

RAQAMLI TASVIRNI QAYTA ISHLASH USULLARINING TAHLILI

Beknazarova Saida Safibullayevna

Professor, Toshkent axborot texnologiyalari universiteti

E-mail: saida.beknazarova@gmail.com

Abduraxmanova Ro'zixon Xasanboy qizi

tayanch doktorant, Namangan davlat universiteti

E-mail: rozixonabduraxmanova0601@gmail.com

<https://doi.org/10.5281/zenodo.20956690>

Annotatsiya: Ushbu maqolada raqamli tasvirlarni qayta ishlashning ikkita muhim yo'nalishi – affin almashtirish usullari va segmentatsiyalash usullari tahlil qilingan. Xususan, tasvirni burchak bo'yicha burish va masshtablash usullarining matematik asoslari, algoritmik yechimlari hamda OpenCV dasturiy kodi va natijalari keltirilgan. Maqolada segmentatsiyalash usullarining tasnifi, birjinslilik predikati tushunchasi, soha o'stirish algoritmlari hamda fizik xususiyatlar asosida segmentatsiyalashga misol sifatida odam teri rangini ajratib olish masalasi OpenCV dasturiy kodi bilan yoritilgan. Maqola natijalari tasvirlarni qayta ishlash sohasida tadqiqot olib borayotgan mutaxassislar, muhandislar va talabalar uchun amaliy ahamiyatga ega.

Kalit so'zlar: tasvirlarni qayta ishlash, affin almashtirish, burchak bo'yicha burish, masshtablash, segmentatsiya, bo'laklash, soha o'stirish, OpenCV.

АНАЛИЗ МЕТОДОВ ОБРАБОТКИ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ

Бекназарова Саида Сафибуллаевна

Профессор Ташкентского университета информационных технологий.

E-mail: saida.beknazarova@gmail.com

Абдурахманова Рузихон Хасанбой кизи

базовый докторант, Наманганский государственный университет

E-mail: rozixonabduraxmanova0601@gmail.com

Аннотация: В статье проанализированы два важных направления цифровой обработки изображений: методы аффинных преобразований и методы сегментации. В частности, представлены математические основы, алгоритмические решения, программный код OpenCV и результаты методов поворота изображения на заданный угол и масштабирования. Также освещены классификация методов сегментации, понятие предиката однородности, алгоритмы выращивания областей и задача выделения цвета кожи человека как пример сегментации по физическим свойствам с использованием программного кода OpenCV. Результаты статьи имеют практическое значение для специалистов, инженеров и студентов, проводящих исследования в области обработки изображений.

Ключевые слова: обработка изображений, аффинное преобразование, поворот по углу, масштабирование, сегментация, разбиение, выращивание областей, OpenCV.

ANALYSIS OF DIGITAL IMAGE PROCESSING METHODS

Beknazarova Saida Safibullayevna

Professor, Tashkent University of Information Technologies

E-mail: saida.beknazarova@gmail.com

Abdurakhmanova Rozikhon Khasanboy qizi

PhD student, Namangan State University

E-mail: rozixonabduraxmanova0601@gmail.com

Abstract: This article analyzes two important areas of digital image processing: affine transformation methods and segmentation methods. In particular, the mathematical foundations, algorithmic solutions, as well as the OpenCV software code and results of image rotation and scaling methods are presented. The article covers the classification of segmentation methods, the concept of homogeneity predicate, region growing algorithms, and the issue of identifying human skin color as an example of segmentation based on physical properties using OpenCV software code. The results of the article are of practical importance for specialists, engineers, and students conducting research in the field of image processing.

Keywords: image processing, affine transformation, angular rotation, scaling, segmentation, fragmentation, region growing, OpenCV.

KIRISH

Raqamli tasvirlarni qayta ishlash sohasi bugungi kunda sun'iy intellekt, kompyuter ko'rishi, tibbiy diagnostika, robototexnika, videokuzatuv tizimlari va sanoat avtomatikasining ajralmas qismiga aylangan. Har kuni milliardlab tasvirlar raqamli kameralar, smartfonlar, tibbiy asboblarni, sun'iy yo'ldoshlar va sanoat tizimlari orqali yaratilmoqda. Ushbu tasvirlarni samarali tahlil qilish, ulardan kerakli ma'lumotlarni ajratib olish va ularni keyingi ishlov berish jarayonlariga tayyorlash dolzarb ilmiy-texnik vazifalardan hisoblanadi.

