiūlytas algoritmas nespėjo veikti realiuoju laiku. Vienas autorių pasiūlytų problemos sprendimo būdų – CUDA technologijos naudojimas. Pasiūlytas algoritmo vykdymas pagal šią technologiją paspartėjo šešis kartus. Tačiau jie taip pat bandė pritaikyti CUDA technologiją ir kitiems algoritmams. Našumas buvo padidintas 14 kartų. Išvadoje autoriai teigia, kad tolesniuose savo tyrimuose naudos CUDA technologiją (Yamasaki *et al.* 2012).

Sanchez, Rodriguez (2012) teigia, kad ne visi algoritmai, naudojant CUDA technologiją, veikia greičiau. Jų atliktuose tyrimuose buvo tiriamas CUDA naudojimas PCMF (angl. *Parallel Ccdf-based Median Filter*), CTMF (angl. *Constant time median filter*) ir Matlab funkcijoms vykdyti. Savo bandymams jie naudojo Jacket biblioteką, BVM (angl. *Branchless vector median*) ir PCMF. Iš autorių pateiktų rezultatų galima pastebėti, kad ne visi algoritmai veikia sparčiau su CUDA technologija. Tik PCMF veikė 2,7–2,9 karto sparčiau, lyginant su CPU realizacija. Visi kiti algoritmai veikė net lėčiau nei vykdant vien CPU. Tai dar kartą įrodo, kad ne visi algoritmai tinkami šiai technologijai. Rekomenduojama kiekvienam algoritmui sukonfigūruoti CPU branduolių ir blokų santykį (Sanchez, Rodriguez 2012).

Sharma *et al.* (2009) teigimu, GPU naudojimo našumas priklauso nuo apdorojamos informacijos kiekio. Kuo didesnis jos kiekis pateikiamas apdoroti, tuo labiau GPU technologija paspartina skaičiavimus. Jų tyrimuose buvo naudojamas Viola ir Jones algoritmas veidui vaizde aptikti. Tyrimo metu algoritmas buvo taikomas pateikiant skirtingo dydžio vaizdus. Pasak autorių, skaičiuojant su 320×320 taškų dydžio vaizdu, CPU skaičiavimo sparta buvo 7,4 kadro per sekundę, o naudojant GPU – net 90 kadrų per sekundę. Tai parodo, kad šiuo atveju GPU yra 12 kartų spartesnis už CPU. Tačiau padidinus vaizdo dydį iki 1280×960 taškų CPU sparta buvo 19 kadrų per sekundę, CPU – 0,5 kadro per sekundę. Šiuo atveju GPU yra 38 kartus greitesnis. Jų išvada buvo tokia, kad CUDA technologija yra naudinga, atliekant didelį kiekį nesudėtingų skaičiavimų. Našumas yra tiesiogiai proporcingas pateikiamų duomenų kiekiui (Sharma *et al.* 2009).

Lozano, Otsuka (2008) teigia, kad GPU technologija yra labai tinkama 3D veido sekimo algoritmui. Autoriai teigia, kad kaip ir Sharma naudojo Viola ir Jones algoritmą veidui aptikti 2D vaizde. Lozano ir Sharma išvada buvo tokia, kad GPU yra labai tinkamas dideliam kiekiui paprastų skaičiavimų atlikti. Autorių išvadose pateikta, kad jei vaizde yra tik vienas veidas, tuomet GPU veikia tris kartus sparčiau, esant šešiems veidams – devynis kartus (Lozano, Otsuka 2008).

Metodas

Tyrimui naudota techninė įranga:

- CPU I – Intel Pentium Dual-Core T4500 2,3 GHz procesorius su 4 GB RAM DDR3;
- CPU II – Intel Core I5 2500K 3,3 GHz procesorius su 4 GB RAM DDR3;
- GPU I – nVidia GeForce GT320M, 450 MHz;
- GPU II – nVidia GeForce GTX 560, 810 MHz.

Tyrimui naudota programinė įranga:

- OpenCV 2.1;
- OpenCV 2.4, naudojanti CUDA technologiją.

