

ГРАФИК ПРОТРЕССОРЛАРДА ТАСВИРЛАРГА ПАРАЛЛЕЛ ИШЛОВ БЕРИШ МОДЕЛИ ВА АЛГОРИТМЛАРИНИ ИШЛАБ ЧИҚИШ

Baxtiyor Uralovich AKMURADOV

dotsent

Texnika fanlari bo'yicha (PhD) falsafa doktori
Menejment va zamonaviy texnologiyalar universiteti
Toshkent, O'zbekiston

Muhriddin Abduxalil o'g'li UMAROV

dotsent

Texnika fanlari bo'yicha (PhD) falsafa doktori
Menejment va zamonaviy texnologiyalar universiteti
Toshkent, O'zbekiston

muhriddin.umarov.1992@gmail.com

Farrux Anvarovich AXMEDOV

katta o'qituvchi

Menejment va zamonaviy texnologiyalar universiteti
Toshkent, O'zbekiston

Annotatsiya

Maqolada raqamli tasvirlarga dastlabki ishlov berish va tasvirlarni shovqinlardan tozalash masalalari tahlil etilgan.

Tayanch so'zlar. grafik protsessor, tasvir intensivligi, shovqinlardan tozalash, parallel ishlov berish, tasvirlarga ishlov berish, dinamik tasvir.

РАЗРАБОТКА МОДЕЛЕЙ И АЛГОРИТМОВ ПАРАЛЛЕЛЬНОЙ ОБРАБОТКИ ИЗОБРАЖЕНИЙ В ГРАФИЧЕСКИХ ПРОЦЕССОРАХ

Бахтиёр Уралович АКМУРАДОВ

доцент

Доктор философии (PhD) по техническим наукам
Университет управления и технологий будущего
Ташкент, Узбекистан

Мухриддин Абдухалил угли УМАРОВ

доцент

Доктор философии (PhD) по техническим наукам
Университет управления и технологий будущего
Ташкент, Узбекистан

muhriddin.umarov.1992@gmail.com

Фаррух Анварович АХМЕДОВ

старший преподаватель

Университет управления и технологий будущего
Ташкент, Узбекистан

Аннотация

В данной статье анализируются вопросы первичной обработки цифровых изображений и удаление построенного шума из изображений.

Ключевые слова: графический процессор, интенсивность изображения, удаление шума, параллельная обработка, обработка изображений, динамическое изображение.

Sun'iy intellekt texnologiyalari va chuqur o'qitishga asoslangan algoritmlarning rivojlanishi bilan yo'l belgilarini tanib olish tizimlari ham jadal rivojlanib bormoqda. Yo'l sahnasi tasvirlari odatda, tashqi muhit ta'sirida turli shovqinlarga ega bo'ladi, bunda tasvirlardan obyektlarni tanib olish murakkablashadi hamda tanib olish aniqligi sezilarni darajada kamayadi. Shundan kelib chiqib, hozirgi kunda kompyuterli ko'rish sohasida tasvirlardan shovqinlarni tozalash va yorqinligini oshirish algoritmini ishlab chiqish muhim masala hisoblanadi [2].

Tadqiqotchilar tumanli tasvirlardan shovqinlarni olib tashlash bo'yicha bir qancha algoritmlarni taklif qilishgan. Xitoylik olim Z.Tan tomonidan tumanli tasvirlar tumansiz tasvirlarga nisbatan kontrasti past ekanligi aniqlandi [3], bunda tasvir kontrastini oshirish orqali tasvirlardan shovqinlarni tozalash g'oyasini taklif qildi. Biroq, bunday usullar tasvirdagi tuman konsentratsiyasi yuqori bo'lgan hollar uchun yaxshi samara bermaydi. Xitoylik olim W.He tomonidan o'z tadqiqotlaridan kelib chiqib, tasvirning intensivligi past bo'lgan kanaliga ishlov berish (DCP - Dark Channel Prior) algoritmini taklif qildi [4]. W.He tomonidan 2013-yilda boshqariladigan filtr (guided filter) ga asoslangan yangi yondashuv taklif qilindi [5]. Bu usul tasvirdagi obyektlarning chegaralarini yaxshi saqlab qolishi bilan boshqa usullardan ajralib turadi. Bundan tashqari ushbu usulni ishlash vaqtini tezlashtirish orqali real vaqt rejimida amalga oshirish mumkin.

