

ДОДАТОК А
Код програми

Файл MyForm.cpp:

```
#include "MyForm.h"

using namespace System;
using namespace System::Windows::Forms;

[STAThreadAttribute]

void Main()
{
    CAMShift::MyForm form;
    Application::Run(% form);
}
```

Файл MyForm.h:

```
#pragma once
#pragma comment(lib, "cv210.lib")
#pragma comment(lib, "cvaux210.lib")
#pragma comment(lib, "cxcore210.lib")
#pragma comment(lib, "highgui210.lib")

#include <iostream>
#include <highgui.h>
#include <cv.h>

CvCapture* capture;
IplImage* image = 0, * hsv = 0, * hue = 0, * mask = 0, * backproject = 0, * histimg = 0;
CvHistogram* hist = 0;
int backproject_mode = 0;
int select_object = 0;
int track_object = 0;
int show_hist = 1;
CvPoint origin;
CvRect selection;
CvRect track_window;
CvBox2D track_box;
CvConnectedComp track_comp;
int hdims = 16;
float hranges_arr[] = { 0,180 };
float* hranges = hranges_arr;
int vmin = 10, vmax = 256, smin = 30;
void OnCamera();
void on_mouse(int event, int x, int y, int flags, void* param);
CvScalar hsv2rgb(float hue);

namespace CAMShift {

    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;
    using namespace cv;

    public ref class MyForm : public System::Windows::Forms::Form
    {
    public:
```

```

MyForm(void)
{
    InitializeComponent();
}

protected:
    ~MyForm()
    {
        if (components)
        {
            delete components;
        }
    }
private: System::Windows::Forms::Button^ button1;

protected:

private:
    System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

void InitializeComponent(void)
{
    this->button1 = (gcnew System::Windows::Forms::Button());
    this->SuspendLayout();
    //
    // button1
    //
    this->button1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 12, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
    this->button1->Location = System::Drawing::Point(12, 124);
    this->button1->Name = L"button1";
    this->button1->Size = System::Drawing::Size(97, 31);
    this->button1->TabIndex = 0;
    this->button1->Text = L"Start";
    this->button1->UseVisualStyleBackColor = true;
    this->button1->Click += gcnew System::EventHandler(this,
&MyForm::button1_Click);
    //
    // MyForm
    //
    this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
    this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
    this->ClientSize = System::Drawing::Size(359, 170);
    this->Controls->Add(this->button1);
    this->FormBorderStyle =
System::Windows::Forms::FormBorderStyle::FixedToolWindow;
    this->Name = L"MyForm";
    this->Text = L"Program";
    this->ResumeLayout(false);

}
#pragma endregion
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^
e)
{
    if (capture = cvCaptureFromCAM(0))
    {

```

```

System::String^ s;
s = System::String::Format("Hot keys: \n"
    "\tESC - quit the program\n"
    "\tc - stop the tracking\n"
    "\tb - switch to/from backprojection view\n"
    "\th - show/hide object histogram\n"
    "To initialize tracking, select the object with mouse\n");
Graphics^ g = MyForm::CreateGraphics();
g->DrawString(s, gnew Drawing::Font("Arial", 10), Brushes::Black,
20, 20);

    OnCamera();
}
else MessageBox::Show("Camera reading Error");
}
};
}

void OnCamera()
{
    cvNamedWindow("Histogram", 1);
    cvNamedWindow("CAMShift", 1);
    cvSetMouseCallback("CAMShift", on_mouse, 0);
    cvCreateTrackbar("Vmin", "CAMShift", &vmin, 256, 0);
    cvCreateTrackbar("Vmax", "CAMShift", &vmax, 256, 0);
    cvCreateTrackbar("Smin", "CAMShift", &smin, 256, 0);

    for (;;)
    {
        IplImage* frame = 0;
        int i, bin_w, c;

        frame = cvQueryFrame(capture);
        if (!frame)
            break;

        if (!image)
        {
            /* allocate all the buffers */
            image = cvCreateImage(cvGetSize(frame), 8, 3);
            image->origin = frame->origin;
            hsv = cvCreateImage(cvGetSize(frame), 8, 3);
            hue = cvCreateImage(cvGetSize(frame), 8, 1);
            mask = cvCreateImage(cvGetSize(frame), 8, 1);
            backproject = cvCreateImage(cvGetSize(frame), 8, 1);
            hist = cvCreateHist(1, &hdims, CV_HIST_ARRAY, &hranges, 1);
            histimg = cvCreateImage(cvSize(320, 200), 8, 3);
            cvZero(histimg);
        }

        cvCopy(frame, image, 0);
        cvCvtColor(image, hsv, CV_BGR2HSV);

        if (track_object)
        {
            int _vmin = vmin, _vmax = vmax;

            cvInRangeS(hsv, cvScalar(0, smin, MIN(_vmin, _vmax), 0),
                cvScalar(180, 256, MAX(_vmin, _vmax), 0), mask);
            cvSplit(hsv, hue, 0, 0, 0);

            if (track_object < 0)
            {
                float max_val = 0.f;

