

GRAFIK PROTSessorLARDA TASVIRLARGA PARALLEL ISHLOV BERISH VA SHOVCINLARDAN TOZALASH

Abjalova Maloxat Ismoilovna
UMFT universiteti magistranti
E-mail: abjalovamaloxat1@gmail.com

ANNOTATSIYA

Hozirgi kunda jahonda tasvirlardan avtomatlashtirilgan holda ob'yektlarni aniqlash, tanib olish va tahlil qilish algoritmlarini takomillashtirish hamda tasvirlarini qayta ishlashda qo'llaniladigan segmentlash usullaridan foydalanish dolzarb hisoblanadi. Global raqamli transformatsiya jarayonlarida tasvirlarga raqamli ishlov berish masalalariga ehtiyoj juda ortgan, bunda kompyuter nigohi texnologiyalari yetakchilik qilmoqda. Olimlar inson miyasini funksiyalarini yaratishga qaratilgan intilishlari miyaning biologik neyronlar va ularning bog'lanishlari modelini (keyinchalik esa dasturiy) yaratish natijalarini berdi.

Kalit so'zlar: Grafik protsessor, tasvirlaeni qayta ishlash, tasvir, obyekt, ilova, interfeys, kompyuter, algoritm, signal, parallel, kodlash.

I. KIRISH

Grafik protsessorlarda tasvirlarga parallel ishlov berish va shovqinlardan tozalashdagi ob'ektlarni tanib olish algoritmi bo'lib, u bilan ishlash uchun interfeysni ishlab chiqishni o'z ichiga oladi. Ish mavzusi tasvirlardagi ob'ektlarni tanib olish uchun mashinani o'rganish usullari va algoritmlari, shuningdek, grafik protsessorlarda tasvirlarga parallel ishlov berish va shovqinlardan tozalash asosida foydalanuvchi interfeysini amalga oshirish. Tasvirlarni tasniflash algoritmlarini ishlab chiqish, neyron tarmoq modellarini o'rgatish va bu komponentlarni yagona dastur interfeysiga integratsiya qilishdir. Tasvirlarga ishlov berish va tahlil etish inson faoliyatining tasvirlarga aloqador bo'lgan bir muncha sohalarda qo'llaniladi. Grafik ishlov berish birligi yoki GPU shaxsiy va biznes hisoblash uchun hisoblash texnologiyasining eng muhim turlaridan biriga aylandi. Parallel ishlov berish uchun mo'ljallangan GPU keng ko'lamdagi dasturlarda, shu jumladan grafik va video renderlashda qo'llaniladi.[1,2]

Grafik ishlov berish komponentlari zamonaviy grafik ishlov berishlar kompyuter grafikasi va tasvirni qayta ishlashda samarali ishlaydi. Ularning parallel tuzilishi ularni katta ma'lumotlar bloklarini parallel ravishda qayta ishlaydigan algoritmlar uchun umumiy maqsadli markaziy protsessorlarga (CPU) qaraganda samaraliroq qiladi. Shaxsiy kompyuterda grafik ishlov berish video kartada yoki anakartga o'rnatilgan

bo'lishi mumkin. Ba'zi protsessorlarda ular markaziy protsessorga o'rnatilgan. 1970-yillarda "grafik ishlov berish" atamasi dastlab grafik protsessor bloki degan ma'noni anglatadi va protsessoridan mustaqil ravishda ishlaydigan va grafik manipulyatsiyasi va chiqishi uchun mas'ul bo'lgan dasturlashtiriladigan ishlov berish blokini tavsiflaydi. Keyinchalik, 1994 yilda Sony ushbu atamani (hozirda grafik ishlov berish birligi uchun) PlayStation konsolining 1994 yilda Toshiba tomonidan ishlab chiqilgan Sony grafik ishlov berishga nisbatan ishlatgan. Bu atama 1999 yilda Nvidia tomonidan ommalashgan. GeForce 256 ni "dunyodagi birinchi grafik ishlov berish" sifatida sotdi. U "bir chipli protsessor" sifatida taqdim etildi.[1,2,3].