Dinamik tasvirlarni qayta ishlash uchun mo'ljallangan grafik protsessorlar (GPU - Graphics Processing Unit) yordamida algoritmlarni real vaqt rejimida parallel amalga oshirish mumkin. Yuqoridagi algoritmlarni GPU da amalga oshirish orqali soniyada 27 ta kadrda ishlov berish mumkin, biroq yuqori formatdagi sifatli dinamik tasvirlarda yaxshi samara bermaydi [6].

Parallel minimum filtrlash algoritmi. An'anaviy minimum filtrlash algoritmidan qo'shni piksellarda ma'lumotlarni hisoblashning katta miqdori mavjud.

Parallel minimum algoritmini amalga oshirish (1), (2) va (3) formulalar yordamida quyidagicha bajariladi:

$$g_x = \begin{cases} f_x & \text{for } x = 1, r + 1, 2r + 1, \dots \\ \min[g_{x-1}, f_x] & \text{for } x = 2, \dots, r; r + 2, \dots, 2r; \\ & 2r + 2, \dots, 3r; \dots \end{cases} \quad (1)$$

$$h_x = \begin{cases} f_x & \text{for } x = n, n - r - 1, n - 2r - 1, \dots \\ \min[h_{x+1}, f_x] & \text{for } x = n - 1, \dots, n - r; \\ & n - r - 2, \dots, n - 2r - 1; \dots \end{cases} \quad (2)$$

$$n_x = \min[g_{x+(r-1)/2}, h_{x-(r-1)/2}] \text{ for } x = 1, \dots, n \quad (3)$$

f:	11	2	6	5	13	7	15	8	4	10	14	9	12	3	1
g:	11					7					14				
	← 2					← 7					← 9				
	← 2					← 7					← 9				
	← 2					← 4					← 3				
	← 2					← 4					← 1				
h:					13					10					1
				5 →				4 →					1 →		
			5 →					4 →				1 →			
		2 →						4 →				1 →			
		2 →						4 →				1 →			
n:	2	2	2	2	5	5	4	4	4	4	4	3	3	1	1

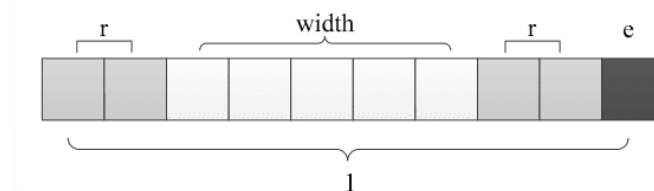
1-rasm. Parallel minimum filtrlash algoritmi ($r = 5$).

Minimum filtrda to'ldirish qiymati 255 ga teng va to'ldirish tartibi 2-rasmda ko'rsatilgan. Birinchi, har bir satr elementlarning ikkala tomoniga r (filtr yadrosining radiusi) elementlari qo'shiladi. Keyin, massivning oxiriga e to'ldirish elementlari qo'shiladi. Yangi vektor o'lchami l ga teng.