Tasvirlarni qayta ishlashning **asosiy maqsadi** – raqamli tasvirlardan kerakli ma'lumotlarni ajratib olish, ularning vizual sifatini yaxshilash yoki keyingi tahlil uchun qulay shaklga keltirishdir. Bu yo'nalish tibbiyotda rentgen va MRT tasvirlarini tahlil qilish, sanoatda nuqsonlarni aniqlash, qishloq xo'jaligida ekinlarni monitoring qilish, xavfsizlik tizimlarida yuz va obyektlarni tanish, avtonom transport vositalarida atrof-muhitni idrok etish kabi sohalar kesimida keng qo'llaniladi.

MATERIAL VA METODLAR

Tasvirlar ustida affin almashtirish usullari. Qo'yilgan masalalarni yyechish jarayonida ba'zan tasvir ustida affin almashtirishlarini amalga oshirishga to'g'ri keladi. Affin almashtirishlari uchun bir qancha usullar mavjud. Quyida ularning ichida ikki o'lchovli tasvirlar uchun eng ko'p foydalaniladigan usullarni ko'rib chiqamiz.

Tasvirni burchak bo'yicha burish. Masalaning mohiyatidan kelib chiqib, tasvirlarni qayta ishlash jarayonida tasvirni biror burchak ostiga burish talab etiladi. Tasvirni burchak bo'yicha burish formulasi quyidagi ko'rinishga ega.

$$x' = x \cdot \cos \alpha - y \cdot \sin \alpha,$$

$$y' = x \cdot \sin \alpha + y \cdot \cos \alpha.$$

Bu yerda - burilish burchagi; - tasvirning burilishi kutilayotgan piksel koordinatasi; - pikselning yangi koordinatasi.

Demak, bu holatda biror burchak ostiga burilishi kerak bo'lgan tasvirning har bir pikseli koordinatasi yuqoridagi formula bo'yicha aniqlanadi va ana shu yangi koordinataga piksel ko'chirib o'tkaziladi. Ta'kidlash kerakki, pikselni ko'chirib o'tkazishda uning rang qiymatiga daxl qilinmaydi, faqat uning joylashuv koordinatasi o'zgartiriladi.

OpenCV dasturiga quyidagi dasturiy kodlarni kiritish orqali affin almashtirish usullaridan biri bo'lgan tasvirni burchak bo'yicha burish usuli korib chiqiladi :

```
#include "mainwindow.h"
```

```
#include "ui_mainwindow.h"
```

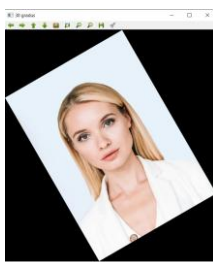
```
#include <opencv2/opencv.hpp>
#include "math.h"
#include "qdebug.h"
using namespace cv;
MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
}
MainWindow::~MainWindow()
{
    delete ui;
}
Mat tasvir = imread("d://1.jpg", 1);
if(tasvir.empty()) {
    qDebug() << "Rasm topilmadi!";
    return;
}
imshow("Original", tasvir);
double burchaklar[] = {30.0, 45.0, 90.0};
double scale = 1.0; // O'lchamni o'zgartirmaslik uchun
for(int i = 0; i < 3; i++) {
    Point center = Point(tasvir.cols/2, tasvir.rows/2);
    double angle = burchaklar[i];
    Mat rot_mat = getRotationMatrix2D(center, angle, scale);
    Rect bbox = RotatedRect(center, tasvir.size(), angle).boundingRect();
    rot_mat.at<double>(0,2) += bbox.width/2.0 - center.x;
    rot_mat.at<double>(1,2) += bbox.height/2.0 - center.y;
    Mat aylantirilgan;
    warpAffine(tasvir, aylantirilgan, rot_mat, bbox.size());
    QString oyna_nomi = QString("%1 gradus").arg(angle);
    imshow(oyna_nomi.toStdString(), aylantirilgan);
}
waitKey(0);
}
```

NATIJARLAR VA MUHOKAMA

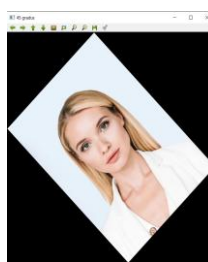
Tasvirlarni turli burchak ostida burish natijalari 1-rasmda keltirilgan.



