

Detekcija pokretnih objekata u video sekvencama zasnovana na paralelnoj obradi u okviru jedinice za grafičku obradu

Dubravko Čulibrk, Dragan Letić i Vladimir Crnojević

Sadržaj — Za detekciju pokretnih objekata u sistemima za video nadzor najčešće se koristi segmentacija prednjeg plana. Kada je potrebno izvršiti detekciju pokretnih objekata, a da kretanje postoji i u samoj pozadini, neophodno je primeniti složene probabilističke modele pozadine. Korišćenje tih modela izuzetno je računski zahtevno i ograničava primene ovih metoda na sekvence sa rezolucijama daleko manjim od onih koje je moguće ostvariti najnovijim kamerama. Većinu probabilističkih algoritama za segmentaciju prednjeg plana moguće je ubrzati paralelnim procesiranjem informacija na nivou piksela. Dalja paralelizacija procesa na nivou paterna obeležja smeštenih u modelu pozadine za svaki piksel, moguća je primenom neuralnih mreža za izradu modela pozadine (Background Modelling Neural Networks – BNNs). U radu je opisana implementacija segmentacije prednjeg plana zasnovana na BNN-u, na NVIDIA™ grafičkom procesoru. Kako bi implementacija bila efikasnija, predloženo je nekoliko modifikacija originalnog algoritma. Eksperimentima je pokazano da je korišćenjem ovog pristupa moguće ubrzati procesiranje videa za red veličine.

Ključne reči — segmentacija prednjeg plana, model pozadine, neuralne mreže, paralelno procesiranje, procesiranje videa, video nadzor.

I. UVOD

DETEKCIJA pokretnih objekata predstavlja fundamentalni problem u kompjuterskoj viziji i ima ključnu ulogu u mnogim aplikacijama, a najčešće je korišćena u automatizovanom video nadzoru [1],[2] i [3], što je fokus i ovog rada.

Poslednjih 25 godina vršena su mnoga istraživanja na temu segmentacije pokretnih objekata. Pretpostavka je da se u video nadzoru koriste nepokretne kamere, pa se očekuje i da je pozadina stacionarna, te je moguće izgraditi adaptivni model pozadine koji može služiti kao osnova za segmentaciju. Specijalna kategorija problema su sekvence snimljene u prirodi u kojima postoje složena kretanja u pozadini (npr. njihanje drveća na vetru, talasanje vode, itd.) i promena iluminacije, gde je stacionarnost pozadine narušena. Kada se koriste ove sekvence najbolji objavljeni

algoritmi koriste multimodalnu procenu verovatnoće vrednosti piksela u pozadini, kao model pozadine i statistički test za određivanje verodostojnosti da piksel pripada procenjenoj raspodeli verovatnoće [1], [4], [2] i [3]. Nažalost, izgradnja i održavanje probabilističkih modela na ovaj način su računski zahtevni, pa je segmentacija u realnom vremenu moguća samo na video sekvencama manje rezolucije, tipično QCIF ili CIF [2] i [1]. Metode bazirane na kernelima su toliko kompleksne da [4] i [5] uopšte ne mogu vršiti obradu u realnom vremenu.

Jedinice za grafičku obradu su pogodne za paralelnu obradu (isti program se izvršava na mnogo podataka u paraleli) sa visokim aritmetičkim intenzitetom (odnos broja aritmetičkih operacija prema broju pristupa memoriji). Isti program se izvršava za svaki element, pa iz tog razloga nije neophodna sofisticirana kontrola toka. Iz istog razloga, kao i zbog visokog aritmetičkog intenziteta, efekat kašnjenja zbog pristupa memoriji se može relaksirati i bez upotrebe velike keš memorije [6]. Programabilnost modernih grafičkih procesora (*Graphics Processing Unit – GPU*) omogućava paralelnu implementaciju za veliki broj algoritama iz domena kompjuterske vizije [7].

Izrada modela pozadine primenom neuralnih mreža (*Background Modelling Neural Networks – BNN*) predstavlja metod za segmentaciju prednjeg plana baziran na kernelima, specijalno dizajniran za obradu u paraleli [1]. Ovo je trenutno jedini način za segmentaciju složenih sekvenci visoke rezolucije u realnom vremenu. U radu je opisana implementacija BNN pristupa na NVIDIA™ grafičkom procesoru koja omogućava obradu sekvenci standardne definicije (*Standard Definition – SD*) u realnom vremenu. Kako bi implementacija bila efikasnija, predloženo je nekoliko modifikacija originalnog algoritma. NVIDIA *Compute Unified Device Architecture – CUDA* [6] i OpenCV [8] su tehnologije korišćene za opisanu implementaciju. Rezultati su testirani na velikom broju sekvenci standardnih za ovaj domen.