$$niqob = 2 * r + 1 \quad (4)$$

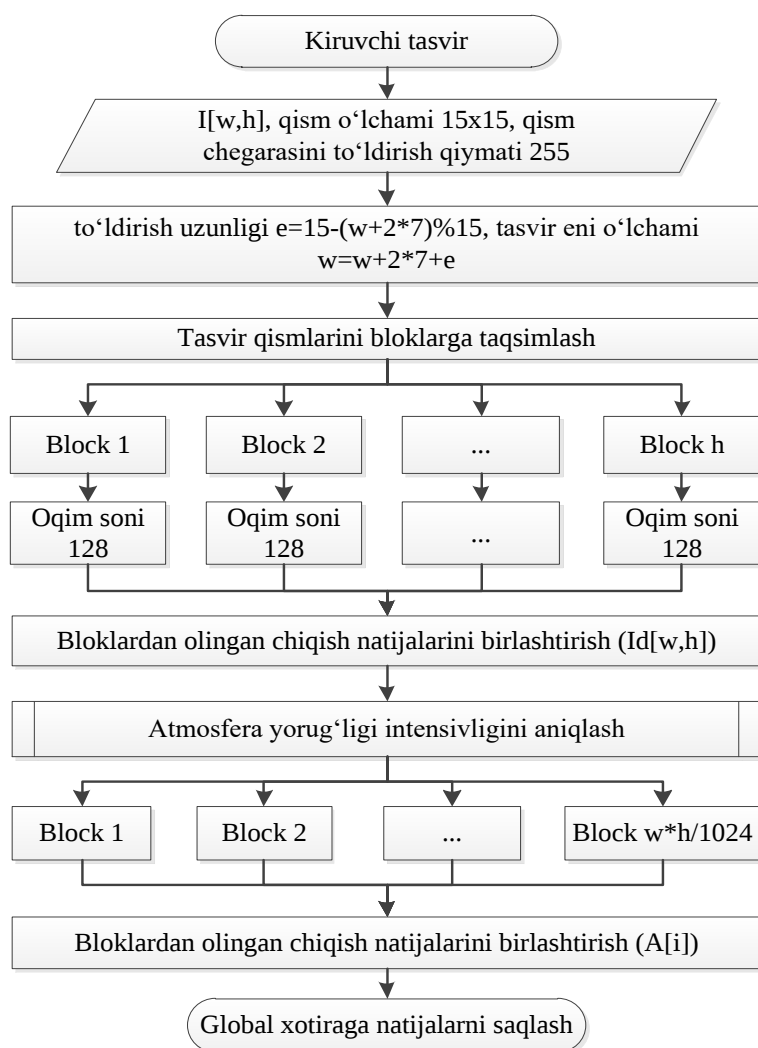
$$e = niqob - (width + 2 * r) \% niqob \quad (5)$$

$$l = width + 2 * r + e \quad (6)$$



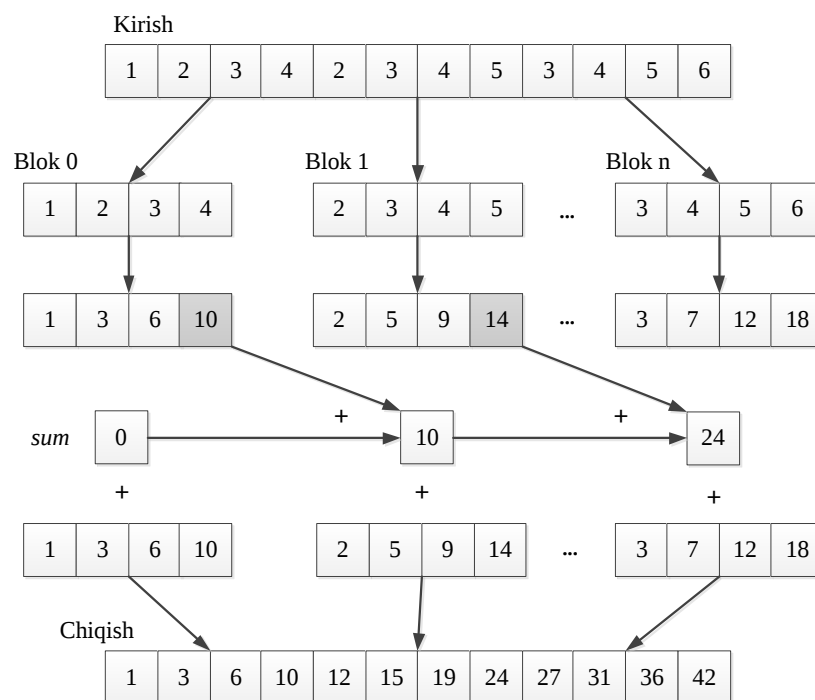
2-rasm. Minimum filtr chegarasini to'ldirish.

CUDA texnologiyasi bilan dasturlashda tasvirning har bir satri uchun oqim bloki va har bir oqim bloki uchun 128 ta oqim o'rnatildi. g_x va h_x massivlarni hisoblaganda kichik massiv diapazonidagi qo'shni nuqtalarga bog'liqlik mavjud, shuning uchun bitta kichik massivni qayta ishlashda bitta oqim ishlatiladi va bir nechta oqimlar parallel ravishda ishlaydi. n_x massivini hisoblashda, barcha piksel natijalari bir-biridan mustaqil bo'lganligi sababli, bitta pikselni qayta ishlash uchun bitta oqim sozlanadi, parallel ravishda qayta ishlash uchun bir nechta oqimlardan foydalanildi. Yuqoridagi parallel minimum filtrlash algoritmining blok sxemasi quyidagi 3-rasmda ko'rsatilgan.



