DOI 10.52209/1609-1825_2022_3_358 ӘОЖ 004.93

Бағдарламалық жасақтамаға арналған бейнелердегі басым құрылымдарды бөлектеу алгоритмдерінің бірі

[1]*МУХАМЕТЖАНОВА Бигуль Олжабаевна, PhD, аға оқытушы, grek79@mail.ru,
[1]КАЙБАСОВА Динара Женисбековна, PhD, аға оқытушы, didin@mail.ru,
[1]САЙМАНОВА Загира Бекетаевна, PhD, аға оқытушы, zagira@mail.ru,
[1]СМАГУЛОВА Асемгуль Сериковна, т.ғ.к., доцент, assemgul_work@mail.ru,
[1]КИСИНА Мира Каиржановна, магистр, оқытушы, motya.2002@mai.ru,
[1]«Әбілқас Сағынов атындағы Қарағанды техникалық университеті» КеАҚ, Қазақстан, Қарағанды, Н. Назарбаев даңғылы, 56,
*автор-корреспондент.

Аңдатпа. Бұрыштық нүктелердің ең көп таралған түрлерінің бірі-бейнелердегі басым құрылымдар. Сандық бейнелерді өңдеуде контрастты арттыратын, объектілердің шекараларын баса көрсететін, айқындықты арттыратын, объектілердің шекараларын және оларды сегментациялауды, шу деңгейін төмендетуді жүзеге асыратын әртүрлі ғаламдық және жергілікті кеңістіктік алгоритмдер қолданылады. Жергілікті кеңістіктік алгоритмдер дискретті бейнені жинақтау және жылжымалы маска операциясына негізделген. Басым құрылымдарды бөлу алгоритмін жасау және доминантты және бұрыштық нүктелер детекторымен бағдарламалық жасақтама алгоритмі негізделген.

Кілт сөздер: бейнені өңдеу, жылжымалы терезе, маска, бұрыштарды анықтау, компьютерлік көру, бейнені тану, морфологиялық бейненің әдістері, сегментациялау, бұрыштар, доминантты құрылымдар, ерекше нүктелер.

Кіріспе. Компьютерлік және ақпараттық технологиялар саласындағы соңғы техникалық әзірлемелер, соның ішінде объектілерді тану арқылы нақты әлемді жақсы көруге мүмкіндік береді. Бейнелерді өңдеудің және объектілерді танудың автоматтандырылған жүйелері үшін құрауыштарды өндіруге инновациялық технологияларды енгізу есебінен техникалық үнемі жетілдіріліп отырады, сол арқылы олардың өнімділігін аппараттық деңгейде арттырады. Бірақ бағдарламалық жасақтама деңгейінде ақпаратты өңдеу процестерін, соның ішінде сандық бейнелерді, оның ішінде бейне мазмұнын жедел талдау үшін оңтайландыру қажет.

Python – бұл интерпретацияланған объектіге бағытталған тіл және бағдарламаларды жасауға арналған интерактивті орта [1]. Оның көмегімен графикалық интерфейсі бар қосымшаларды жасай алуға болады, мәліметтер базасымен жұмыс жасай алуға болады, Web-сайттарын құруға және тағы басқалар жасауға болады. Python бағдарламалау тілі айқын және түсінікті синтаксиске ие және математикалық есептеулерді бағдарламалау үшін жақсы. Ғылыми есептеулерде numpy, scipy, matplotlib және sympy қолданылады. Олар теңдеулерді шешудің классикалық сандық алгоритмдерін, сызықтық алгебраның есептерін, белгілі бір интегралдарды есептеуді, жуықтауды, дифференциалдық теңдеулер мен олардың жүйелерін шешуді жүзеге асырады. Matplotlib пакеті екі өлшемді және үш өлшемді деректерді визуализациялаудың жақсы дамыған мүмкіндіктеріне ие. NumPy және SciPy пакеттерінің негізі сандық есептеулер болып табылады, сонымен қатар Sympy кітапханасына негізделген символдық есептеулер Python жақсы жұмыс істей алады [1-2]. Теңдеулер мен жүйелерді шешу, интеграция және саралау, шекті есептеу, қатарларға ыдырау және жинақтау, дифференциалдық теңдеулер мен жүйелердің шешімін іздеу, өрнектерді жеңілдету sympy пакетінің мүмкіндіктері толық.