U glavi 2 dat je kratak pregled objavljenih radova iz ove oblasti. U glavi 3 opisani su osnovni aspekti BNN pristupa. U glavi 4 nalaze se razmatranje CUDA ideje i detalji implementacije na jedinici za grafičku obradu. Glava 5 posvećena je prezentaciji i diskusiji rezultata. U poslednjoj, 6. glavi izložen je zaključak rada.

Dubravko Čulibrk, Fakultet tehničkih nauka, Univerzitet u Novom Sadu, Trg Dositeja Obradovića 6, 21000 Novi Sad, Srbija (e-mail: alef.tau@gmail.com).

Dragan Letić, Fakultet tehničkih nauka, Univerzitet u Novom Sadu, Trg Dositeja Obradovića 6, 21000 Novi Sad, Srbija (e-mail: leta1983@hotmail.com).

Vladimir Crnojević, Fakultet tehničkih nauka, Univerzitet u Novom Sadu, Trg Dositeja Obradovića 6, 21000 Novi Sad, Srbija (e-mail: crnojevic@uns.ac.rs).

II. PREGLED RELEVANTNE LITERATURE

Probabilističke metode su se pokazale kao najbolje za segmentaciju prednjeg plana. One vrše procenu statistike piksela koji pripadaju pozadini i koriste ih kako bi ih razlikovali od onih koji pripadaju prednjem planu [1], [3], [5] i [2]. Statistički modeli na bazi Gausove raspodele čiji parametri se rekursivno koriguju kako bi se ispratile postepene promene u pozadini, predloženi su u [11]. U skorije vreme, modeli na bazi Gausove raspodele, značajno su poboljšani korišćenjem mešavine Gausovih raspodela (*Mixture of Gaussians – MoG*) kao modela za gustinu raspodele verovatnoće vrednosti piksela. Više Gausovih raspodela (od 3-5) se koriste za procenu gustine raspodele verovatnoće [3]. MoG pristup Staufera i Grimsona [3] je najčešće korišćen pristup za segmentaciju. Proteklih godina, izvršeno je nekoliko unapređenja njihovog originalnog algoritma [12].

Modeli na bazi Gausove raspodele su parametarski u smislu da inkorporiraju pretpostavku o obliku gustine raspodele verovatnoće koju pokušavaju da procene. Ovo može dovesti do grube aproksimacije gustine raspodele verovatnoće što utiče na njihove performanse [2]. Iz ovog razloga je poslednjih godina posebna pažnja posvećena neparametarskim modelima. Neparametarska procena gustine na bazi kernela (jezgara), za potrebe segmentacije prednjeg plana i praćenje objekata u video nadzoru, predložena je u [4]. Autori su predstavili dobre kvalitativne rezultate predloženog sistema, ali procena segmentacije nije data kvantitativno, niti su poredili svoj sistem sa drugima. Ovaj princip je računski zahtevan jer broj kernela odgovara broju posmatranih vrednosti piksela. U 2003, Li *et al.* predložili su neparametarski metod za detekciju pokretnih objekata [2], koji predstavlja hibrid između probabilističkih i metodologija zasnovanih na NF filtrima. Inicijalni model pozadine koji koriste Li *et al.* jeste slika pozadine propuštena kroz NF filter. Autori koriste i sekundarni, probabilistički, model za vrednosti piksela detektovane kao prednji plan oduzimanjem trenutnog frejma od referentne slike pozadine. Njihov model predstavlja kvantizovan histogram vrednosti piksela koje se pojavljuju na lokaciji svakog piksela.

Pristup zasnovan na izgradnji modela pozadine primenom neuralne mreže, predložen je u [1]. Mreža predstavlja biološki verodostojnu implementaciju Bajesovog klasifikatora zasnovanu na neparametarskim estimatorima gustine verovatnoće zasnovane na kernelima. Težinski faktori mreže predstavljaju model pozadine koji se neprestano koriguje. Procene gustine raspodele sadrže fiksni broj kernela nepromenljive širine. Ovom metodologijom dobijaju se rezultati bolji od onih koji se dobijaju algoritmicima koje su predložili Li *et al.* i korišćenjem MoG-a sa 30 Gausovih raspodela. Korišćenjem paralelizma neuralnih mreža moguće je ubrzati algoritme za segmentaciju zasnovane na izgradnji modela pozadine na bazi jezgara.