II. ASOSIY QISIM.

2.1 Grafik prosessorlar va GPU da dasturlash.

Kompyuter fanining boshqa amaliy sohalari singari, parallel dasturlash ham bir necha bosqichlardan o'tdi. Parallel dasturlash 1960-yillarda operatsion tizimlar sohasida paydo bo'lgan. Parallel dastur vazifani bajarish uchun birgalikda ishlaydigan bir nechta jarayonlarni o'z ichiga oladi. Ketma-ket dasturda bitta nazorat tarmog'i mavjud, parallel dasturda esa bir nechta. Parallel dastur jarayonlarining birgalikdagi ishi ularning o'zaro aloqasi orqali amalga oshiriladi.

Parallel algoritm-bu turli xil hisoblash qurilmalarida bo'laklab bajarilishi va keyinchalik olingan natijalarni birlashtirib to'g'ri natijaga olish mumkin bo'lgan algoritm. Parallel dasturlash bir vaqtning o'zida bir nechta hisoblash tugunlarida kodni bajarilishi hisobiga resurslardan samarali foydalanadigan dasturlarni yaratish uchun ishlatiladi.[2]

Hozirgi parallel hisoblashlarning aksariyati parallel ketma-ket dasturlar sifatida amalga oshiriladi. Bunday holda, mavjud ketma-ket algoritmda parallel ravishda bajarilishi mumkin bo'lgan mustaqil ketma-ket qismlar ajratiladi va buni inobatga olgan holda parallel ketma-ket dastur yoziladi. Yorug'lik yig'adigan minglab mayda hujayralardan tashkil topgan CCDlar tasvirni minglab mayda bo'laklarga bo'linadi. Ular kichik katakchalardan iborat bo'lib, ularning har biri tasvirning bitta bo'lagiga mos keladigan ma'lumotlarni yozib oladi va uzatadi.

Tasvirlar ikki yoki uch o'lchovli bo'lishi mumkin, masalan, fotosurat yoki ekran displeyi yoki haykal yoki gologramma kabi uch o'lchovli. Ular kameralar, nometalllar, linzalar, teleskoplar, mikroskoplar va boshqalar kabi optik qurilmalar va inson ko'zлари yoki suv kabi tabiiy ob'ektlar va hodisalar tomonidan olinishi mumkin. [3,4]

Piksellarining ekranda joylashishini shaxmat doskasidagi kataklarning joylashishiga yoki rangli jadvalga o'xshatish mumkin. Monitor turiga qarab matritsa o'lchamlari piksellarda 640x480, 800x600, 1024x768, 1600x1200... kabi bo'lishi mumkin. Hozirgi kunda tasvirni 16 bitli kodlash imkoniga ega dasturlar mavjud bo'lib,

ular yordamida kulrangning 256 xil emas, balki 65536 xil tusi ishtirok etgan tasvirlar hosil qilish mumkin. Ammo bunday tasvirlarni saqlashda fayl hajmi 8 bitliga nisbatan 2 marta katta bo'ladi. Ranglarning har birining yorqinligi 0 dan 255 gacha bo'lishi mumkin. Natijada $256 \times 256 \times 256$ ta (o'n olti milliondan ko'proq) ranglar hosil qilish mumkin. [3,5]

2.2. Eksperimental natijalar. Usulimizning samaradorligini ko'rsatish uchun biz tajribalarimizda tashqi ko'rinishlarning haqiqiy tasvirlaridan foydalandik. Biz tajribalarimizda radiometrik kalibrlashni qo'llamagan edik, ammo biz gamma-tuzatuvini o'chirilgan kamera natijalarni yaxshilaydi, deb hisoblaymiz, ayniqsa, haqiqiy ranglarga nisbatan mos ranglarni olish uchun. Pentium 4 va 1 GB xotiraning ikki protsessoridan foydalangan holda 600×400 o'lchamli tasvirlar uchun hisoblash vaqti, taxminan besh-etti daqiqaga (bir nechta teglar bilan grafik kesishlarni qo'llash). Rasmning o'rta va pastki qismida to'g'ridan-to'g'ri zaiflash tasviri va havo nuri ko'rsatilgan. To'g'ridan-to'g'ri o'chirilgan tasvirda kameradan uzoqda joylashgan sariq mashina ko'rinayotganini kuzatish mumkin. Internetdan olingan 1-2-rasmda tuman bosgan tashqi manzara ko'rsatilgan. Rasmning o'rtasi va pastki qismi. To'g'ridan-to'g'ri zaiflashuvni va havo nurini ko'rsatadi. 2-rasmda yomg'irli ob-havoning kirish tasviri, taxminiy yo'nalish ko'rsatilgan. [4]



Rasm 1. Yuqori: kirish tasviri.
O'rta: to'g'ridan-to'g'ri susaytirish.
Pastki qism: havo nuri.