NumPy – бұл ашық бастапқы Python пакеті. Оны ғылыми және сандық есептеулер үшін пайдалануға болады. Бұл көбінесе массивтерде тиімді есептеу үшін қолданылады. Негізінен біртекті көпөлшемді массивті өңдеу үшін қолданылады. Бұл ғылыми есептеулерге арналған негізгі кітапхана. Демек, ол көп өлшемді массивтің қуатты нысандарына және осы массивтермен жұмыс жасау кезінде пайдалы интегралдық құралдарға ие. Бұл кез-келген Python ғылыми бағдарламалауында маңызды, оған машиналық оқыту, статистика, биоинформатика және т.б. кіреді, ол негізінен төменгі деңгейдегі тілдердегі массивтерге өте ұқсас іргелес массивтерде математикалық операцияларды орындауға бағытталған.

Кітапхананың көптеген функциялары жүйеде файл түрінде ұсынылған [2]. Олар бейнелерді өңдеудің ең көп қолданылатын алгоритмдерін оңтайлы қолданады. Python-ның ерекшелігі – матрицалық операцияларға бағдарлау. Python жүйесінің артықшылықтары:

- матрицалармен жұмыс жасау үшін операциялардың көп саны. Бұл сызықтық алгебралық теңдеулер жүйесін шешуге байланысты әртүрлі есептерді бағдарламалауды да, бейнелермен жұмыс істеуді де жеңілдетеді;
- әр түрлі математикалық әдістердің үнемі дамып келе жатқан кітапханаларының бай жиынтығы (сандық әдістер, дифференциалдық теңдеулер, сигналдарды өңдеу, есептеу геометриясы және т.б.);
- интернетте көптеген қосымша кітапханалар мен мүмкіндіктер бар;
- әр түрлі деректерді (екі өлшемді және үш өлшемді функциялар, жиындар және т.б.) жылдам және қарапайым (әзірлеуші үшін) визуализациялаудың үлкен мүмкіндіктері;
- үнемі қол жетімді командалық жолы бар аудармашы бар (қажетті уақытта бағдарламаны белгілі бір қадамда алынған деректерді кідіртуге және көруге, оларды графикалық түрде көрсетуге, өзгертуге, шақырылған функцияның параметрлерін түзетуге және т.б.), бұл бағдарламадағы қателерді түзетуді және түсінуді жеңілдетеді;

Python-ды бағдарламаларда қолдану мүмкіндігі бар, осылайша алгоритмдерді Python ортасының көмегімен тексеруге және өз бағдарламаларында Python құралдарын қолдануға болады [1] (деректерді визуализациялау және басқа да күнделікті операциялар). Django – бұл Python бағдарламалау тілінде жазылған күрделі сайттар мен веб-қосымшаларды жасауға жарамды бай мүмкіндіктері бар бағдарламалық жасақтама.

Django – Python тіліндегі веб-қосымшаларға арналған шеңбер. Django Apache (mod_python модулімен) және PostgreSQL-ді дерекқор ретінде пайдалану үшін жасалған. Қазіргі уақытта PostgreSQL-ден басқа, Django басқа ДҚБЖ-мен жұмыс істей алады: MySQL (MariaDB), SQLite, Microsoft SQL Server, DB2, Firebird, SQL Anywhere және Oracle. Деректер базасымен жұмыс істеу үшін Django өзінің ORM қолданады, онда деректер моделі Python сыныптарымен сипатталады және оған сәйкес мәліметтер базасының схемасы жасалады.

Бастапқыда, Django-ның дамуы архитектурада айтарлықтай көрініс тапқан жаңалықтар ресурстарымен ыңғайлы жұмыс істеуді қамтамасыз ету үшін жүргізілді: құрылым ақпараттық сипаттағы веб-сайттарды тез дамытуға көмектесетін бірқатар құралдарды ұсынады. Мысалы, әзірлеушіге сайттың әкімшілік бөлігі үшін контроллерлер мен беттер жасаудың қажеті жоқ, Django-да жасалған кез-келген сайтқа қосылатын және бір серверде бірден бірнеше сайтты басқара алатын кірістірілген мазмұнды басқару қосымшасы бар. Әкімшілік қосымша барлық жасалған әрекеттерді тіркеп, сайтты толтырудың кез келген объектілерін жасауға, өзгертуге және жоюға мүмкіндік береді және пайдаланушылар мен топтарды басқаруға арналған интерфейсті ұсынады.