```

```

cvSetImageROI(hue, selection);
cvSetImageROI(mask, selection);
cvCalcHist(&hue, hist, 0, mask);
cvGetMinMaxHistValue(hist, 0, &max_val, 0, 0);
cvConvertScale(hist->bins, hist->bins, max_val ? 255. /
max_val : 0., 0);

cvResetImageROI(hue);
cvResetImageROI(mask);
track_window = selection;
track_object = 1;
cvZero(histimg);
bin_w = histimg->width / hdims;
for (i = 0; i < hdims; i++)
{
    int val = cvRound(cvGetReal1D(hist->bins, i) *
histimg->height / 255);

    CvScalar color = hsv2rgb(i * 180.f / hdims);
    cvRectangle(histimg, cvPoint(i * bin_w, histimg->
>height),
                cvPoint((i + 1) * bin_w, histimg->height -
val),
                color, -1, 8, 0);
}
}
cvCalcBackProject(&hue, backproject, hist);
cvAnd(backproject, mask, backproject, 0);
cvCamShift(backproject, track_window,
            cvTermCriteria(CV_TERMCRIT_EPS | CV_TERMCRIT_ITER, 10, 1),
            &track_comp, &track_box);
track_window = track_comp.rect;
if (backproject_mode)
    cvCvtColor(backproject, image, CV_GRAY2BGR);
if (!image->origin)
    track_box.angle = -track_box.angle;
cvEllipseBox(image, track_box, CV_RGB(255, 0, 0), 3, CV_AA, 0);
}

if (select_object && selection.width > 0 && selection.height > 0)
{
    cvSetImageROI(image, selection);
    cvXorS(image, cvScalarAll(255), image, 0);
    cvResetImageROI(image);
}
cvShowImage("CAMShift", image);
cvShowImage("Histogram", histimg);
c = cvWaitKey(10);
if ((char)c == 27)
    break;
switch ((char)c)
{
case 'b':
    backproject_mode ^= 1;
    break;
case 'c':
    track_object = 0;
    cvZero(histimg);
    break;
case 'h':
    show_hist ^= 1;
    if (!show_hist)
        cvDestroyWindow("Histogram");
    else
        cvNamedWindow("Histogram", 1);
    break;
}

```


ДОДАТОК Б

Висвітлення результатів у статті

Міністерство освіти і науки України



NURE

Харківський національний університет
радіоелектроніки

ЗБІРНИК

студентських наукових статей

«Автоматизація та приладобудування»

«Automation and Development of Electronic Devices»

ADED-2025

(Випуск 2)

[електронне видання]



<http://nure.ua/department/kafedra-komp-yuterno-integrovanih-tehnologiy-avtomatizatsiyi-ta-mehatroniki-kitam>