Jedinice za grafičku obradu postaju sve interesantnije među ljudima koji se bave kompjuterskom vizijom [7], zbog velike razlike u rastu performansi u poređenju sa centralnim procesorima i *Field Programmable Gate Arrays – FPGA* iste generacije [13]. Jedna implementacija

segmentacije prednjeg plana na GPU-u je opisana u [14]. Pristup je namenjen za korišćenje u zatvorenom prostoru i nije pogodan za segmentaciju sekvenci sa složenim kretanjima u pozadini. Umesto korišćenja probabilističkog modela, autori izvode sofisticiranija obeležja za regione piksela i oslanjaju se na kriterijum kolinearnosti kako bi postigli segmentaciju.

U ovom radu je opisana implementacija jednog od savremenih probabilističkih modela, koji uspešno izlazi na kraj i sa sekvencama u kojima ima složenog kretanja u pozadini, na bazi neuronskih mreža za modelovanje pozadine (*engl. Background Modelling Neural Networks – BNN*)[1]. Izrada modela pozadine primenom neuralnih mreža se koristi kako bi se razvio klasifikator specijalno dizajniran za segmentaciju prednjeg plana u video sekvencama. Neuralna mreža uči bez nadzora (*eng. unsupervised*). Sama gradi statistički model dinamičkih promena vrednosti obeležja piksela. Naučeni model se koristi za klasifikaciju piksela u prednji plan ili pozadinu, za svaki frejm video sekvence. Više detalja vezanih za BNN pristup, njegove performanse i primenljivost moguće je naći u [1].

III. BNN-OVI NA JEDINICI ZA GRAFIČKU OBRADU

Nаша implementacija BNN-a na GPU-u izvršena je korišćenjem NVIDIA CUDA-e. CUDA je model za paralelno programiranje i softversko okruženje projektovano da bi se prevazišli izazovi prilikom razvoja aplikacija koje koriste paralelnu obradu ne bi li iskoristili snagu više jezgara u jedinici za grafičku obradu. Programeri upoznati sa programskim jezikom C vrlo brzo se upoznaju sa CUDA okruženjem [6]. Kompajlirani CUDA program može se izvršavati na proizvoljnom broju jezgara, samo sistem mora biti upoznat sa stvarnim brojem jezgara.

CUDA proširuje programski jezik C na taj način što programer definiše C funkcije, koje se nazivaju kerneli, koje, kada budu pozvane, se izvršavaju N puta u paraleli pomoću N različitih CUDA niti (*engl. thread*). Zbog ograničenog broja registara dostupnih svakom tredu, treba voditi računa da kôd kernela bude jednostavan i treba izbegavati kontrolu toka instrukcija. Niti su organizovane u blokove. Niti unutar jednog bloka mogu međusobno saradivati deljenjem podataka preko zajedničke memorije i sinhronišući njihova izvršavanja zbog koordinacije pristupa memoriji. Blokovi moraju biti projektovani tako da se mogu izvršavati nezavisno i u proizvoljnom redosledu. Korišćenjem velikog broja niti po bloku može se smanjiti kašnjenje koje se dobija prilikom pristupa memoriji i postiže se maksimalna opterećenost multiprocera u računске svrhe. Broj niti po bloku ograničen je mogućnostima hardvera.

Tipičan tok programa koji koristi CUDA-u uključuje pokretanje centralnog procesora host-a, koji prosleđuje ulazne podatke memoriji GPU-a i pobuđuje kernele. Kada kernel završi obradu, podaci se mogu prebaciti iz memorije GPU-a u memoriju host-a. Kako bi se izbeglo kašnjenje usled pristupa memoriji, podaci smešteni u memoriji GPU-a trebali bi biti pogodno poravnati i treba im se pristupati kao uzastopnim 32-bitnim rečima.