Django веб – Instagram, Disqus, Mozilla, The Washington Times, Pinterest, lamoda және т.б. құрлымдарды сияқты ірі және танымал сайттарда қолданылады.

Зерттеу әдістері. Бейнедегі доминантты құрылымдарды бөлу анықтамасы, бұл мысал жеделдетілген сегментке негізделген функцияларды тестілеу алгоритмімен бұрышты қалай анықтауға болатындығын көрсетеді [3]. Бұл алгоритм Харрис бұрышын анықтауға балама болып табылады. Жылдам алгоритм бұрыштың потенциалды центрінің айналасындағы дөңгелек аймақты тексеру арқылы бұрыштың бар-жоғын анықтайды [4]. Егер пиксельдің іргелес бөлігі центрден және шекті мәннен ашық болса немесе центр минус шекті мәннен күңгірт болса, тест бұрышты анықтайды.

Бағдарламалық қамтамасыздандыру да бұл алгоритм осьтер бойымен тек төрт пиксельді тексеріп, ықтимал бұрыштарды алып тастау үшін жылдам тексеруге мүмкіндік береді. Бағдарламалық алгоритмдер толық мәтінді жылдам тест сәтті болған жағдайда ғана орындайды. Аппараттық құралдарды іске асыру барлық сынақтарды параллель түрде оңай жасай алады, сондықтан жылдам тест әсіресе тиімді емес және осы мысалға кірмейді [5-6].

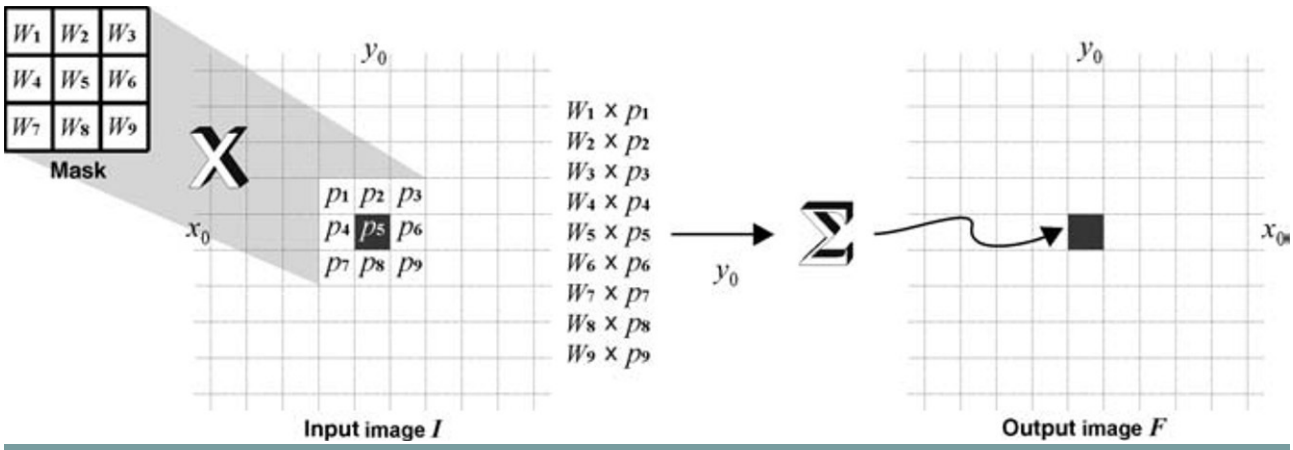
Жылдам алгоритмді көптеген мөлшерде немесе масштабта қолдануға болады. Бұл мысалда бұрыштар он алты пиксель шеңберімен анықталады. Осы он алты пиксельде, егер кез-келген он екі іргелес пиксель ашық немесе күңгірт шегіне сәйкес келсе, онда бұрыш ашылады. Python-да бұрыштарды жылдам анықтау. Компьютерлік көру жүйесі объектілерді жылдам анықтау функциясында бұрыштарды жылдам анықтаудың бағдарламалық алгоритмін қамтиды. Функцияда минималды контраст пен минималды сапаны орнатудың параметрлері бар.