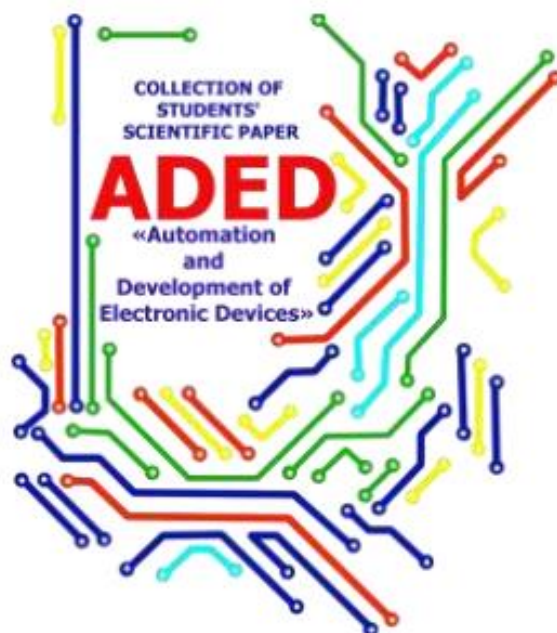
<http://itez.zntu.edu.ua/>



<http://kafea.kdu.edu.ua>

Харків 2025

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки
кафедра комп'ютерно-інтегрованих технологій, автоматизації та робототехніки
(КІТАР)



ЗБІРНИК

студентських наукових статей

«Автоматизація та приладобудування»

«Automation and Development of Electronic Devices»

ADED-2025

(Випуск 2)

[електронне видання]

Харків 2025

ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ КЕРУВАННЯ КВАДРОКОПТЕРАМИ: АНАЛІЗ ФУНКЦІОНАЛЬНИХ АСПЕКТІВ ТА ПЕРСПЕКТИВИ РОЗВИТКУ

Мамін В.А.

Харківський національний університет радіоелектроніки

Україна, 61166, Харків, пр.Науки 14

E-mail: vladyslav.mamin@nure.ua

Анотація: У статті розглядається основні функціональні аспекти інтелектуальних систем керування квадрокоптерами. Проведений аналіз функціональних аспектів та перспектив інтелектуальних систем у напрямку адаптивного керування.

Ключові слова: інтелектуальні системи, штучні нейронні мережі, квадрокоптери.

INTELLIGENT QUADCOPTER CONTROL SYSTEMS: ANALYSIS OF FUNCTIONAL ASPECTS AND DEVELOPMENT PROSPECTS

Mamin V.A.

Kharkiv National University of Radio Electronics

Ukraine, 61166, Kharkiv, Nauky Ave. 14

E-mail: vladyslav.mamin@nure.ua

Abstract: This article examines the main functional aspects of intelligent quadcopter control systems. An analysis of the functional aspects and prospects of intelligent systems in the direction of adaptive control.

Keywords: intelligent systems, artificial neural network, quadcopters.

З розвитком технологій та підвищенню обчислювальної спроможності у мікропроцесорів постала потреба в обробці великих масивів інформації, використання баз знань для генерації направленої діяльності - це призвело до створення інтелектуальних систем.

Під інтелектуальною системою розуміють об'єднану інформаційним процесом структуру технічних засобів і програмного забезпечення, котрі працюють у взаємозв'язку з оператором або незалежно від нього; здатну на основі відомостей і знань при наявності мотивації синтезувати, генерувати рішення про дію та знаходити ефективні шляхи вирішення завдань[1].

Виділяється декілька засобів для створення інтелектуальних систем керування:

- експертні системи;
- штучні нейронні мережі (artificial neural networks);
- нечітка логіка (fuzzy logic);
- еволюційні методи і генетичні алгоритми (genetic algorithms).

Використання штучних нейронних мереж дозволяє розв'язати задачі керування нелінійними ОК шляхом створення адаптивних САК з навчаємим нейрорегулятором. При адаптації навчання використовується для отримання інформації про стан і характеристики САК, що необхідні для оптимального керування в умовах невизначеності. Тобто адаптація тотожна оптимізації в умовах недостатньої інформації.

Квадрокоптери є універсальними пристроями спостереження або виконувачами примітивних задач. Завдяки стабільності у повітрі та можливості переміщення у трьох основних напрямках вони стають ідеальними пристроями для цих видів робіт.

Однією з основних функцій квадрокоптерів є можливість здійснювати аеріальні зйомки. Вони оснащені вбудованими камерами або кріпленнями для камер, які дозволяють знімати відео та фотографії з висоти.

Для безпілотних систем дуже поширена концепція використання штучного інтелекту в різних областях, у тому числі й в управлінні, де забезпечується адаптивність, самонавчання та здатність до прийняття рішень у реальному часі. Завдяки цьому такі системи можуть ефективно реагувати на зміни навколишнього середовища, компенсувати похибки сенсорів і підтримувати стабільність польоту навіть у складних умовах.