Pasirinktas šablonų atitikties algoritmas buvo realizuotas naudojant pirmiau išvardytas bibliotekas.

Pirmiausia buvo sugeneruoti atsitiktiniai skirtingų dydžių vaizdai. Generuojami vaizdai buvo atskirti į dvi grupes: pagrindinį vaizdą ir šabloną. Visuose vaizduose generuotas vaizdas buvo atsitiktinis triukšmas. Pagrindinio vaizdo dydis buvo keičiamas nuo 100×100 iki 1900×1900 taškų, didinant vaizdą vertikaliąja ir horizontaliąja kryptimi kas 200 vaizdo taškų. Šablono dydis buvo keičiamas nuo 10×10 iki 190×190 taškų, jo dydžio kitimo žingsnis horizontaliąja ir vertikaliąja kryptimis – 20 vaizdo taškų.

Vykdant tyrimą, buvo atliekami šie testai:

- Pagrindinio vaizdo dydžio keitimas;
- Šablono dydžio keitimas.

Matuojamos tokios charakteristikos:

- Algoritmo įvykdymo trukmė;

a) b)

2 pav. Vaizdai: a – pagrindinis; b – šablonas
Fig. 2. Images: a – main; b – template

− Atliktas skaičiavimų skaičius per sekundę.

Siekiant eliminuoti atsitiktines įvykių baigtis, visi bandymai buvo atlikti po 100 kartų.

Šablonų atitikties metodas įgyvendinamas slenkant šabloną per visą vaizdą. Vaizde ieškomas regionas, kuriame šablonas turi didžiausią koreliacijos koeficientą (Allusse *et al.* 2008). Tyrimui taikytas koreliacijos nustatymo metodas CV_TM_CCORR_NORMED. Koreliacija apskaičiuojama pagal tokią lygtį:

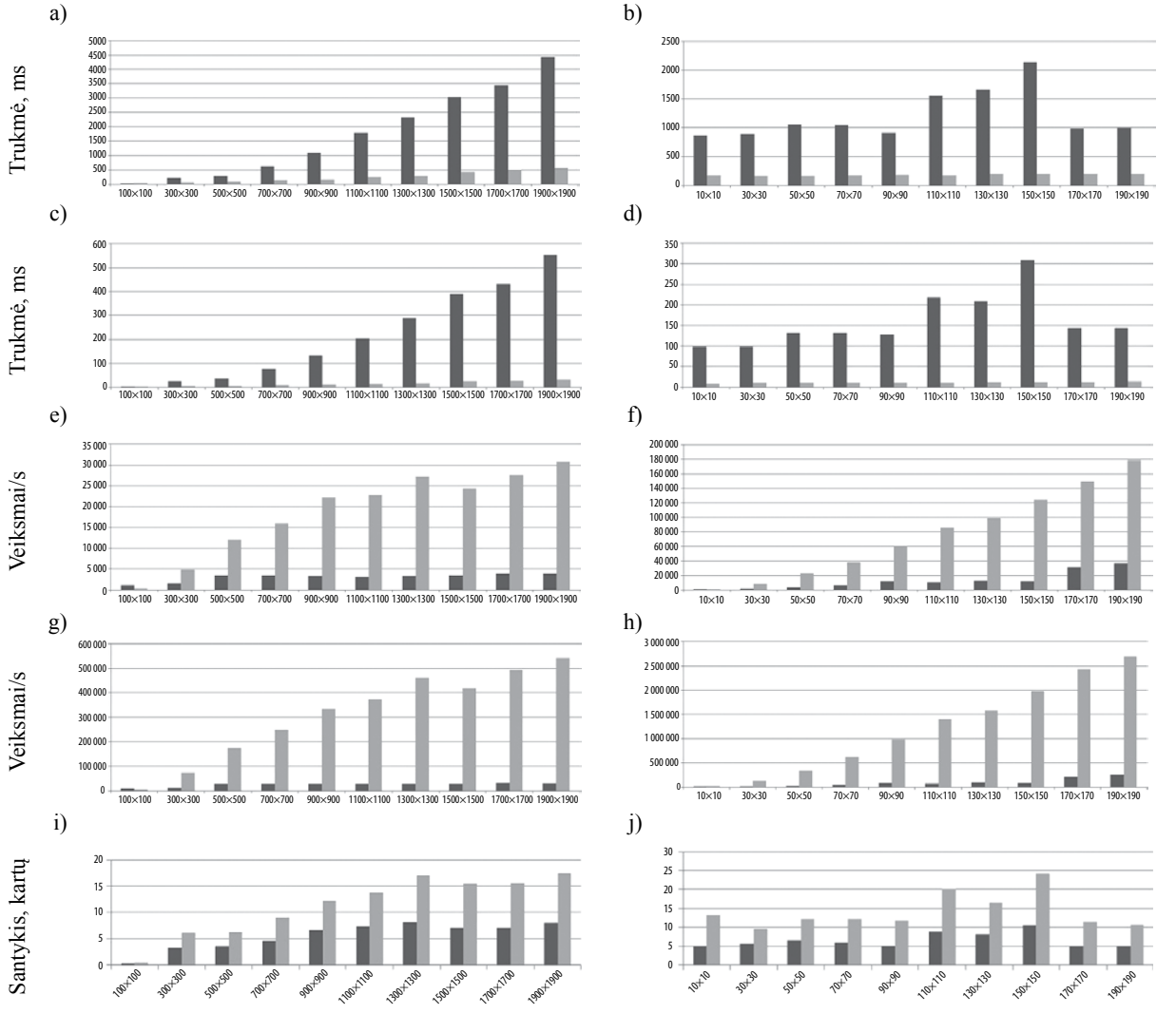
$$R(x,y)=\frac{\sum_{x',y'}\left(T(x',y')\cdot I'(x+x',y+y')\right)}{\sqrt{\sum_{x',y'}T(x',y')^2\cdot\sum_{x',y'}I(x+x',y+y')^2}}, \quad (1)$$