Алгоритмнің жылдам шектеулері, бұрыштарды анықтаудың басқа әдістері FAST әдісі сияқты жұмыс істемейді, ал күтпеген нәтиже – FAST x және y осьтеріне жақсы сәйкес келетін компьютер жасаған кескіндердегі бұрыштарды анықтамайды. Анықталған бұрышта бұрыштың екі жиегін қамтитын ортасында күңгірт немесе ашық пиксель мәндерінің сақинасы болуы керек болғандықтан, айқын бейнелер жақсы жұмыс істемейді. Мысалы, Харрис бұрышын анықтау мысалында қолданылатын кіріс бейнедегі жылдам алгоритмді қолданады [7].

Екі өлшемді аймақта жинақтау.

Екі өлшемді жинақтаудың математикалық анықтамасы келесідей:

$$g(x,y) = \sum_{k=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h(j,k) \cdot f(x-j, y-k). \qquad (1)$$



1-сурет – Сызықты сүзудің ерекше жағдайы ретінде фрагменттермен өңдеу кіріс бейнесі I; өңдеу шығысындағы F бейнесі

Іс жүзінде ол сәйкес келеді:

$$g(x,y) = \sum_{k=-n_2}^{n_2} \sum_{j=-m_2}^{m_2} h(j,k) \cdot f(x-j, y-k). \qquad (2)$$

мұндағы m_2 маска енінің жартысына тең, ал n_2 маска биіктігінің жартысына тең, яғни,

$$m_2 = \left[\frac{m}{2}\right], \qquad (3)$$

$$n_2 = \left[\frac{n}{2}\right]. \qquad (4)$$

Python бейне өңдеу құралдарының жиынтығында 2D жинақтауды жүзеге асыру үшін қолдануға болатын екі функция бар:

- conv2: ол екі матрица арасындағы 2D түйінін есептейді. Екі матрицадан басқа, ол шығыс мөлшерін анықтайтын үшінші параметрді қабылдайды;
- толық: толық 2D түйінін қайтарады;
- same: а өлшемімен бірдей жиынтықтың орталық бөлігін қайтарады;
- valid: нөлдік жиектерсіз есептелген жинақтау бөліктерін ғана қайтарады.

Бейнедегі әр масканы сегіз бейне жасау арқылы бүктеу керек [8]. Бұл бейнелерді үш өлшемді матрицада сақтап, оны маскалар үшін жасадық. Барлық маскалар бір матрицада сақталғандықтан, барлық сегіз әрекетті кодтың аз жолдарымен орындау үшін циклды қолдана аламыз.

Әр масканы for циклын пайдаланып бейнемен жинақтаймыз.

```
# 8 масканың әрқайсысымен фрагментті жинақтау
        for s in range(0, 8):
          # Уақытша жұмыс массивін толтыру
          Mask[:,:] = Mask09[s, :, :]
          # S-ші масканы фрагментпен жинақтау
          Conv[s] = convolute(W, Mask, n)
```

Әр пиксель үшін барлық бейнелердің максималды мәнін табамыз. Тағы да, олардың барлығы бір матрицада сақталғандықтан, біз оны кодтың бір жолында жасай аламыз.

Conv массивінің максималды модульдерін есептеу, фрагменттің орташа мәнін қалыпқа келтіру және осы мәнді edges шекаралары мен бұрыштарының массивіне енгізу (i,j):

```
Edges[i, j] = np.max(np.abs(Conv)) / (w0 + 0.0001)
```

Алдыңғы қадамда дисплей мақсаттары үшін нәтиже артты. Егер бейненің шекті мәнін орнатқымыз келсе (барлық алдыңғы шекара детекторлары сияқты), алдымен оның мәндері [0, 255] диапазонында болатындай етіп бейнені масштабтау керек. Ол үшін барлық ағымдағы мәндерді қажетті диапазондағы мәндермен салыстыратын сызықтық түрлендіру функциясын жасай аламыз. Бейнені сұр реңктердің диапазонында көрсету үшін түрлендіру функциясын және түрлендіру қажет.