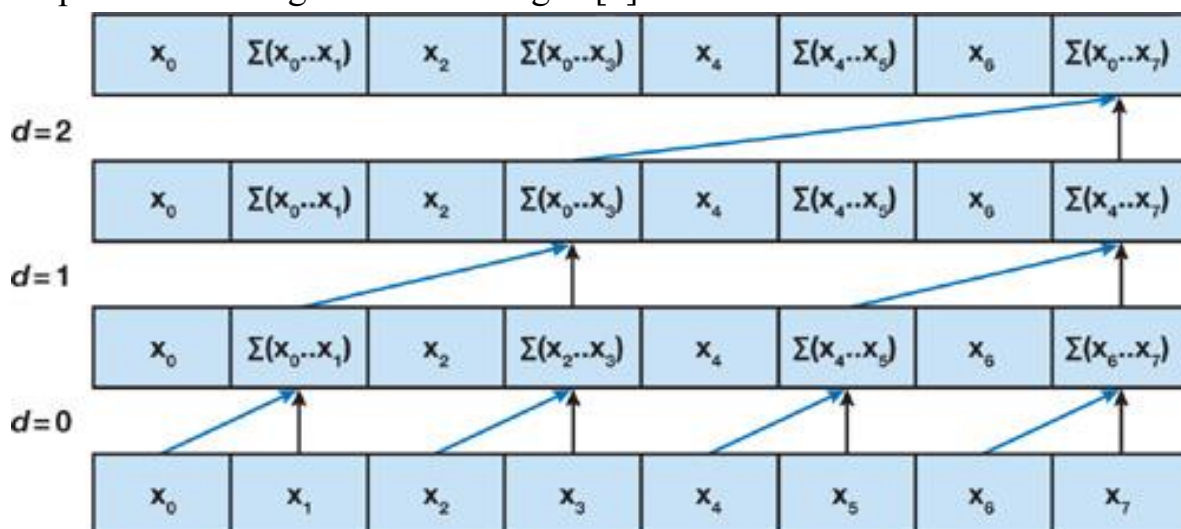
2-rasm. Yuqori: kirish tasviri.
O'rta: to'g'ridan-to'g'ri susaytirish.
Pastki qism: havo nuri.

zaiflashuv, yer haqiqati va taxminiy havo nuri. Biz aniq ob-havoda olingan tasvirni asosli haqiqat deb hisoblaymiz. Oxirgi natija uchun xulosa chiqarish shartli rejimlar (ICM) yordamida, dastlabki qiymatlar sifatida loyqa Y-tasvirdan foydalangan holda

amalgam oshirildi. Garchi ICM global optimallikni kafolatlay olmasa ham, natija ko‘rish sezilarli darajada yaxshilanganligini ko‘rsatadi. Bu bizning xarajat funksiyamizning samaradorligi va boshlang‘ich qiymatlari bilan bog‘liq.[4]