З боку функціональних аспектів відмічаються як приклад декілька напрямків використання:

- розпізнавання середовища;
- стабілізація польоту;
- навігація в середовищі;
- системи комунікації.

Інтелектуальні системи можуть застосовуватися для розпізнавання та навігації в навколишньому середовищі за допомогою технологій комп'ютерного зору, LIDAR, GPS та інерціальних вимірювальних одиниць (IMU). Такий підхід забезпечує здатність безпілотного літального апарата аналізувати рельєф місцевості, будувати оптимальні траєкторії руху, а також визначати й відстежувати задані об'єкти або області спостереження.

Все частіше використовують інтелектуальні системи керування для стабілізації польоту квадрокоптера разом з класичними як PID або LQR контроллерами. Це дозволяє квадрокоптеру аналізувати динаміку польоту та навчатися й адаптуватись до зовнішніх умов як вітер, зміна ваги.

Для забезпечення стабільного та якісного каналу зв'язку інтелектуальні системи здатні самостійно аналізувати рівень сигналу, затримку передачі та автоматично коригувати робочі параметри, зокрема частоту, протокол або потужність передавача.

Дані аспекти(рис.1) можуть використовуватись в проектуванні інтелектуальної системи контролю, навігації та комунікації в процесі польоту.

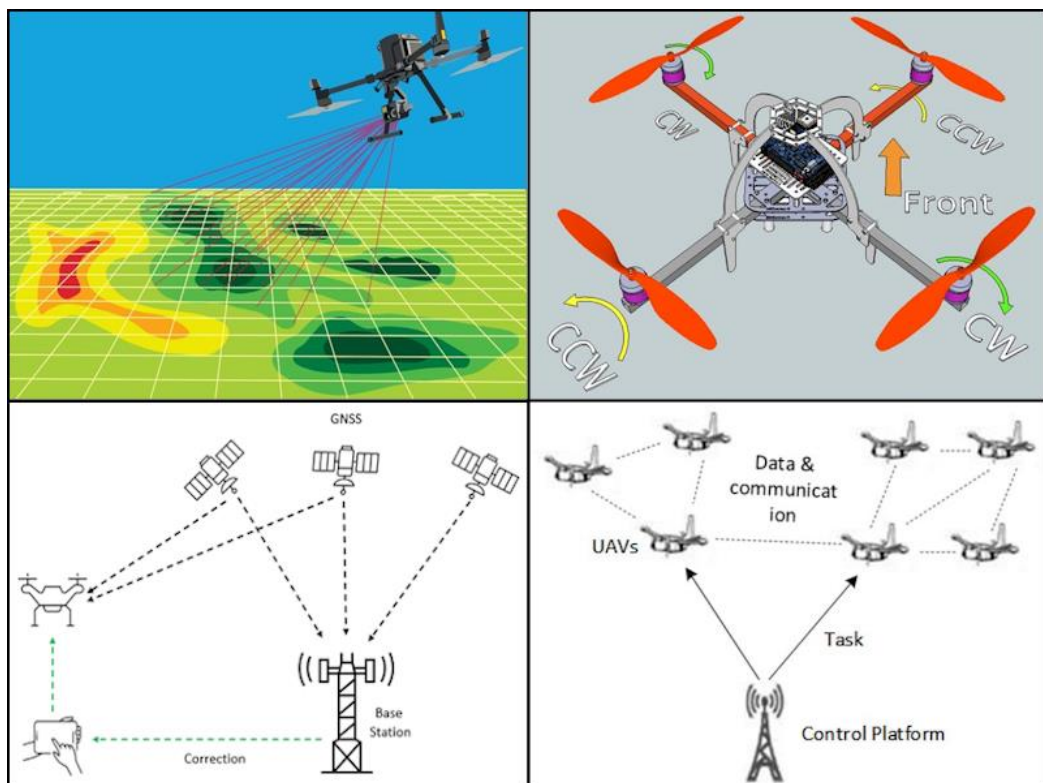


Рисунок 1 – Види функціональних аспектів інтелектуальних системах керування квадрокоптера

Поєднання інтелектуальних систем розпізнавання середовища, навігації, стабілізації та комунікації створить умови, за яких безпілотник зможе самостійно орієнтуватися, ухвалювати рішення й взаємодіяти з іншими апаратами у спільному повітряному просторі.