Динамикалық диапазонның созылу (қысу) коэффициенті:

```
coeff = (255 – MinEdge) / (MaxEdge – MinEdge)
Edges = np.uint8(coeff * Edges)
Threshold = threshold
    print(«Сурет шегі:», Threshold, MaxEdge, MinEdge)
      for i in range(0, M):
        for j in range(0, N):
          if Edges[i, j] > 160:
            Corners[i, j] = 255
            rectangle('Position',[i,j,10,10],'EdgeColor','r','Linewidth',3)
```

Жиектерді анықтау – адамның көру жүйесінде бар мүмкіндікті имитациялауға тырысатын бейнені өңдеудің негізгі операциясы. 2D сұр реңкті бейнелердегі жиектер әдетте қарқындылық функциясының күрт өзгеруі ретінде анықталады [8]. Жалпы алғанда, жиекті кейбір мүмкіндіктерге (мысалы, сұр деңгей, түс немесе текстура) сәйкес әртүрлі сипаттамаларға ие бейненің екі аймағы арасындағы шекара ретінде анықтауға болады. Жиектерді анықтау көптеген бейнелерді

өңдеу әдістерінің негізгі қадамы болып табылады: жиектер анықталғаннан кейін, сол жиектермен қоршалған аумақтар сегменттерге бөлінеді және сәйкесінше өңделеді. Бейнені өңдеу әдебиетінде жиекті анықтаудың көптеген әдістері бар. Олар қарапайым конволюционды маскалардан (мысалы, Собель және Приютт) [7] биологиялық шабыттандырылған әдістерге (мысалы, Марр-Хильдрет) дейін және олар шығаратын нәтижелердің сапасы айтарлықтай өзгереді. Canny жиегі детекторы жиекті анықтаудың ең танымал заманауи әдісі болып табылады. Python-да Pruitt, Sobel, Gaussian Laplacian және Canny сияқты жиекті анықтаудың бірнеше әдістерін жүзеге асыратын жиек функциясы бар. Жиекті анықтау алгоритмінің нәтижелері әдетте қажетсіз нүктелерді жояды, бос орындарды жояды және жиектерге негізделген бейнені сегменттеу шешімінің келесі қадамдарында пайдаланылатын таза жиектерге әкелетін жиекті байланыстыру алгоритмімен өңделеді. Хафмен түрлендіруі жиекті анықтау нәтижелерінде ұзын түзу жиектерді табу үшін кеңінен қолданылатын әдіс болып табылады.

Кирштің бастапқы жиектерін анықтау. Кирш компас ядросы – бірнеше алдын ала анықталған бағыттардағы максималды жиектің күшін анықтайтын сызықты емес жиекті табу. Ол компьютер ғалымы Рассел А.Кирштің атымен аталған [8]. Көптеген дифференциалды маскалар немесе екі өлшемді түйіннің дискретті ядролары белгілі [9]. Оператор бір негізгі масканы алып, оны барлық 8 компас бағытында 45 градус қадаммен бұрады: солтүстік, шығыс, оңтүстік, батыс, солтүстік-батыс, оңтүстік-батыс, оңтүстік-шығыс және солтүстік-шығыс (2-сурет). Кирш маскалары келесідей анықталады.

Кирш операторының шекаралық амплитудасы барлық бағыттағы максималды мән ретінде есептеледі [10]. Белгілі бір пикселге арналған жиектер туралы ақпарат осы пикселге жақын орналасқан пикселдердің жарықтығын зерттеу арқылы алынады. Егер маңайдағы барлық пикселдердің жарықтығы бірдей болса, онда бұл нүктеде жиектер болмауы мүмкін. Дегенмен, көршілердің кейбіреулері басқаларға қарағанда әлдеқайда жарқын болса, онда бұл нүктеде жиек болуы мүмкін. Кирштің жиекті анықтау алгоритмі туындыларды есептеу кезінде пикселді және оның көршілерін сақтау үшін 3×3 пиксельдік матрицаны пайдаланады. 3×3 пиксельді кесте жиынтық кесте деп аталады, себебі ол жинақтау терезесі түріндегі алгоритмде бейне бойынша шарлайды.