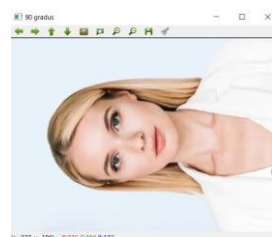
a)



b)



c)

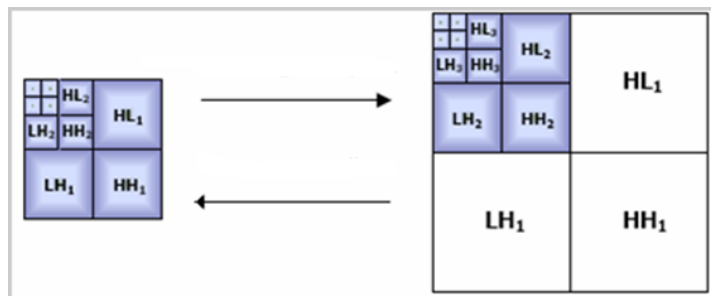


d)

1-rasm. Tasvirni burish natijalari. a) berilgan tasvir; b) 30° ga burilgan;

c) 45° ga burilgan; d) 90° ga burilgan.

Tasvirni masshtablash. Tasvirlarni qayta ishlashda uning egallagan hajmi katta ahamiyatga ega. Har qanday kattalikdagi tasvirlar uchun umumiy bo'lgan algoritmlarni qurish ancha murakkab ish hisoblanadi. Shu bilan birga katta hajmli tasvirlarni qayta ishlash uchun ko'p vaqt sarflanadi. Shu sababli, ko'pincha tasvir hajmini kichraytirish talab etiladi. Bundan asosiy maqsad algoritmlar ishlash vaqtini kamaytirish orqali tizimning ishlash tezligini oshirishdir. Oddiy usulda darajani kamaytirish yoki orttirish yo'li bilan tasvirni masshtablash sxemasi 2-rasmda ko'rsatilgan.



2-rasm. Tasvirni masshtablashning umumiy sxemasi.

Umumiy holda tasvirni masshtablashning formula ko'rinishini quyidagicha ifodalash mumkin:

$$X' = X \cdot S_x, \quad Y' = Y \cdot S_y,$$

yoki matritsa shaklida:

$$P' = P \cdot S,$$

bu yerda S - masshtablash koeffitsienti; – tasvirda qaralayotgan pikselning dastlabki koordinatasi; - pikselning yangi koordinatasi.

Demak, bu holatda agar $S < 1$ dan kichik bo'lsa tasvir kichrayadi, aksincha, agar katta bo'lsa, tasvir kattalashadi.

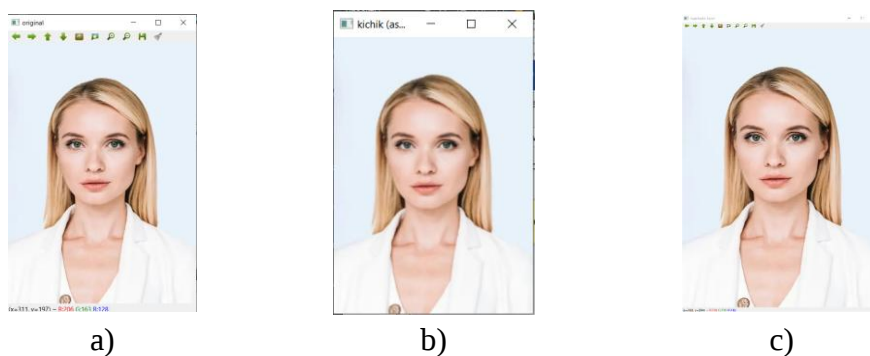
Bu yerda shuni ta'kidlash kerakki, tasvirni kichraytirish jarayonida bitta yangi koordinata uchun bir nechta piksellar mos tushib qolishi mumkin. Bu holatda ularning o'rtacha rang qiymatini olish maqsadga muvofiqdir. Aksincha, tasvirni kattalashtirilganda dastlabki tasvirning bitta pikseli yangi (kattalashayotgan) tasvirda bir nechta qo'shni bo'lgan piksellarga mos tushishi mumkin. Bunda o'sha bitta piksel rangi bir nechta piksellarda o'z aksini topadi. Bunday holat, agar masshtablash koeffitsienti juda katta bo'lsa, tasvir sifatining tushib ketishiga sabab bo'lishi ham mumkin.