2.3 Ish uchun samarali parallel skanerlash.

Skanerlashni amalga oshirishimiz ish samarasizligi tufayli katta massivlarda juda yomon ishlashi mumkin. GPU dagi parallellikdan foydalanib, ketma-ket algoritm samaradorligiga yaqinlashadigan algoritmni topmoqchimiz. Sodda algoritm tomonidan bajariladigan $\log_2 n$ ishining qo‘shimcha omilidan qochadigan CUDA uchun samarali skanerlash algoritmini ishlab chiqishdir. Ushbu algoritm Blelloch (1990) tomonidan taqdim etilgan algoritmga asoslangan. Parallel hisoblashda paydo bo‘ladigan algoritmik naqshdan foydalanamiz: muvozanatli daraxtlar. G‘oya kiritilgan ma‘lumotlarda muvozanatli ikkilik daraxtni qurish va prefiks yig‘indisini hisoblash uchun uni ildizga va ildizdan tortib olishdan iborat. n ta bargli binar daraxt $d = \log_2 n$ darajaga ega va har bir d darajasida 2 ta tugun mavjud. Agar biz har bir tugunga bitta qo‘shimchani bajarsak, u holda daraxtning bir marta o‘tishida $O(n)$ qo‘shimchalarini bajaramiz. Biz yaratgan daraxt haqiqiy ma‘lumotlar strukturasi emas, balki biz har bir ip o‘o‘tishning har bir bosqichida nima qilishini aniqlash uchun foydalanadigan tushunchadir. Ushbu samarali skanerlash algoritmidan biz umumiy xotiradagi massivda amallarni bajaramiz. Algoritm ikki bosqichdan iborat: pasaytirish va yuqoriga surish bosqichi. Qisqartirish bosqichida biz 3-rasmda ko‘rsatilganidek, daraxtning ichki tugunlarida qisman summalarni hisoblab, barglardan ildizga o‘tamiz. Bu parallel qisqartirish deb ham ataladi, chunki bu bosqichdan so‘ng ildiz tugun (massivdagi oxirgi tugun) massivdagi barcha tugunlarning yig‘indisini ushlab turadi. Qisqartirish bosqichi uchun psevdokod 3-algoritmda keltirilgan.[2]



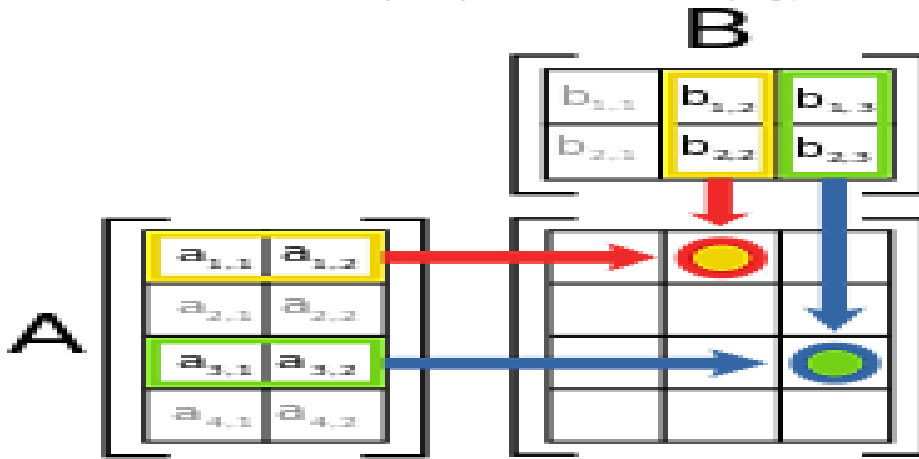
3-rasm. Ish uchun samarali yig‘indini skanerlash algoritmining yuqoriga ko‘tarish yoki kamaytirish bosqichining tasviri.

2.3 Tasvir va subyektiv miqdoriy tajribalar

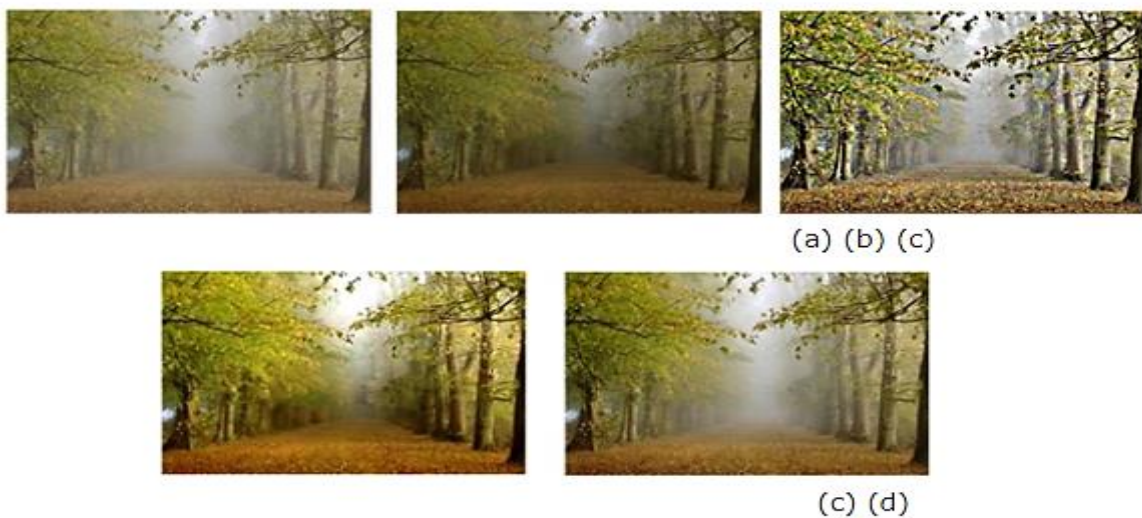
4-rasmda ikkita A va B matritsalarining ko‘paytmasi diagrammatik tarzda tasvirlangan bo‘lib, mahsulot matritsasiidagi har bir kesishma A qatoriga va B ustuniga qanday mos kelishini ko‘rsatadi.