В перспективі всі ці аспекти можуть призвести політ квадрокоптера або дрону до повної автономності завдяки використанню удосконаленими технологіями штучного інтелекту, що поєднують алгоритми глибокого навчання, обробку сенсорних даних і адаптивне керування.

Сучасні тенденції розвитку інтелектуальних систем керування квадрокоптерами демонструють рух у напрямку створення повністю автономних, самонавчальних і взаємопов'язаних безпілотних платформ. Застосування таких систем дає змогу реалізувати ефективну координації між кількома апаратами, підвищити точність виконання завдань та забезпечити автономну роботу у складних і непередбачуваних умовах. У перспективі це сприятиме розширенню сфер застосування квадрокоптерів.

ЛІТЕРАТУРА

1. ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ АВТОМАТИЗАЦІЇ : моногр. / О.Г. Аврунін, С.І. Владов, М.В. Петченко та ін. – Кременчук : Видавництво «НОВАБУК», 2021. – 322 с. – ISBN 978-617-639-347-4.
2. Інтелектуальні системи керування : конспект лекцій для студентів спеціальності «Прилади та системи керування літальними апаратами». / В. О. Апостолюк, О. С. Апостолюк. – К.: НТУУ «КПІ», 2008. – 88 с. – Бібліогр.: с. 84 –85. – 50 пр.
3. Теорія систем керування: підручник / В.І. Корнієнко, О.Ю. Гусєв, О.В. Герасіна, В.П. Щокін; М-во освіти і науки України, Нац. гірн. ун-т. – Дніпро: НГУ, 2017. – 497 с. ISBN 978-966-350-650-0
4. P.J. Antsaklis, "Intelligent Control," Encyclopedia of Electrical and Electronics Engineering , Vol. 10, pp. 493-503, John Wiley & Sons, Inc., 1999.
5. Celal Onur GÖKÇE, "Intelligent Quadcopter Control Using Artificial Neural Networks" Afyon Kocatepe University Journal of Science and Engineering, AKU J. Sci. Eng. 23 (2023) 015202 (138-142) DOI: 10.35414/akufemubid.1229424
6. Chebanchyk D. Analysis of Object Identification Methods for FPV Drones / D. Chebanchyk, V.Yevsieiv // Manufacturing & Mechatronic Systems 2025 : Theses of Reports of IX-st International Conference, October 25-26, 2025. - Kharkiv, 2025. - P. 30-33.
7. Hybrid Approaches to Building Intelligent Robotic Systems on FPGAs and MCUs for Industry 5.0 Tasks / V. Yevsieiev, S. Maksymova, N. Demska, N. Starodubcev // Theoretical and Applied Aspects of Device Development on Microcontrollers and FPGAs (MC&FPGA-2025) : VII International Scientific and Practical Conference, June 27-28, 2025. – Kharkiv : NURE – P.30-35.
8. Nevliudov, I., Yevsieiev, V., Maksymova, S., & Chala, O. (2025). A small-sized robot prototype development using 3D printing. acta mechanica et automatica, 19(1).
9. Yevsieiev, V., Abu-Jassar, A., & Maksymova, S. (2024). Calculation of the Distance to Objects in Collaborative Robots Workspace Using Computer Vision. Journal of universal science research, 2(11), 240-255.
10. Gurin, D., Yevsieiev, V., Maksymova, S., & Alkhalailah, A. (2024). Using Convolutional Neural Networks to Analyze and Detect Key Points of Objects in Image. Multidisciplinary Journal of Science and Technology, 4(9), 5-15.

Науковий керівник: Цимбал Олександр Михайлович, доктор технічних наук, професор кафедри КІТАР, Харківського національного університету радіоелектроніки

ДОДАТОК В

Демонстраційний матеріал у вигляді презентації

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Розроблення автоматичної системи трекінгу для комп'ютерного зору квадрокоптера

Виконав:
Ст. гр. КТРСм-24-2
Мамін В.А

Керівник:
Проф. каф. КІТАР
Цимбал О.М.