OpenCV dasturiga quyida berilgan kodlarni kiritish orqali tasvir masshtablanadi:

```
destroyAllWindows();
Mat tasvir = imread("d://1.jpg", 1);
if(tasvir.empty()) {
    qDebug() << "Rasm topilmadi!";
    return;
}
imshow("original", tasvir);
qDebug() << "Original:" << tasvir.cols << tasvir.rows;
Mat katta;
cv::resize(tasvir, katta, Size(), 2, 2, INTER_CUBIC);
imshow("masshtabli tasvir", katta);
Mat kichik;
```

```
cv::resize(tasvir, kichik, Size(), 0.5, 0.5, cv::INTER_AREA);
qDebug() << "Kichik:" << kichik.cols << kichik.rows;
Mat kichik_rgb;
cvtColor(kichik, kichik_rgb, COLOR_BGR2RGB);
QImage img(kichik_rgb.data, kichik_rgb.cols, kichik_rgb.rows,
    kichik_rgb.step, QImage::Format_RGB888);
QLabel *label = new QLabel();
label->setWindowTitle("kichik (aspekt saqlangan)");
QPixmap pix = QPixmap::fromImage(img);
label->setPixmap(pix.scaled(400, 400, Qt::KeepAspectRatio, Qt::SmoothTransformation));
label->resize(label->pixmap()->size());
label->show();
waitKey(0);
}
```

OpenCV dasturida tasvirni masshtablashning natijalari 3-rasmda ko'rsatilgan.



3-rasm. Tasvirni masshtablash namunalari. a) berilgan tasvir;

b) tasvir kichraytirilgan ($S=0.6$); c) tasvir kattalashtirilgan ($S=1.3$).

Tasvirlarni qayta ishlashda segmentasiyalash usullari. “Segmentatsiya” so‘zining ma‘nosi “Bo‘laklash” so‘ziga mos keladi. Shu sababli, bu ikki so‘z bir ma‘noni beradi. Ilmiy adabiyotlarda har ikki termindan foydalanish hollari uchraydi.

Tasvirlarni bo‘laklash deb, ularni talqin etish mumkin bo‘lgan bo‘laklarga ajratish tushuniladi. Bo‘laklashning amaliy jihatdan muhim xususiy holi – bu yorug‘lik, geometrik va boshqa xususiyatlari tomonidan ham, mohiyati jihatidan ham turlicha bo‘lgan obyektlarni ajratib olish masalasidir. Bo‘laklashning muhim vazifalaridan biri tasvirga ishlov berishning keyingi bosqichlarida ishlatilmaydigan axborotni tashlab yuborishdir.

Masalaning bir necha matematik ifodasi mavjud, ularning umumiyrog‘i bir jinslilik predikati orqali berilgan. Agar $f(x,y)$ - bo‘laklanayotgan tasvir yorug‘lik funksiyasi; x – uning aniqlanish sohasining chekli to‘plamostisi; $S=\{S1,S2,...,Sk\}$ – x ni k ta bo‘shmas bog‘langan to‘plamostilarga ajratish; $Pn-S$ to‘plamida aniqlangan va faqatgina biror $Si: i \in [0,k]$ to‘plamostining ikki nuqtasi ma‘lum bir birjinslilik kriteriysini qanoatlantirgandagina I “rost” (TRUE) qiymatni oladigan predikat bo‘lsa, tasvirni bo‘laklash deb, uni $S^*=\{S1^*,S2^*,...,Sk^*\}$ bo‘laklarga ajratish tushuniladi.

Tasvirlarni bo‘laklashda elementlarning o‘xshash yoki farqlanishini nazariy asoslovchi birjinslilik predikati muhim ahamiyatga ega. Bo‘laklashning biror algoritmini qo‘llash jarayonida bo‘laklash sohalari va bir jinsli sohalarni qanday belgilashni oldindan hal etish lozim. Bunday belgilarning soni obyektlar yoki obyekt sinflarining soni bilan aniqlanadi.