Ұсынылған алгоритм өңдеу уақытын азайтуға және шектелген жиектерді табуға тырысады. Сондықтан ол С, О, Ш, Б, СБ және ОБ сияқты жоғарыда көрсетілген алты Кирш шекаралық операторын пайдаланады. С және О тік сызықтарды анықтауға көмектеседі, Ш және Б көлденең сызықтарды анықтауға көмектеседі. Солтүстік – батыс және оңтүстік – батыс қиғаш сызықтарды анықтай алады.

Ғылыми нәтижелері. Басым құрылымдарды анықтау алгоритмдерін жасау және проблемаларды шешудің тиімді құралы болып табылатын сан-

$K^1 =$ | -3 | -3 | 5 |
|---|---|---|
| -3 | 0 | 5 |
| -3 | -3 | 5 |

, $K^2 =$ | 5 | 5 | 5 |
|---|---|---|
| -3 | 0 | -3 |
| -3 | -3 | -3 |

, $K^3 =$ | 5 | -3 | -3 |
|---|---|---|
| 5 | 0 | -3 |
| 5 | -3 | -3 |

, $K^4 =$ | -3 | -3 | -3 |
|---|---|---|
| -3 | 0 | -3 |
| 5 | 5 | 5 |

$K^5 =$ | -3 | 5 | 5 |
|---|---|---|
| -3 | 0 | 5 |
| -3 | -3 | -3 |

, $K^6 =$ | 5 | 5 | -3 |
|---|---|---|
| 5 | 0 | -3 |
| -3 | -3 | -3 |

, $K^7 =$ | -3 | -3 | -3 |
|---|---|---|
| 5 | 0 | -3 |
| 5 | 5 | -3 |

, $K^8 =$ | -3 | -3 | -3 |
|---|---|---|
| -3 | 0 | 5 |
| -3 | 5 | 5 |

2-сурет – 45 градус бұрыштар үшін 3×3 өлшемді Кирш маскалары

3-сурет – Бейнедегі доминантты құрылымдарды бөлу

дық бейнелердегі доминантты және бұрыштық нүктелерді бөлектеу арқылы бағдарламалық жасақтамасы әзірленді. Бұл ұсынылған жұмыс шекті мәнінің орнына максималды қарқындылық мәнін таңдайды, өйткені шекті мән бейненің қарқындылығына байланысты. Бұл тек күшті жиектерді анықтауға көмектеседі маскамен сканерленген деректерді үлкен маскалармен есептеулерде қолдану қиынға соғады. Өлшемдердің біртіндеп өсетін маскаларымен есептеулер сызықтық және аудан параметрлері және бұрыштың фондық аймаққа өту сәті туралы ақпаратты қамтиды. Кирш маскаларымен салыстырғанда масштабталатын маскалар [11] бұрыштық құрылым мен фон арасындағы шекара белгілі бір пиксельдің ішінде оның ортасынан өтеді, ал екі іргелес пиксель олардың жақтары мен шеттерінде емес.

Қорытынды. Масштабталатын детекторларды салуға ыңғайлы бейнедегі доминантты құрылымдарды анықтауға арналған маска матрицаларың қолданып бағдарламадағы алгоритмнің нәтижесін көрсеттік. Ұсынылған тәсілмен анықталатын құрылымдарға екі өлшемді массивтің ортасында шыңдары бар бұрыштар жатады. Бұл құрылымдар бейнелер массивін қайта өту үшін жылжымалы терезені қажет етпейді, өйткені біріктіру шекаралармен немесе периметрлермен жұмыс істейтін төменгі деңгейдегі негізгі процедураларға негізделген шкаланың немесе иерархияның келесі деңгейінде жүреді [11]. Бұл әдісті дәстүрлі Кирш маскаларымен және жылдам бейнені талдау алгоритмімен қатар қолдануға болады, бірақ маскалардың мөлшерін таңдауға шектеусіз.