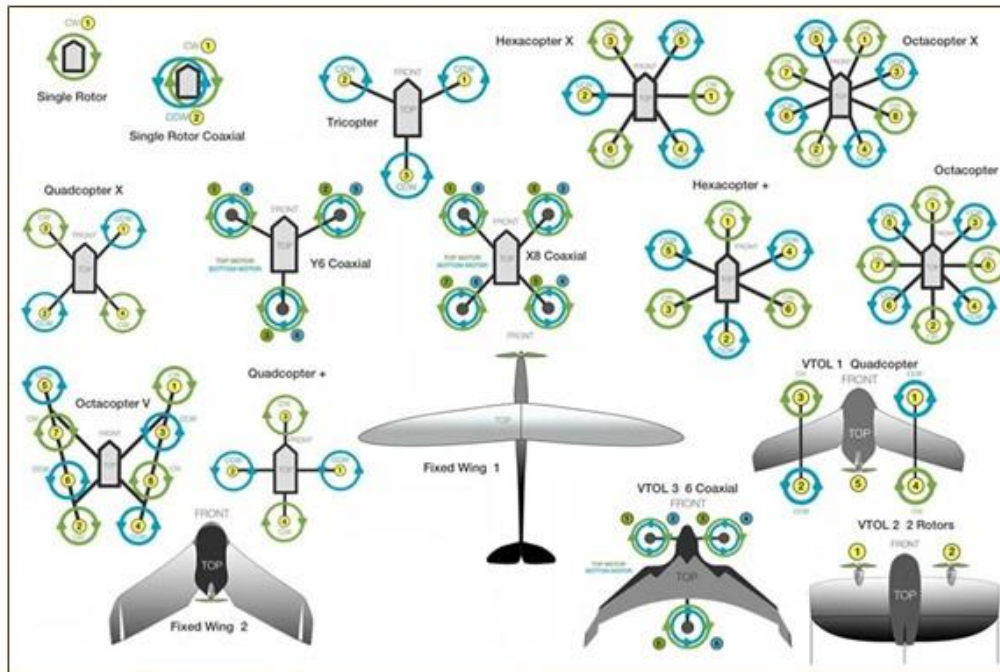
Харків 2025

МЕТА РОБОТИ ТА ПОСТАНОВКА ЗАДАЧІ

Мета кваліфікаційної роботи – розробка програмного забезпечення для трекінгу об'єктів з використанням комп'ютерного зору квадрокоптера.

Задачі кваліфікаційної роботи:

- провести аналіз типів квадрокоптерів, їх технічні засоби та їх використання;
- провести аналіз методів обробки зображень, алгоритмів для використання їх у комп'ютерному зорі квадрокоптера;
- розробити програмне рішення для виконання трекінгу об'єкта використовуючи комп'ютерний зір квадрокоптера;
- провести експериментальну частину розробки та зробити виновки роботи програми.



Квадрокоптери в більшості класифікуються за різними критеріями, кількістю та розташування гвинтів або крил, мети використання, дальності польоту, масі корисного навантаження, базового механізму.

КЛАСИФІКАЦІЯ БПЛА

За типом конструкції:

- трикоптери оснащений 3 гвинтами;
- квадрокоптери із 4 гвинтами;
- гексакоптери із 6 гвинтами;
- октокоптери із 8 гвинтами.

За технічними характеристиками:

- вертольотні/коптерні;
- літакового (Fixed Wing);
- гібридні (VTOL).



СФЕРА ВИКОРИСТАННЯ КВАДРОКОПТЕРІВ

З поширенням квадрокоптерів у повсякденному житті вони отримали безліч сфер застосування за рахунок своєї простої конструкції яку можна легко змінити або переробити під інший напрям роботи, якщо привести як приклад сфер використання то можна виділити такі як аерозйомка, забезпечення безпеки, екстрені служби, логістичні.

Сфери використання:

- аерозйомка (професійна, аматорська);
- екстрені служби (використання квадрокоптерів для пошуку);
- охороний (патрулювання територій);
- логістичний (використання для перевезення пакунків).

ТЕХНІЧНІ ЗАСОБИ КОМП'ЮТЕРНОГО ЗОРУ

Камера в квадрокоптері є одним з ключових компонентів, залежно від якого віддається перевага тій чи іншій моделі. У її функції входить не тільки фото та відеозйомка, а й маневреність агрегату в польоті. Існує кілька різновидів камер для дронів.



Аналогова камера
KaraFPV SPY 1200TVL (SPY);



Цифрова камера
Foxeer T Rex Micro 1500TVL (HS1252);




Екшн камера
A7 Sport Full HD 1080P.