Bo'laklash masalasini yyechishga ikki xil yondoshish mumkin. Birinchisi bir sohadan ikkinchisiga o'tishda tasvir nuqtalari xususiyatlarining «uzilish» g'oyasiga asoslangan. U bo'laklashni sohalarning chegaralarini aniqlash masalasiga keltiradi. Ikkinchisi esa, bir xil xususiyatli nuqtalarni ajratib olib, ularni birlashtirish, nom berish yoki mazmunan nishonlashga asoslangan. Birinchi yo'nalish sohalarning chegaralarini ajratib bo'laklash deyilsa, ikkinchisi soha nuqtalarini tanlab nishonlash asosida bo'laklash deyiladi.

Tasvirlarni bo'laklashda **soha o'stirish usullari** ko'proq qo'llaniladi. Bu usullar tasvirlarni bo'laklash uchun lokal belgilar haqidagi axborotdan faol foydalanishga asoslangan va g'oyasi juda sodda. Manba tasvirda bir nechta boshlang'ich nuqta olinadi va ma'lum belgilar bilan belgilanadi, so'ngra ularning atrofidagi nuqtalar tahlil qilinadi. Agar ko'rilayotgan ikki nuqta (dastlab boshlang'ich va atrofdagi biror nuqta, keyin esa oldingi qadamda aniqlangan nuqta va uning biror qo'shnisi) uchun birjinslilik sharti qanoatlansa, qo'shni nuqta bo'laklangan sohaning belgisi bilan belgilanadi. Keyin belgilangan yangi nuqta uchun yana shu jarayon qaytariladi, ya'ni atrofda u bilan birjinslilik shartini qanoatlantiruvchi yangi nuqta izlanadi. Bu jarayon toki tasvirning hamma nuqtalari belgilanmaguncha davom etadi.

Agar oldindan tasvirdagi birjinsli maydonlar soni va boshlang'ich nuqtalarning o'rni (maydon chegarasidan etarlicha uzoqda) ma'lum bo'lsa, va birjinslilik mezoni hisoblash uchun murakkab bo'lmasa, ko'rilayotgan usul yordamida etarlicha sodda algoritmlar tuzish va sifatli natijalar olish mumkin, aks xolda (amalda ko'pincha shunday) bu usullar ancha murakkab jarayonga aylanadi.

Umuman olganda, tasvirlarni segmentatsiyalar usullarini quyidagi to'rtta sinfga ajratish mumkin.

- 1) Vizual birjinsli sohalar asosida segmentatsiyalash.
- 2) Sohalar chegaralarini ajratish orqali segmentatsiyalash.
- 3) Piksel xususiyatlari asosida segmentatsiyalash.
- 4) Fizik xususiyatlar asosida segmentatsiyalash.

Masalan, fizik xususiyatlar asosida segmentatsiyalashga misol sifatida odam teri rangini ajratib olish masalasini ko'rishimiz mumkin. Ma'lumki, rangli tasvirlarda bitta nuqtadagi rang qiymati uchta rang (*R-qizil*, *G-yashil*, *B-ko'k*) aralashmalaridan tashkil topgan. Teri rangidagi sohalarni ajratish uchun aynan shu ranglar uchun quyidagi shartlar bajarilishi talab etiladi:

$$R > 95 \text{ va } G > 40 \text{ va } B > 20 \text{ va } \max(R, G, B) - \min(R, G, B) > 15 \text{ va}$$

$$|R - G| > 15 \text{ va } R > G \text{ va } R > B \text{ va } [(R * 100) / (R + G + B)] < 57 \text{ va}$$

$$[(G * 100) / (R + G + B)] < 35 \text{ va } [(B * 100) / (R + G + B)] < 35.$$

Mat SkinRegion(Mat img)

```
{  
    Mat skinMat;  
    img.copyTo(skinMat);  
    Vec3b white(255, 255, 255);  
    for (int j = 0; j < img.rows; j++) {  
        for (int i = 0; i < img.cols; i++) {  
            Vec3b pixel = img.at<Vec3b>(j, i);  
            uchar B = pixel[0];  
            uchar G = pixel[1];  
            uchar R = pixel[2];  
            int total = R + G + B;  
            if (total == 0) continue;
```