3-rasm. Parallel minimum filtrlash algoritmining blok sxemasi.

Parallel o'rtacha filtrlash algoritmi. O'rtacha filtrni amalga oshirishda berilgan o'lchamdagi pikselning o'rtacha qiymati - bu berilgan sohadagi barcha elementlar yig'indisining elementlar soniga bo'linmasiga teng. Shuning uchun filtr yadrosidagi piksellarni yig'ish samaradorligini oshirishga e'tibor qaratiladi [1]. Agar kiruvchi bir o'lchovli massiv hajmi n bo'lsa, algoritmning vaqt murakkabligi $O(n)$ ga teng. Bu algoritm ikki qismdan iborat, yuqoriga (up-sweep) va pastga (down-sweep) tushirish darajadagi hisoblashlarga bo'linadi. Yuqoriga tushirish bosqichida har bir darajadagi tugunlar sonini yarmiga kamaytiriladi. Pastga tushirish bosqichida birinchi navbatda oxirgi elementga nol o'rnatiladi, so'ngra elementlar qo'shiladi. Algoritmning hisoblash murakkabligi $2 * (n - 1)$ qo'shish va $(n - 1)$ almashtirishdan iborat bo'lib, ularning vaqt murakkabligi $O(n)$, ya'ni ketma-ket algoritm bilan bir xil. Algoritmning samaradorligini oshirish uchun barcha hisoblash amallari umumiy xotirada amalga oshiriladi. Parallel o'rtacha filtrlash algoritmning hisoblash murakkabligi $O(n)$ ga teng va vaqt sarfini yanada kamaytirish uchun ma'lumotlarni bo'laklarga bo'lishdan foydalaniladi. 2.10-rasmdan ko'rsatilgandek, parallel o'rtacha filtrlash algoritmida kirish massivining uzunligi 2^n bo'ladi. Kirish ma'lumotlarining uzunligi n bo'lsa va har bir blok m ma'lumotni hisoblash uchun mas'ul bo'lsa, n/m oqim bloklarini yaratish kerak, har bir blok uchun $m/2$ oqim ajratiladi (4-rasm). Har bir kichik matritsaning parallel o'rtacha filtrlagandan keyin, har bir blokda oxirgi element qo'shimcha massiv sum_i da qayd etiladi, bu i -blokdan oldingi barcha elementlarning yig'indisi hisoblanadi.



4-rasm. Parallel oʻrtacha filtrlash algoritmi ($n = 12, m = 4$).

Natijalarni olish uchun sum_i massivni saqlashga global xotirani ajratish kerak. Barcha blok ichidagi hisob-kitoblar tugashini kutish va har bir filtr yadrosining oʻrtacha qiymatini hisoblash kerak. Bu bir oʻlchovli oʻrtacha filtrni amalga oshirish uchun ikkita yadro funksiyasini chaqirish kerakligini anglatadi; birinchi yadro yigʻindini hisoblash uchun, ikkinchi yadro oʻrtacha qiymatni hisoblash uchun ishlatiladi. Satrlar va ustunlarni ajratish uchun koʻchirilgan filtrlash algoritmidan foydalanildi hamda tasvirning bir satrini qayta ishlash uchun bitta oqimli blok foydalanildi [7]. Bir oʻlchovli filtr yadrosining barcha hisob-kitoblari uchun maʼlumotlarga kirishning kechikishini kamaytirish uchun umumiy xotiradan foydalanildi. Tasvir oʻlchami $w * h$ va filtr yadrosi radiusi r boʻlsa, bir oʻlchovli filtr yadrosini amalga oshirishning oʻziga xos sxemasi quyidagicha:

1-bosqich. Har bir oqim blokida h oqimlar ajratiladi va oqim bir vaqtning oʻzida $2 * k$ maʼlumotni qayta ishlay oladi.