TABELA 1: VREME IZVRŠAVANJA ZA NVIDIA GEFORCE 8400M GS I 250 GTS.

| Test sekvenca | Rezolucija | Broj frejmova | A vreme (ms/frejmu) | | B vreme (ms/frejmu) | |
|---------------|----------------|---------------|---------------------|---------|---------------------|---------|
| | | | 8400M | 250 GTS | 8400M | 250 GTS |
| „Zavesa“ | 160x128 (QCIF) | 2960 | 33 | 0.011 | 54 | 16 |
| „Most“ | 384x288 | 2250 | 130 | 31 | 180 | 47 |
| „Tunel“ | 720x576(SD) | 9018 | N/A | 96 | 542 | 145 |

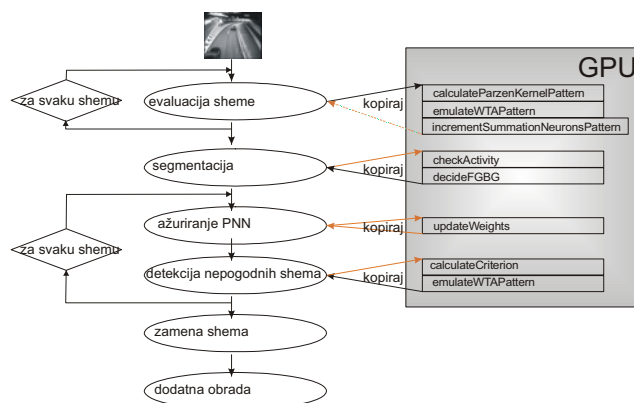
BNN pristup, opisan u glavi 3, omogućuje paralelizaciju izračunavanja na nivou jednog neurona sheme. Na Sl.1. prikazan je tok implementacije programa za jedan frejm iz sekvence. CUDA kerneli su izlistani sa desne strane slike u osenčenom pravougaoniku. Podrutine centralnog procesora date su sa leve strane. Frejmove video sekvence čita centralni procesor. Odziv svakog neurona sheme smeštenog u probabilističku neuralnu mrežu (*Probabilistic Neural Network – PNN*) procenjuje se zasebno na GPU-u, zbirno stanje neurona se inkrementira, a funkcija WTA mreže emulirana je poređenjem maksimalne vrednosti određene do tog trenutka sa procenjenim odzivom patern neurona. Kada su svi odzivi patern neurona procenjeni, primenjuje se segmentacija prednjeg plana tako što se određuje koji BNN-ovi imaju malu aktivnost i procenjuje se izlaz za aktivne BNN-ove. Nakon segmentacije težinski faktori probabilističkih neuralnih mreža (*Probabilistic Neural Network – PNN*) se koriguju i procenjuje se kriterijum zamene za svaki neuron sheme. Na samom kraju, neadekvatne neurone shema menja centralni procesor hosta. Najčešće se na kraju vrši dodatna obrada za uklanjanje povezanih komponenti, čija je veličina manja od očekivane veličine objekta, iz segmentacione maske. Ovaj korak, takođe obavlja centralni procesor.

Eksperimenti su vršeni na dva različita principa vezana za korišćenje memorije. U prvom setu eksperimenata, BNN-ovi su smešteni u memoriji hosta, a težine paterna su kopirane za svaki patern po potrebi i primljene nakon obrade. U drugom, svi podaci vezani za BNN smešteni su u memoriji GPU-a, smanjujući pri tom broj prenosa podataka između GPU-a i hosta. Na ovaj način, povećani su zahtevi za memorijom unutar samog GPU-a. Prenosi podataka koji su na ovaj način eliminisani, predstavljeni su u boji na Sl.1.

IV. REZULTATI

Pored CUDA SDK, za video ulaz i izlaz, kao i za završnu obradu rezultata korišćen je OpenCV.

Algoritam je testiran na velikom broju (30+) javno dostupnih video sekvenci snimljenih kamerama za video nadzor. Korišćene su sekvence sa složenom pozadinom, kao u radovima [1], [2], [16] i [17] i sekvence koje je snimila naša istraživačka grupa. S obzirom da implementacija na grafičkom procesoru ne utiče na tačnost segmentacije, fokusiraćemo se na računске performanse. Na Sl.2. prikazan je primeri segmentacije na frejm iz sekvence „Tunel“. Zeleni pikseli na slici segmentacije predstavljaju vrednosti koje su već u modelu, ali su klasifikovani kao prednji plan. Žuti pikseli odgovaraju neaktivnim BNN-ovima, npr. pikseli koji se trenutno ne nalaze u modelu pozadine.