čia x – pagrindinis vaizdas; y – šablonas; R – koreliacijos koeficientas (Wilt 2012).

Rezultatai

Tyrimo rezultatai, keičiant pagrindinio vaizdo dydį, pavaizduoti 3 pav. kairiajame grafikų stulpelyje (a, c, e, g ir i dalyse). Rezultatai, keičiant šablono dydį, pavaizduoti 3 pav. dešiniajame grafikų stulpelyje (b, d, f, h ir j dalyse). Rezultatai pateikiami juostelių grafikais, čia juostelės iš kairės rodo CPU, o juostelės iš dešinės – GPU rezultatus.

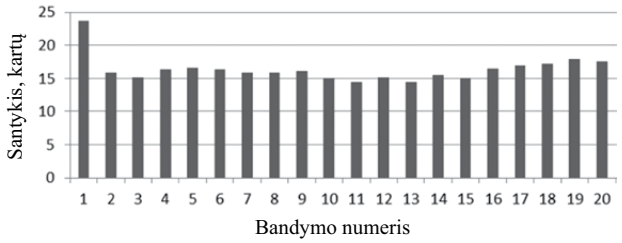
Skaičiavimo trukmės tarp CPU I ir GPU I palyginamos 3 pav. 1 eilutėje (a ir b dalyse), o CPU II ir GPU II – 2 eilutėje (c ir d dalyse). Veikimo sparta atitinkamai tarp CPU I ir GPU I palyginama 3 eilutėje (e ir f dalyse), o CPU II ir GPU II – 4 eilutėje (g ir h dalyse). Penktoje eilutėje pateikiami santykiai taip CPU ir GPU rezultatų (i ir j dalyse).