АНАЛІЗ МЕТОДІВ ОБРОБКИ ЗОБРАЖЕНЬ

Перед тим, як методи комп'ютерного зору можуть бути застосовані до відеоданих з тим, щоб витягти певну частку інформації, необхідно обробити відеодані, з тим щоб вони задовольняли деяким умовам, залежно від використовуваного методу.

Етапи роботи комп'ютерного зору з використанням методів обробки зображень:

- фільтрація шумів, нормалізація, підвищення контрасту;
 - конвертація колірної моделі зображення;
 - виділення контурів та ліній;
 - виділення ознак.
- 



ІТЕРАТИВНИЙ АЛГОРИТМ CAMSHIFT (CONTINUOUSLY ADAPTIVE MEAN SHIFT)

CAMShift являє собою ітеративний алгоритм кластеризації, який сегментує зображення та відстежує кластери даних шляхом пошуку областей з найвищою щільністю точок. Основний принцип його роботи полягає в постійному русі кожної точки у напрямку максимальної щільності, що дозволяє алгоритму визначати кількість точок та форму кластерів.

Основні етапи роботи алгоритму:

- вибір регіону інтересу (трекінгу);
- розрахунок колірної гистограми;
- розрахунок зворотної проєкції;
- пошук максимуму розподілу ймовірностей;
- обчислення розмір об'єкта для адаптації форми.

ІТЕРАТИВНИЙ АЛГОРИТМ CAMSHIFT (Continuously Adaptive Mean Shift)

Візуалізація алгоритму Mean Shift;



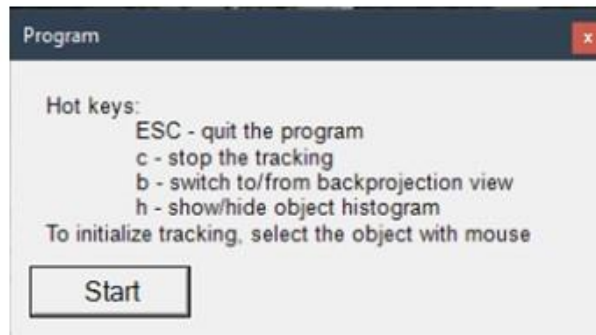
Візуалізація алгоритму CAMShift.



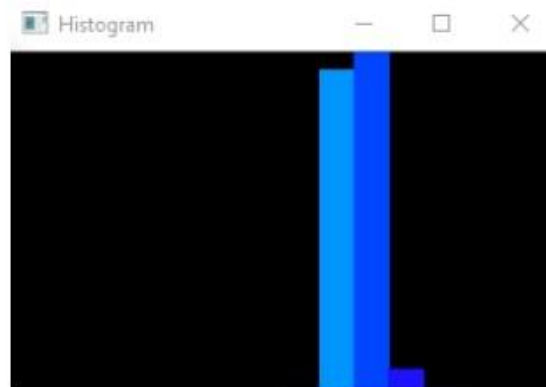
Mean shift window
initialization

ПРОГРАМНА РЕАЛІЗАЦІЯ ТРЕКІНГУ ОБ'ЄКТІВ

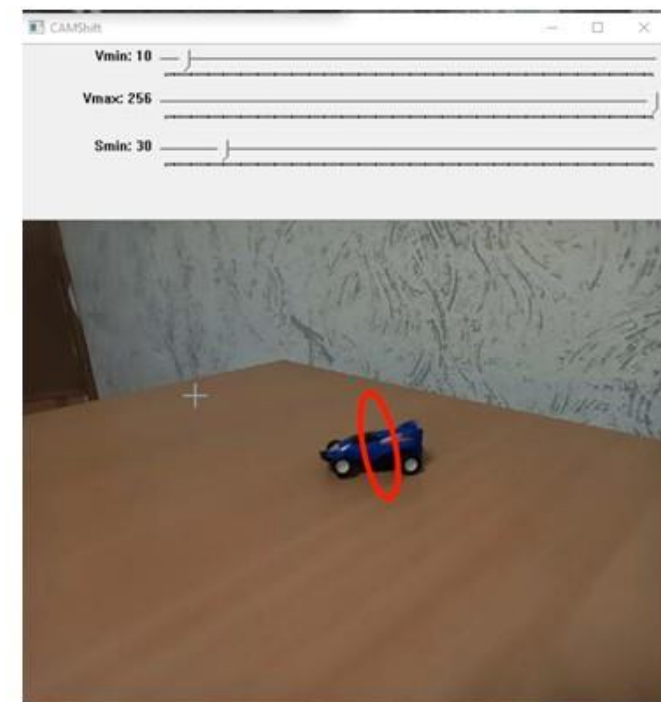
Програмна реалізація була зроблена з використання як основи платформи Windows Forms для більш зручного виводу зображень камери та налаштувань параметрів програми.