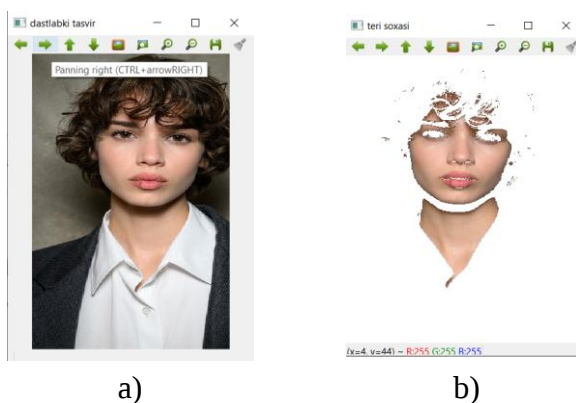
```
bool isSkin = (R > 95) && (G > 40) && (B > 20) &&
              (max(max(R,G),B) - min(min(R,G),B) > 15) &&
              (abs(R-G) > 15) && (R > G) && (R > B) &&
              ((R*100/total) < 57) &&
              ((G*100/total) < 35) &&
              ((B*100/total) < 35);
if (!isSkin) {
    skinMat.at<Vec3b>(j, i) = white;
}
}
}

return skinMat;
}

void MainWindow::on_pushButton_clicked()
{
    Mat tasvir = imread("D:/2.jpg", 1);

    if (tasvir.empty()) {
        cout << "Rasm topilmadi!" << endl;
        return;
    }
    imshow("dastlabki tasvir", tasvir);
    Mat teri_img = SkinRegion(tasvir); // Funksiyani chaqirish
    imshow("teri sohasi", teri_img);
    waitKey(0);
}
```

Teri rangini bo‘laklab olish dasturi natijasi quyida (4-rasm) ko‘rsatilgan.



4-rasm. Berilgan rangli tasvir (a) va teri rangi sohasi (b).

XULOSA

Xulosa qilib aytganda, tasvirni qayta ishlash usullari tasvirlardan kerakli ma’lumotlarni ajratib olish, ularning vizual sifatini yaxshilash yoki keyingi tahlilda foydalanish uchun xizmat qiladi. Yuqorida keltirilgan tasvirni burchak bo‘yicha burish, tasvirni masshtablash yoki kichraytirish, tasvirni bo‘laklash usullarining matematik yechimlari va OpenCV dasturiy kodlari orqali olingan natijalar ularning qanday ishlashini nazariy jihatdan tushunishga imkon beradi. Shunday qilib, tasvirni qayta ishlash usullarini nazariy matematik asos va amaliy dasturiy vositalar

bilan birgalikda o‘zlashtirish, ularni real muammolarni yuechishda to‘g‘ri va samarali qo‘llashning asosiy kalitidir.

FOYDALANILGAN ADABIYOTLAR RO‘YXATI

1. Форсайт Д. А., Понс Дж. Компьютерное зрение. Современный подход. -М.: Вильямс, 2004. -926 с.Грузман И.С., Киричук В.С. и др. Цифровая обработка изображений в информационных системах: Учебное пособие.- Новосибирск: Изд-во НГТУ, 2000. – 168 с.
2. Rafael C. Gonzalez, Richard E. Woods. Digital Image Processing, Prentice Hall. -2002. - 793 p.
3. Pratt, William K. Digital image processing : PIKS Scientific inside / William K. Pratt.—, 4th ed. – 2007. – 782 p.
4. Гонсалес Р., Вудс Р., Цифровая обработка изображений. — М.: Техносфера, 2005, 2006. — 1072 с.
5. Джордж Стокман, Линда Шапиро. Компьютерное зрение = Computer vision. — М.: Бином. Лаборатория знаний, 2006. — 752 с.
6. Красилников Н. Н. Цифровая обработка 2D- и 3D-изображений; БХВ-Петербург - Москва, 2011. - 608 с.
7. S. Xoshimov. Raqamli tasvirlarni qayta ishlash. O‘quv-uslubiy majmua – Namangan,2022