ӘДЕБИЕТТЕР ТІЗІМІ

1. Лутц М. Программирование на Python, том II, 4-е издание. – Пер. с англ. – СПб: Символ-Плюс, 2011. – 992 с.
2. Федоров, Д.Ю. Программирование на языке высокого уровня Python: учеб. пособие для прикладного бакалавриата / Д.Ю. Федоров. – 2-е изд., перераб. и доп. – М.: Издательство Юрайт, 2019. – 161 с.
3. Gonzalez R.C. and Woods R.E., Digital Image Processing. NY: Pearson, 2018.
4. W. Zhang, C. Sun, T. Breckon, and N. Alshammari, «Discrete curvature representations for noise robust image corner detection» IEEE Trans. on Image Processing, vol. 28, pp. 4444-4459, 2019.
5. A. Dutta, A. Kar, and B. N. Chatterji, «Corner Detection Algorithms for Digital Images in Last Three Decades» Institution of Electronics and Telecommunication Engineers Technical Review, vol. 25, pp. 123-133, 2008.
6. J. Chen, L. Zou, J. Zhang, and L. Dou, «The Comparison and Application of Corner Detection Algorithms» Journal of Multimedia, vol. 4, pp. 435-441, 2009.
7. Мухаметжанова Б.О., Сагатбекова Д.Е. Сандық бейнелердегі доминантты құрылымды анықтау әдістері Вестник АУЭС, 2020. – № 3 (50). – С. 77-82.
8. Мухаметжанова Б.О., Искаков К.Т., Олейникова А.В. Бейнелерді өңдеудің және танудың сандық әдістері – Вестник КазАТК, Алматы, 2021. – 2 (117). – С. 69-76.
9. Rosten E., Porte R., and Drummond T., «Bystree ı luchshe: podhod mashınnogo obucheniıa k obnarujeniıu uglov» [Faster and Better: A Machine Learning Approach to Corner Detection], IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, 2010. pp. 105-119.
10. Kazantsev I.G., Mukhametzhanova B.O., Suvorovsky O.Y. Corner Detection Based on Scalable Masks // 2021 IEEE International Conference on Smart Information Systems and Technologies (SIST). Nur-Sultan, Kazakhstan – 2021, DOI:10.1109 / SIST50301.2021.9465940.
11. Kazantsev I.G., Mukhametzhanova B.O., Iskakov K.T., and Mirgalikyzy T., «Detection of the Corner Structures in Images by Scalable Masks» Journal of Applied and Industrial Mathematics, vol. 14, pp. 73-84, 2020.

Один из алгоритмов выделения доминантных структур для программного обеспечения

*[1]**МУХАМЕТЖАНОВА Бигуль Олжабаевна,** PhD, старший преподаватель, grek79@mail.ru,*
*[1]**КАЙБАСОВА Динара Женисбековна,** PhD, старший преподаватель, didin@mail.ru,*
*[1]**САЙМАНОВА Загира Бекетаевна,** PhD, старший преподаватель, zagira@mail.ru,*
*[1]**СМАГУЛОВА Асемгуль Сериковна,** к.т.н., доцент, assemgul_work@mail.ru,*
*[1]**КИСИНА Мира Каиржановна,** магистр, преподаватель, motya.2002@mai.ru,*
[1]НАО «Карагандинский технический университет имени Абылкаса Сагинова», Казахстан, Караганда, пр. Н. Назарбаева, 56,

**автор-корреспондент.*

***Аннотация.** Одним из наиболее распространенных типов особых точек являются углы на изображении. Расположение углов можно определить, используя локальные детекторы. Входом локальных детекторов является черно-белое изображение. На выходе формируется матрица с элементами, значения которых определяют степень правдоподобности нахождения угла в соответствующих пикселях изображения. Выполняется отсечение пикселей со степенью правдоподобности, меньшей некоторого порога. Для оставшихся точек принимается, что они являются особыми. Маски выделения углов на изображениях, применяемые при обра-*

ботке скользящим по изображению окном. Обоснована разработка алгоритма распределения доминантных структур и программного алгоритма с детектором доминантных и угловых точек.