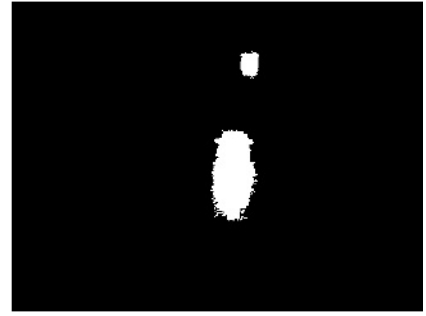
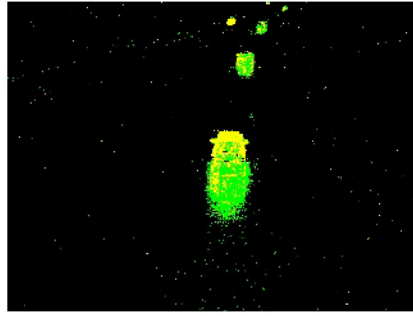


Sl. 1. Tok implementacije programa na grafičkom procesoru.

Testiranje je vršeno na dva različita NVIDIA grafička procesora: nešto stariji GeForce 8400M GS za portabilne računare, koji sadrži 16 CUDA jezgara, 256 MB memorije i radni takt od 400 MHz i GeForce GTS 250, koji sadrži 128 CUDA jezgara, 1 GB memorije i radni takt 738 MHz. U Tabeli 1 prikazana su vremena obrade dobijena za GeForce 8400M na sekvencama sa standardnim rezolucijama, za oba pristupa bazirana na BNN-ovima smeštenim na grafičkom procesoru (A vreme) i u memoriji hosta (B vreme). Vremena obrade dobijena su za BNN-ove koji sadrže 10 neurona za svako obeležje, koristeći 256 tredova za 8400M i 384 treda za 250 GTS. Brzina učenja (β) podešena je na 0.05, prag aktivacije (θ) podešen je na 0.5, a parametar glatкости (σ) na 7. U prikazanim rezultatima nisu uračunata vremena potrebna za završnu obradu, ali je uračunato vreme potrebno za prenos podataka između hosta i uređaja. Ukoliko je moguće obraditi 10-15 frejmova po sekundi (67-100 ms/frejmu) na rezoluciji 160x120 piksela, smatra se da je moguće vršiti procesiranje složenih pozadina u realnom vremenu. Na osnovu vremena datih u Tabeli 1 može se zaključiti da BNN pristup, kad je implementiran na grafičkom procesoru može vršiti segmentaciju u realnom vremenu na televizijskoj sekvenci standardne definicije (SD). Eliminacijom nepotrebnog prenosa podataka između hosta i uređaja dobijeno je prosečno ubrzanje od 1.38 puta za 8400M GS i 1.51 puta za 250 GTS.

V. ZAKLJUČAK

U radu je opisana implementacija najefikasnijeg probabilističkog algoritma za segmentaciju prednjeg plana u okviru grafičke jedinice korišćenjem CUDA-e i OpenCV-a. Paralelna obrada omogućila je segmentaciju videa, sa složenim sadržajem zadnjeg plana, rezolucije i brzine osvežavanja koja nikad nije postignuta za probabilističke algoritme. Zahvaljujući paralelnoj obradi omogućena je i segmentacija u realnom vremenu



Sl. 2. Segmentacija sekvence "Tunel": originalni frejm (levo), segmentacija pre (centar) i posle dodatne obrade (desno).

korišćenjem BNN pristupa. Korišćenjem komercijalnog programabilnog grafičkog hardvera i BNN pristupa za segmentaciju prednjeg plana moguće je obrađivati televizijski video signal standardne definicije (SD) u realnom vremenu.

Pored predstavljanja nekoliko aspekata praktične implementacije originalnog algoritma, predložena su dva pristupa za čuvanje podataka neuralne mreže i poređeni su rezultati dobijeni obradom najčešće korišćenih test sekvenci snimljenih kamerama za video nadzor. Dalji rad biće usmeren ka prebacivanju kompletne obrade na grafički procesor. Pored završne obrade, kao logično proširenje predložene metodologije, na grafičkom procesoru biće implementirano i praćenje pokretnih objekata bazirano na neuralnim mrežama.

ZAHVALNICA

Ovaj rad delom je sponzorisan E!UREKA REMSIS projektom.