3 pav. CPU (juostelės iš kairės) ir GPU (juostelės iš dešinės) tyrimų rezultatai keičiant: pagrindinį vaizdą (kairė pusė), šabloną (dešinė pusė). Pateikiama: skaičiavimo trukmė (CPU I, GPU I – 1 eilutė, a ir b dalys; CPU II, GPU II – 2 eilutė, c ir d), veikimo sparta (CPU I, GPU I – 3 eilutė, e ir f dalys; CPU II, GPU II – 4 eilutė, g ir h), CPU ir GPU santykis (5 eilutė, i ir j)

Fig. 3. Investigation results using CPU (bars at left) and GPU (bars at right) and changing: main image (left column), template (right column). Presented are: calculation duration (CPU I, GPU I – 1 row, a, b parts; CPU II, GPU II – 2 row, c, d parts), calculation speed (CPU I, GPU I – 3 row, e, f parts; CPU II, GPU II – 4 row, g, h parts), CPU vs GPU ratio (5 row, i, j parts)

4 pav. pateikiamas apibendrintas dviejų tirtų vaizdo plokščių rezultatų palyginimas.



4 pav. Algoritmo įvykdymo GPU I ir GPU II įrenginiuose trukmių santykis (1–10 keičiant vaizdo dydį, 11–20 keičiant šablono dydį)

Fig. 4. Relationship between GPU I and GPU II execution times (1–10 changing main image, 11–20 changing template)

Diskusija

Keičiant pagrindinio vaizdo dydį pastebėta, kad kinta ir algoritmo įvykdymo trukmė. Vaizdo didinimas kartu didina atliekamų veiksmų skaičių. Algoritmo vykdymo sparta turėtų didėti tokiu pat dėsningumu skaičiuojant tiek CPU, tiek GPU įrenginiuose. Matyti, kad vykdymo greitis skiriasi (3 pav., a, c). Kuo daugiau informacijos apdorojama, tuo sparčiau vyksta skaičiavimai, tai lemia lygiagrečiųjų skaičiavimų vykdymas (3 pav., e, g). Skaičiuojant tik su CPU, gaunami priešingi rezultatai. Sparta mažėja labiau. Iš 3 pav. e, g dalių duomenų matyti, kad skaičiavimų skaičius, atliekamas per sekundę, praktiškai nekinta, algoritmą vykdant CPU, t. y. nėra taikomas lygiagrečiojo skaičiavimo metodas. Tai įrodo, kad didelis procesoriaus branduolio taktinis dažnis nėra reikalingas šiam algoritmui įgyvendinti. Kuo daugiau skaičiavimų atliekama tuo pačiu metu, tuo greičiau veikia algoritmas.

Kito bandymo metu, keičiant šablono dydį, pastebėta, kad algoritmo įvykdymo trukmės kitimo dėsningumai yra panašūs. Didinant šabloną, didėjo ir algoritmo įvykdymo trukmė. Tai matyti iš 3 pav. b ir d dalių. Išimtis pastebėta, kai šablono dydis pasiekė 150×150 taškų. Algoritmo įgyvendinimo sparta CPU įrenginyje padidėjo, tačiau stebint GPU, jokių pokyčių nepastebėta.

Iš gautų rezultatų matoma, kad pasirinkto algoritmo GPU efektyvumas padidėjo iki 24 kartų. Remiantis kitų mokslininkų darbais buvo galima tikėtis spartos padidėjimo nuo 2,7 iki 14 kartų. Šis koeficientas turėtų augti, didėjant algoritmo sudėtingumui ir algoritmo įvykdymo trukmių CPU ir GPU įrenginiuose santykiui.

Nustatyta, kad GPU II našumas yra 15–17 kartų didesnis už GPU I. Tai galima paaiškinti branduolių, esančių GPU, skaičiumi. GPU II turi 16 kartų daugiau vidinių branduolių.

Išvados

Atlikti bandymai su dviem nVidia GPU procesoriais parodė, kad šablonų atitikties algoritmas veikia iki 17 kartų sparčiau, keičiant pagrindinio vaizdo dydį, o keičiant šablono dydį – iki 24 kartų.

Algoritmo įvykdymo trukmė yra tiesiogiai proporcinga pagrindinio vaizdo dydžiui ir šablono dydžiui.