Ключевые слова: *обработка изображений, скользящее окно, маска, определение углов, компьютерное зрение, обработка видео, распознавание изображений, методы изображения, сегментация, углы, доминантные структуры, особые точки.*

One of the Algorithms for Identifying Dominant Structures for Software

[1]***MUKHAMETZHANOVA Bigul,** PhD, Senior Lecturer, grek79@mail.ru,*
[1]***GABBASOVA Dinara,** PhD, Senior Lecturer, didin@mail.ru,*
[1]***SAIMANOVA Zagira,** PhD, Senior Lecturer, zagira@mail.ru,*
[1]***SMAGULOVA Asemgul,** Cand. of Tech. Sci., Associate Professor, assemgul_work@mail.ru,*
[1]***KISINA Mira,** master, Teacher, motya.2002@mai.ru,*
[1]NPJSC «Abylkas Saginov Karaganda Technical University», Kazakhstan, Karaganda, N. Nazarbayev Avenue, 56,
**corresponding author.*

Abstract. *One of the most common types of singular points are angles in the image. The location of the corners can be determined using local detectors. The input of the local detectors is a black-and-white image. At the output, a matrix is formed with elements whose values determine the degree of plausibility of finding the angle in the corresponding pixels of the image. Pixels are clipped with a degree of plausibility less than a certain threshold. For the remaining points, it is assumed that they are special. Masks for highlighting corners in images that are used when processing with a sliding window on the image. Unlike known algorithms, large-size mask matrices are constructed by simply adding rows and columns of smaller masks, leaving the submatrices unchanged.*

Keywords: *image processing, sliding window, mask, angle detection, computer vision, video processing, image recognition, image methods, segmentation, angles, dominant structures, singular points.*

REFERENCES

1. Lutc M. Programmirovanie na Python, tom II, 4-e izdanie. – Per. s angl. – Saint Petersburg: Simvol-Plyus, 2011. – 992 p.
2. Fedorov, D.YU. Programmirovanie na yazyke vysokogo urovnya Python: ucheb. posobie dlya prikladnogo bakalavriata / D.YU. Fedorov. – 2-e izd., pererab. i dop. – Moscow: Publ. YUrajt, 2019. – 161 p.
3. Gonzalez R.C. and Woods R.E., Digital Image Processing. NY: Pearson, 2018.
4. W. Zhang, C. Sun, T. Breckon, and N. Alshammari, «Discrete curvature representations for noise robust image corner detection» IEEE Trans. on Image Processing, vol. 28, pp. 4444-4459, 2019.
5. A. Dutta, A. Kar, and B. N. Chatterji, «Corner Detection Algorithms for Digital Images in Last Three Decades» Institution of Electronics and Telecommunication Engineers Technical Review, vol. 25, pp. 123-133, 2008.
6. J. Chen, L. Zou, J. Zhang, and L. Dou, «The Comparison and Application of Corner Detection Algorithms» Journal of Multimedia, vol. 4, pp. 435-441, 2009.
7. Mukhametzhanova B.O., Sagatbekova D.E. Methods of determining the dominant structure in digital images Vestnik AUES, 2020. – No. 3 (50). – pp. 77-82.
8. Mukhametzhanova B.O., Iskakov K.T., Oleynikova A.V. Quantitative methods of image processing and recognition – Vestnik KazATK, Almaty, 2021. – 2 (117). – pp. 69-76.
9. Rosten E., Porte R., and Drummond T., «Bystree ı luchshe: podhod mashınnogo obucheniıa k obnarujeniıu uglov» [Faster and Better: A Machine Learning Approach to Corner Detection], IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, 2010. pp. 105-119.
10. Kazantsev I.G., Mukhametzhanova B.O., Suvorovsky O.Y. Corner Detection Based on Scalable Masks // 2021 IEEE International Conference on Smart Information Systems and Technologies (SIST). Nur-Sultan, Kazakhstan – 2021, DOI:10.1109 / SIST50301.2021.9465940.
11. Kazantsev I.G., Mukhametzhanova B.O., Iskakov K.T., and Mirgalikyzy T. «Detection of the Corner Structures in Images by Scalable Masks» Journal of Applied and Industrial Mathematics, vol. 14, pp. 73-84, 2020.