$$\begin{matrix} 4 \times 2 \text{ matrix} \\ \begin{bmatrix} a_{11} & a_{12} \\ \cdot & \cdot \\ a_{31} & a_{32} \\ \cdot & \cdot \end{bmatrix} \end{matrix} \cdot \begin{matrix} 2 \times 3 \text{ matrix} \\ \begin{bmatrix} \cdot & b_{12} & b_{13} \\ \cdot & b_{22} & b_{23} \end{bmatrix} \end{matrix} = \begin{matrix} 4 \times 3 \text{ matrix} \\ \begin{bmatrix} \cdot & c_{12} & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & c_{33} \\ \cdot & \cdot & \cdot \end{bmatrix} \end{matrix} \quad \begin{matrix} c_{12} = a_{11}b_{12} + a_{12}b_{22} \\ c_{33} = a_{31}b_{13} + a_{32}b_{23} \end{matrix}$$

4-rasmda doiralar bilan belgilangan chorrahalaridagi qiymatlar:[4]



Tasvirlarni paralelligi. Shakl 4. “toys.jpg” da turli dehazing algoritmlarini qo‘llash orqali erishilgan natijalar, bu erda (a) maqsadli tuman tasviri; (b) U va boshqalar; (c) He va Sun; (d) Ketan va boshqalar. va (e) Yu va boshqalar.



Bunda “trees.jpg” da turli xil dehazing algoritmlarini qo‘llash orqali erishilgan natijalar, bu erda (a) Maqsadli tuman tasviri; (b) U va boshqalar; (c) U va boshqalar; (d) Ketan va boshqalar. va (e) Yu va boshqalar.[5]

Dehazing qaysi algoritm qo'llanilishidan qat'iy nazar erishiladi. Biroq, yaqindan kuzatish shuni ko'rsatadiki, Ref. U va boshqalar, He et Sun. Yu va boshqalarga qaraganda yuqoriroq, bu algoritmlar ko'k rangli porlash bilan qoplangan natijalarni taqdim etadi, bu kamchilik ularning atmosfera yorug'ligi parametrini etarli darajada baholamaganligi bilan bog'liq. "trees.jpg" ma'lumotlar to'plami bilan erishilgan natijalar ham Ketan va boshqalarning ushbu tendentsiyasiga mos keladi. dehazing effekti va kontrast balansi o'rtasidagi mukammal muvozanatga erishish. Natijalar Refs. U va boshqalar. va He va Sun ishlash jihatidan yaqindan kuzatib boradi, eng kam yaxshilangan natijalar Yu va boshqalarda erishiladi.[3,4]

Yuqoridagi kichik bo'limda biz algoritmni metrik sifatida vizual effektdan foydalangan holda taqqoslagan bo'lsak, ushbu kichik bo'limda ushbu turli xil algoritmlarning haqiqatda qanday ishlashini va ularning bo'lmaganlarga qanday ta'sir qilishini yoritishga harakat qilish uchun ko'proq miqdoriy taqqoslash amalga oshiriladi. tasvirning vizual komponenti. Bu erda o'tkazilgan tajribalarda biz quyidagi tasvir ko'rsatkichlarini baholash orqali turli xil dehazing algoritmlari tasvir sifatiga qanday ta'sir qilishini tekshiramiz:

1. O'rtacha kvadrat xato (MSE).
2. Eng yuqori signaldan shovqin nisbati (PSNR)
3. Signaldan shovqin nisbati (SNR).
4. Strukturaviy o'xshashlik indeksi (SSIM)

Miqdoriy taqqoslash uchun bu erda qo'llaniladigan deyarli barcha ko'rsatkichlar tabiatan to'liq mos yozuvlar bo'lganligi sababli, algoritmlarning dehazing samaradorligini haqiqatda aks ettiruvchi afzalliklarni samarali olish uchun turli xil algoritmlardan ham tumansiz, ham dehazed chiqish talabi bilan amalga oshiriladi. . Shu sababli, tumansiz va ular bilan birga kelgan tumanli tasvirlar ma'lumotlar to'plami bizning ma'lumotlarimizga ko'ra deyarli mavjud emasligi sababli, biz tumansiz tashqi tasvirlarni tanlaymiz. Ketan va boshqalarda qo'llanganiga o'xshash hazing funksiyasidan foydalanib tumanni sintez qilamiz. Ushbu ma'lumotlar to'plami bilan biz turli xil algoritmlar yordamida dehazingni amalga oshirishni davom ettiramiz va keyin asl tasvirlarni mos yozuvlar sifatida va dehazatsiya qilingan chiqish tasvirlarini maqsad sifatida ishlatib, o'lchovlarni chiqaramiz. Quyidagi 1-jadvalda ushbu tekshirish sxemasi orqali olingan natijalar keltirilgan.

1-jadvalda keltirilgan natijalardan ko'rinib turibdiki, qabul qilingan ko'rsatkichlar haqiqatan ham turli xil algoritmlarning dehazatsiya samaradorligi bilan tegishli korrelyatsiyani saqlab turadi va turli xil dehazatsiya algoritmlarining maqsadli tasvirlarning ichki sifatlariga ta'siri haqida ma'lumot beradi. inson ko'ziga sezilmaydigan yo'l. Umuman olganda, Ketan va boshqalar. eng yuqori SSIM qiymatiga erishadi va shuning uchun chiqish tasvirini tumansiz tasvirga yaqinlashtiradigan dehazing aniqligiga erishadi. He va Sun va He va boshqalar. ham maqbul natijalarga erishadi.[5]

Algorithms	Image Properties			
	MSE	PSNR	SNR	SSIM
He et Sun	7205.8	9.5	5.9	0.6
Ketan et al.	6120.5	9.7	6.2	0.7
Yu et al.	9607.5	6.4	5.2	0.3
He et al.	8403.7	8.2	5.7	0.5

Jadval 1. Ko‘rsatkichlar sifatida ichki tasvir xususiyatlaridan foydalangan holda eng so‘nggi texnologiyalarni taqqoslash. (“toys.jpg” va “trees.jpg” orasidagi o‘rtacha) bu sohada, keyin Yu va boshqalarda taqdim etilgan algoritm. [4,5]

FOYDALANILGAN ADABIYOTLAR RO‘YXATI: (REFERENCES)

1. Denni Atkin. “Kompyuter xaridori: Siz uchun to‘g‘ri GPU”. Asl nusxadan arxivlangan, 2007-05-06 . Olingan 2007-05-15.
2. Barron, ET; Glorioso. RM (1973 yil sentyabr). “Mikro boshqariladigan periferik protsessor”. MICRO 6: Mikrodasturlash bo‘yicha 6-yillik seminarining konferentsiya yozuvi. Mikro 6: 122–128.
3. Xalilov M. A. Tasvirlama ma‘lumotni qayta ishlashda rangli tasvirni kulrang tasvirga o‘tkzish algoritmini tadqiq etish. Bituruv malakaviy ish. – Namangan: Namangan davlat universiteti, 2019. 109 b.
4. Tan, R. Visibility in bad weather from a single image C. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008 pp. 1–8.
5. Harris, M. Sengupta, S. Owens, J.D. Parallel prefix sum (scan) with CUDA. GPU Gems **2007**, 3, 851–876.