Основне вікно програми;



Вікно колірної гістограми;



Вікно виведення камери.

ПРОГРАМНА РЕАЛІЗАЦІЯ ТРЕКІНГУ ОБ'ЄКТІВ

Програмна реалізація створення гістограми;

```

if (track_object < 0)
{
    float max_val = 0.f;
    cvSetImageROI(hue, selection);
    cvSetImageROI(mask, selection);
    cvCalcHist(&hue, hist, 0, mask);
    cvGetMinMaxHistValue(hist, 0, &max_val, 0, 0);
    cvConvertScale(hist->bins, hist->bins, max_val ? 255. / max_val : 0., 0);
    cvResetImageROI(hue);
    cvResetImageROI(mask);
    track_window = selection;
    track_object = 1;

    cvZero(histimg);
    bin_w = histimg->width / hdims;
    for (i = 0; i < hdims; i++)
    {
        int val = cvRound(cvGetReal1D(hist->bins, i) * histimg->height / 255);
        CvScalar color = hsv2rgb(i * 180.f / hdims);
        cvRectangle(histimg, cvPoint(i * bin_w, histimg->height),
            cvPoint((i + 1) * bin_w, histimg->height - val),
            color, -1, 8, 0);
    }
}

```

Програмна реалізація трекінгу вибраного об'єкту;

```

cvCalcBackProject(&hue, backproject, hist);
cvAnd(backproject, mask, backproject, 0);
cvCamShift(backproject, track_window,
    cvTermCriteria(CV_TERMCRIT_EPS | CV_TERMCRIT_ITER, 10, 1),
    &track_comp, &track_box);
track_window = track_comp.rect;

if (backproject_mode)
    cvCvtColor(backproject, image, CV_GRAY2BGR);
if (!image->origin)
    track_box.angle = -track_box.angle;
cvEllipseBox(image, track_box, CV_RGB(255, 0, 0), 3, CV_AA, 0);
}

if (select_object && selection.width > 0 && selection.height > 0)
{
    cvSetImageROI(image, selection);
    cvXorS(image, cvScalarAll(255), image, 0);
    cvResetImageROI(image);
}

cvShowImage("CAMShift", image);
cvShowImage("Histogram", histimg);

```

ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА РОБОТИ

Розробивши програму з використанням ітеративного алгоритму CAMShift були проведені експерименти на якість трекінгу в різних умовах освітлення, складності об'єкта та наявності загороджень та схожих с об'єктом по формі чи кольору. В експерименті використовувались об'єкти для імітації умов, перешкоди, різні умови освітлення кімнати.

Загалом було зроблено чотири експерименти:

- трекінг контрастного об'єкта;
- трекінг об'єкта в умовах обмеженого освітлення;
- трекінг об'єкта в умовах блокування іншим об'єктом;
- трекінг об'єкта в умовах неповного блокування схожим за кольором об'єктом.

ВИСНОВКИ

За кваліфікаційною роботою був проведений:

- аналіз типів квадрокоптерів, їх технічні засоби та їх використання;
- аналіз методів обробки зображень, алгоритмів для використання їх у комп'ютерному зорі квадрокоптера;
- розроблення програмного рішення для виконання трекінгу об'єкта використовуючи комп'ютерний зір квадрокоптера;
- експериментальна частина розробки.

На основі отриманих даних експерименту програма з використанням ітеративний алгоритму CAMShift забезпечує стабільний трекінг об'єкта в умовах обмеженого освітлення та наявності шумових перешкод. Були виявлені обмеження в роботі програмного рішення у випадках часткового та повного перекриття іншими об'єктами.