LITERATURA

- [1] D. Culibrk, O. Marques, D. Socek, H. Kalva, B. Furht, "Neural network approach to background modeling for video object segmentation," *IEEE Trans. on Neural Networks*, vol. 18. Number 6, IEEE Press (Nov. 2007) 1614-1627.
- [2] L.Li, W.Huang, I.Y.-H.Gu, Q.Tian, "Statistical Modeling of Complex Backgrounds for Foreground Object Detection," *IEEE Trans. Image Processing*, vol. 13, pp. 1459-1472, November 2004.
- [3] C. Stauffer, W. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, pp. 747-757. (2000).
- [4] A. Elgammal, R. Duraiswami, D. Harwood, L. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proc. of the IEEE*, vol. 90, No. 7, pp. 1151-1163. (2002).
- [5] Y. Sheikh, M. Shah, "Bayesian modeling of dynamic scenes for object detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1778-1792. (2005).
- [6] Nvidia cuda programming guide version 2.0. http://www.nvidia.com/object/cuda_develop.html (Dec. 2008).
- [7] S. Lefebvre, S. Hornus, F. Neyret, "Computer Vision on the GPU," *GPU Gems 2 - Programming Techniques for High-Performance Graphics and General-Purpose Computation*. Addison Wesley (2005) 651-667.
- [8] G. Bradski, A. Kaehler, "Learning OpenCV: Computer Vision with the OpenCV Library," O'Reilly, Cambridge, MA (2008).
- [9] L. Rosin, "Thresholding for change detection," *Proc. of the Sixth International Conference on Computer Vision (ICCV'98)*, (1998).
- [10] S. Chien, Y. Huang, B. Hsieh, S. Ma, L. Chen, "Fast video segmentation algorithm with shadow cancellation, global motion compensation, and adaptive threshold techniques," *IEEE Trans. on Multimedia*, vol. 6, pp. 732-748. (2004).
- [11] T. Boulton, R. Micheals, X. Gao, P. Lewis, C. Power, W. Yin, A. Erkan, "Frame-rate omnidirectional surveillance and tracking of

- camouflaged and occluded targets," *Proc. of IEEE Workshop on Visual Surveillance*, pp. 48-55. (1999).
- [12] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," *International Conference on Pattern Recognition (ICPR)*. Volume 2. (Aug.2004) 28-31.
- [13] M. Gipp, G. Marcus, N. Harder, A. Suratanee, K. Rohr, R. Knig, R. Manner, "Haralicks texture features computed by gpus for biological applications," *IAENG International Journal of Computer Science* 36(1) (Feb. 2009).
- [14] Using an Extended Colinearity Criterion, G.B.F.B.S, "Reduction of coding artifacts using chrominance and luminance spatial analysis," *VMV 2005*. (Nov. 2005).
- [15] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Stat.*, Vol. 33, pp. 1065-1076. (1962).
- [16] A. Monnet, A. Mittal, N. Paragios, V. Ramesh, "Background modeling and subtraction of dynamic scenes," *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, Washington, DC, USA, IEEE Computer Society (2003) 1305-1312.
- [17] D. Culibrk, B. Antic, V. Crnojevic, "Real-time stable texture regions extraction for motion-based object segmentation," *Proceedings of the 20th British Machine Vision Conference (BMVC 2009)*. (Sept. 2009).

ABSTRACT

Moving object detection in visual surveillance videos is usually accomplished through background subtraction. When segmentation of interesting moving objects from the background is to be done in the presence of moving objects in the background itself, the process calls for the use of complex probabilistic background models. Maintaining such models is computationally intensive and limits the real-time applications of such methodologies to low resolution sequences, far below the acquisition ability of state-of-the-art cameras. While most probabilistic foreground segmentation approaches can benefit from parallel processing, since they allow for parallelization of operations done for each pixel, the Background Modelling Neural Networks (BNNs) allow the process to be parallelized even further, since the computation can be parallelized at the level of feature patterns stored in the model for each pixel. The paper presents a parallel implementation of BNN-based moving object detection, running on NVIDIA™ Graphics Processing Unit (GPU). Experiments show an order of magnitude speed-up can be achieved in this manner, allowing for significant increase in the resolution of sequences processed in real time.

GPU-BASED PARALLEL PROCESSING APPROACH TO MOVING OBJECT SEGMENTATION IN VIDEO

Dubravko Culibrk, Dragan Letić, Vladimir Crnojević