

# Aproximarea poligonală a contururilor obiectelor

Oprisiu Ionut Daniel

Grupa 30233

Rationale and task management:

## 1. Obiectivul Proiectului și Overview

Acest proiect vizează dezvoltarea unei soluții pentru aproximarea poligonală a contururilor obiectelor din imagini color (BMP, 24 biți/pixel). Metoda se desfășoară în două etape principale:

1. **Extragerea contururilor:** Se obțin contururile obiectelor prin aplicarea tehnicilor de preprocesare și edge detection (folosind, de exemplu, algoritmul Canny și funcțiile de detectare a contururilor din OpenCV).
2. **Aproximarea poligonală:** Se simplifică contururile extrase prin utilizarea algoritmului Ramer-Douglas-Peucker (prin funcția `approxPolyDP` din OpenCV), reducând numărul de puncte care descriu forma, păstrând totodată integritatea geometriei obiectului.

Această soluție reduce complexitatea datelor vizuale, facilitând aplicații practice precum vectorizarea imaginilor și recunoașterea obiectelor.

## 2. Context și Motivație

În cadrul materiei de procesare a imaginilor, proiectul abordează:

- **Vectorizarea grafică:** Transformarea imaginilor raster în reprezentări vectoriale, utile pentru editare, analiză și stocare eficientă.
- **Recunoașterea obiectelor:** Simplificarea contururilor pentru extragerea caracteristicilor esențiale, facilitând clasificarea și analiza formelor.
- **Aplicații industriale și de cercetare:** Reducerea complexității datelor geometrice contribuie la îmbunătățirea algoritmilor de analiză vizuală și la eficientizarea proceselor de prelucrare a imaginilor.

Implementarea acestui proiect va integra cunoștințele dobândite în laborator privind preprocesarea, extragerea și analiza contururilor.

## 3. Planul de Implementare (Project Plan)

Proiectul va fi structurat în următoarele etape:

### Etapa 1: Analiza și Preprocesarea Imaginilor

- **Cercetare Preliminară:** Studierea tehnicilor de preprocesare, extragere a contururilor și a algoritmului Ramer-Douglas-Peucker.
- **Colectarea Datelor:** Selectarea unor imagini color BMP (24 biți/pixel) reprezentative pentru testare.
- **Preprocesare:**
  - Convertirea imaginilor la grayscale (dacă este necesar) pentru a simplifica analiza.
  - Aplicarea unui filtru Gaussian pentru reducerea zgomotului, facilitând astfel extragerea contururilor clare.

## **Etapă 2: Extragerea și Aproximarea Contururilor**

- **Extragerea Contururilor:**
  - Aplicarea unui algoritm de edge detection (de exemplu, Canny) pentru evidențierea tranzițiilor semnificative în intensitatea pixelilor.
  - Utilizarea funcției findContours din OpenCV pentru a obține contururile obiectelor.
- **Aproximarea Poligonală:**
  - Aplicarea algoritmului Ramer-Douglas-Peucker prin funcția approxPolyDP pentru a reduce numărul de puncte ale fiecărui contur.
  - Experimentarea cu parametrul de toleranță pentru a obține un echilibru optim între precizie și simplificare.

## **Etapă 3: Evaluare și Feedback**

- **Testare Inițială:** Evaluarea rezultatelor prin compararea contururilor aproximative cu cele originale și analiza performanței în aplicații precum recunoașterea obiectelor.
- **Consultare:** Discuții pentru validarea metodei și identificarea posibilelor îmbunătățiri.

## **4. Resurse și Tehnologii Utilizate**

- **C++:** Limbajul de implementare, integrând conceptele de procesare a imaginilor.
- **OpenCV:** Pentru preprocesarea imaginilor, extragerea contururilor și aplicarea algoritmului de aproximare (prin funcțiile findContours și approxPolyDP).
- **CMake:** Pentru gestionarea build-ului proiectului și facilitarea compilării.
- **Gestionarea Codului:** Utilizarea Git și GitHub pentru versionare și documentarea progresului.

## **5. Provocări și Soluții**

- **Alegerea Parametrilor Optimi:** Stabilirea valorilor ideale pentru filtrul Gaussian și toleranța algoritmului Ramer-Douglas-Peucker printr-o abordare iterativă, folosind diverse seturi de imagini.
- **Optimizarea Performanței:** Asigurarea unui timp de execuție rezonabil prin optimizarea buclelor și a funcțiilor de procesare.
- **Calibrarea Rezultatelor:** Evaluarea subiectivă și obiectivă a aproximărilor pentru a se asigura păstrarea formei obiectelor în mod fidel.

## 6. Timeline și Etapele de Dezvoltare

- **Săptămânile 2–4 (Phase 1):** Definirea obiectivelor, redactarea documentului de proiect, cercetare și planificare.
- **Săptămânile 5–10 (Phase 2):** Implementarea preprocesării, extragerea contururilor și aplicarea algoritmului de aproximare.
- **Săptămânile 11–12 (Phase 3):** Testarea pe diverse scenarii și documentarea rezultatelor.
- **Săptămânile 13–14 (Phase 4):** Îmbunătățiri și optimizări bazate pe feedback-ul obținut în timpul testărilor.

## 7. Concluzie

Proiectul va demonstra integrarea teoriei și practicii în procesarea imaginilor, prin dezvoltarea unei soluții care extrage și simplifică contururile obiectelor. Documentarea detaliată și abordarea iterativă vor asigura o soluție robustă și eficientă, utilă în aplicații practice precum vectorizarea imaginilor și recunoașterea obiectelor.

### Implementation:

Describe the architecture of your project and explain how each module is integrated into the final solution. For each module, detail its functionalities and the design choices you made. Discuss what you tried before arriving at your final version and explain why those alternatives were not the right decisions.

### Testing:

Identify some test cases to evaluate your solution. Explain the rationale behind choosing these tests and, for each case, provide qualitative and/or quantitative analysis of the results. Discuss why the tests are successful or why they fail.

### Improvements:

Select two test cases that initially failed and explain how you resolved the issues. Describe the challenging process and any modifications or additions you made to the previous solution. Clearly justify your design decisions throughout this process.