2-bosqich. Har bir blok uchun ajratilgan umumiy xotira hajmi l ga teng.

$$l = [(w + 2 * r + 2 * k) / (2 * k)] * 2 * k \quad (7)$$

$$e = l - (w + 2 * r) \quad (8)$$

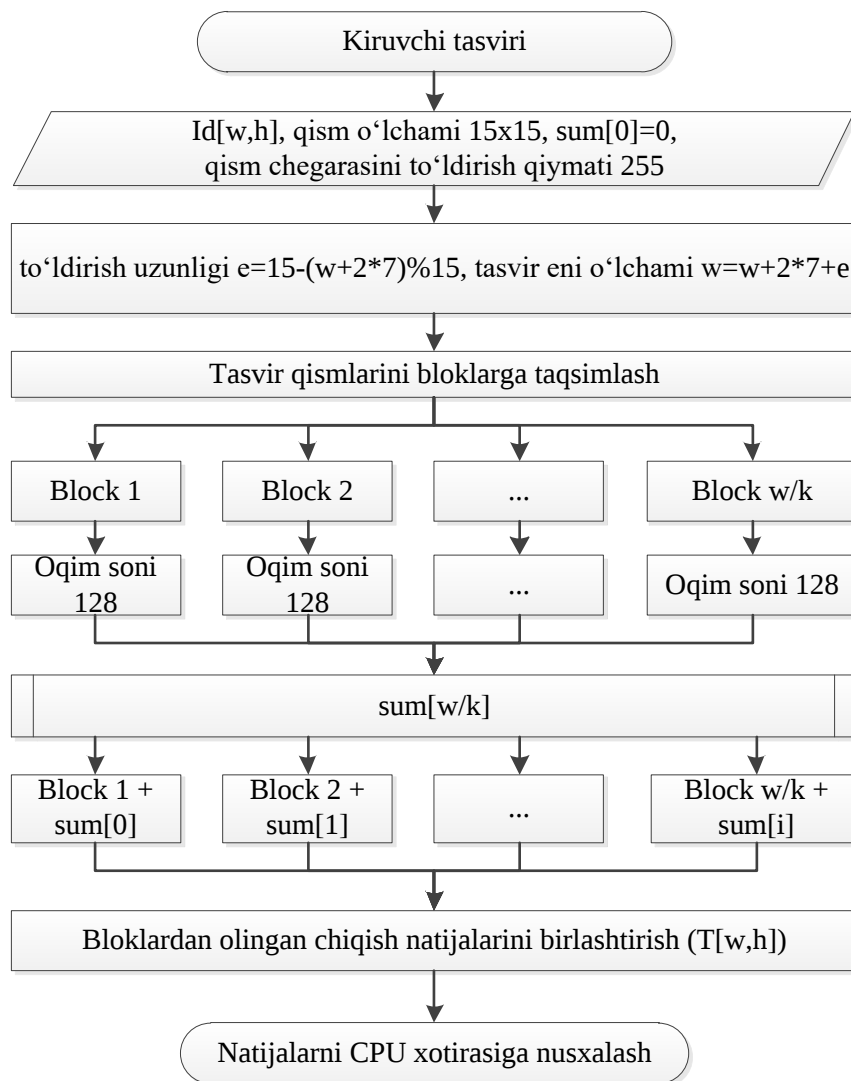
3-bosqich. Har bir blokni hisoblash uchun umumiy xotirada T ajratiladi, har bir oqim blokidagi barcha ma'lumotlar $l/(2 * k)$ kichik matritsaga bo'linadi. 5-rasmda ko'rsatilgandek, kichik massiv i oldidagi barcha piksellar yig'indisini saqlash uchun har bir oqim blokida $l/(2 * k)$ o'lchamdagi umumiy xotira sum_i ajratadi.

$$sum_i = 0 \quad (9)$$

$$sum_i = sum_i + T[i * k * 2 - 1] \quad (10)$$

4-bosqich. Filtr yadrosi o'lchami $niqob = 2 * r + 1$. Filtrlashning o'rtacha natijasi n_x da qayd etiladi.

Oqim k qanchalik katta bo'lsa, shuncha ko'p keraksiz hisob-kitoblar o'rnatiladi, lekin agar k juda kichik bo'lsa, GPU resurslari behuda ketadi [8]. Shuning uchun, o'rtacha filtrlash algoritmining samaradorligini oshirish uchun oqim konfiguratsiyasi muhim ahamiyatga ega. Ushbu tadqiqot ishida ko'p sonli tajriba natijalarini baholagandan so'ng, har bir blok uchun oqimlar soni 128 qilib belgilandi. Parallel o'rtacha filtr algoritmining blok sxemasi 5-rasmda keltirilgan.



5-rasm. Parallel o'rtacha filtrlash algoritmining blok sxemasi.