Padidinus šabloną iki 150×150 vaizdo taškų, CPU našumas padidėjo, tačiau, toliau didinant šabloną, jis vėl proporcingai mažėjo. Tai rodo, kad vaizdo ir šablono dydžių pasirinkimas daro poveikį skaičiavimo spartai.

GPU procesoriaus našumas yra tiesiogiai proporcingas branduolių skaičiui, esančiam jame.

Būsimuose darbuose tikslinga atlikti šiuos tyrimus:

1. Ištirti pasirinktą algoritmą daugiau nei vieną branduolį turinčiame CPU.
2. Ištirti kitus algoritmus ir įvertinti, kurie algoritmai veikia greičiau panaudojus CUDA technologiją.
3. Nustatyti optimalų blokų skaičių tiriamiems algoritmams.

Literatūra

Allusse, Y.; Horain, P.; Agarwal, A.; Saipriyadarshan, C. 2008. GpuCV: an opensource GPU-accelerated framework for image processing and computer vision, *Advances in Visual Computing* 5359: 430–439.

Fung, J.; Mann, S. 2008. Using graphics devices in reverse: GPU-based image processing and computer vision, in *2008 IEEE International Conference on Multimedia and Expo*, 23 June – 26 April, 2008, Hannover, 9–12.

Lozano, O. M.; Otsuka, K. 2008. Simultaneous and fast 3D tracking of multiple faces in video by GPU-Based stream processing, in *Proceedings of IEEE International Conference on Acoustics*, *Speech and Signal Processing*, 31 March – 04 April, 2008, Las Vegas, 713–716.

Sanchez, R. M.; Rodriguez, P. A. 2012. Bidimensional median filter for parallel computing architectures, in *Proceedings of IEEE International Conference on Acoustics*, *Speech and Signal Processing*, 1549–1552.

Sharma, B.; Thota, R.; Vydyanathan, N.; Kale, A. 2009. Towards a robust, real-time face processing system using CUDA-enabled GPUs, in *2009 International Conference on High Performance Computing (HiPC)*, 16–19 December, 2009, Kochi, 368–377. http://dx.doi.org/10.1109/HIPC.2009.5433189

Wilt, N. 2012. *CUDA handbook: a comprehensive guide to GPU programming*. ISBN-10: 0321809467.

Yamasaki, T.; Fujikawa, T.; Katto, J. 2012. Improving the performance of STIF using bilateral filter and its application to generic object recognition, in *Proceedings of IEEE International Conference on Acoustics*, *Speech and Signal Processing*, 945–948.

CPU AND GPU (CUDA) TEMPLATE MATCHING COMPARISON

E. Borcovas, G. Daunys

Abstract

Image processing, computer vision or other complicated optical information processing algorithms require large resources. It is often desired to execute algorithms in real time. It is hard to fulfill such requirements with single CPU processor. NVidia proposed CUDA technology enables programmer to use the GPU resources in the computer. Current research was made with Intel Pentium Dual-Core T4500 2.3 GHz processor with 4 GB RAM DDR3 (CPU I), NVidia GeForce GT320M CUDA compliable graphics card (GPU I) and Intel Core I5-2500K 3.3 GHz processor with 4 GB RAM DDR3 (CPU II), NVidia GeForce GTX 560 CUDA compatible graphic card (GPU II). Additional libraries as OpenCV 2.1 and OpenCV 2.4.0 CUDA compliable were used for the testing. Main test were made with standard function MatchTemplate from the OpenCV libraries. The algorithm uses a main image and a template. An influence of these factors was tested. Main image and template have been resized and the algorithm computing time and performance in Gtpix/s have been measured. According to the information obtained from the research GPU computing using the hardware mentioned earlier is till 24 times faster when it is processing a big amount of information. When the images are small the performance of CPU and GPU are not significantly different. The choice of the template size makes influence on calculating with CPU. Difference in the computing time between the GPUs can be explained by the number of cores which they have.

Keywords: image processing, GPGPU, template matching, CUDA.