Grafik protsessorda tasvirlardan shovqinlarni tozalash va yorqinligini oshirish modeli. Xost va qurilma o'rtasida xotira uzatish butun jarayonning katta qismini egallaydi. Statik tasvirlarni qayta ishlashda bu muhim emas. Biroq, video tasvirlarni qayta ishlashda katta hajmdagi ma'lumotlarni uzatish va ma'lumotlarni uzatishdan kelib chiqadigan kechikishlar mavjud. Bunda filtr yadrosining o'lchami 15×15 qilib o'rnatiladi, ya'ni $r = 7$ ga teng. Tumanli kirish tasvirining intensivligi past bo'lgan kanal tasvirini olish uchun ko'chirilgan filtrlash algoritmiga asoslangan minimum filtrdan foydalanildi. Ushbu jarayonda g_x va h_x massivlarni hisoblaganda kichik massiv oralig'idagi qo'shni piksellarga bog'liqlik mavjud. Bitta kichik massivni qayta ishlash uchun bitta oqimdan foydalanildi va bir nechta oqimlar parallel ravishda ishlaydi, shuning uchun, bitta oqim bir pikselni qayta ishlaydi. Atmosfera yorug'ligi intensivligini aniqlashda tasvirning intensivligi past

Tasvirlarga ishlov berish algoritmining CPU va GPU da bajarilish vaqt sarfini baholash

Tasvir o'lchami	800*600	1280*720	1600*720	1920*1080
CPU	8.071	17.118	21.66	50.74
GPU	1.52	2.6	3.24	5.67
Samaradorlik	5.30	6.58	6.68	8.94

Xulosa o'rnida aytish mumkinki, tasvirning intensivligi past bo'lgan kanali asosida statik tasvirlardan shovqinlarni tozalash algoritmlari yaxshi natijalarni ko'rsatmoqda. Biroq, ushbu algoritmning murakkabligi sababli real vaqt rejimida dinamik tasvirlarga ishlov berish talablariga javob bermaydi. Ushbu tadqiqot ishida kuzatuv kamerasidan olingan tasvirlariga grafik protsessorda parallel ishlov berish modeli va algoritmlari taklif qilindi. O'tkazilgan tadqiqot natijalari 1080p formatdagi video tasvirlarni qayta ishlash imkonini namoyish qildi. Asinxron xotira uzatish va ko'chirilgan filtrlashga asoslangan algoritmlardan foydalangan holda grafik protsessorda bajarish markaziy protsessorga nisbatan 8.94 marta yuqori tezlikka erishdi.

FOYDALANILGAN ADABIYOTLAR:

1. Умаров М.А. OPENMP технологиясидан фойдаланган ҳолда массивларга параллел ишлов бериш // Ўзбекистон Миллий ахборот агентлиги-ЎЗА илм-фан бўлими (электрон журнал), 2020. – №3. – Б. 327-333.
2. Huang, F.; Lan, S.; Wu, J.; Liu, D. Haze removal in real-time based on CUDA. J. Comput. Appl. 2013, 33, -p. 183–186.
3. Jaso Sanders, Edward Kandrot. CUDA by example: An introduction to general-purpose GPU programming // Pearson Education, Inc. 2011. ISBN 978-0-13-138768-3.
4. Jin-Hwan Kim, Won-Dong Jang, Jae-Young Sim, Chang-Su Kim. Optimized contrast enhancement for real-time image and video dehazing // J. Vis. Commun. Image R. 24 (2013) 410–425.

5. Kaiming He, Jian Sun, Xiaoou Tang. Single Image Haze Removal Using Dark Channel Prior // IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 33, No. 12, December 2011.
6. Wei Qi Wu, Keyan Wang, Yunsong Li, Kai Liu, Bormin Huang. Haze Removal Algorithm Using GPU // Remote Sens. 2021, 13, 85. <https://doi.org/10.3390/rs13010085>.
7. Xin Feng, Youni Jiang, Xuejiao Yang, Ming Du, Xin Li. Computer vision algorithms and hardware implementations: A survey // Integration, the VLSI Journal 69 (2019), p.309–320. <https://doi.org/10.1016/j.vlsi.2019.07.005>.
8. Umarov M.A., Jamshid Elov J., Khalilov S., Narzullayev I., Karimov M. An algorithm for parallel processing of traffic signs video on a graphics processor // *International Conference on “Information Science and Communications Technologies (ICISCT 2022)”*. 28-30 September, Toshkent-2022. -5p.