

# White Paper Alignment Based on Line and Corner Detection

Daria Dohan

## 1 Descriere generală

Acest proiect are ca scop detectarea automată a unei foi albe dintr-o imagine, identificarea colțurilor acesteia și aplicarea unei transformări perspective pentru a o alinia în mod corect, astfel încât să apară dreptunghiulară într-o imagine nouă. Programul este o implementare personalizată a pașilor OpenCV, folosind funcții scrise manual pentru conversie în tonuri de gri, binarizare, morfologie și procesare geometrică.

## 2 Scopul proiectului

- Detectarea unei foi albe (chiar și cu text scris) dintr-o imagine.
- Identificarea precisă a celor 4 colțuri ale foi.
- Transformarea perspectivei pentru a obține o imagine aliniată a foi.

## 3 Arhitectura soluției

### 3.1 Pașii principali

1. **Conversie grayscale manuală:**

$$Gray = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \quad (1)$$

2. **Threshold adaptiv manual:** Se calculează media locală într-o fereastră de dimensiune `blockSize`. Pixelul este setat ca alb (255) dacă este mai întunecat decât `mean - C`, altfel negru (0).
3. **Morfologie (închidere):** Se aplică dilatare urmată de eroziune pentru a umple golurile și a unifica marginile.
4. **Detecția colțurilor:** Se folosesc `findContours` și `approxPolyDP` pentru a obține contururi și a selecta cel mai mare poligon convex cu 4 puncte.

5. **Sortarea colțurilor:** Colțurile sunt sortate în ordinea stânga-sus, dreapta-sus, dreapta-jos, stânga-jos.
6. **Transformare perspectivă și afișare:** Se aplică transformarea folosind `getPerspectiveTransform` și `warpPerspective`.

## 4 Structura codului

### 4.1 Funcții implementate

- `customGrayscale`: convertire imagine RGB la grayscale.
- `customThreshold`: binarizare adaptivă bazată pe medie locală.
- `customDilate` & `customErode`: operații morfologice scrise manual.
- `customMorphClose`: morfologie de tip închidere (dilatare + eroziune).
- `findPaperCorners`: detectarea conturului foi.
- `sortCorners`: sortare a colțurilor pentru perspectivă corectă.
- `alignPaper`: aplicarea transformării perspective și afișare.
- `main`: funcția principală care încarcă imaginea, detectează colțurile și apelează transformarea.

## 5 Exemple de rulare

- **Imagine inițială:** Imaginea este încărcată de pe disc. Dacă imaginea nu este validă, se afișează un mesaj de eroare.
- **Colțuri detectate:** Se trasează linii verzi între colțuri și se desenează cercuri roșii în pozițiile acestora. Se afișează fereastra „Colțuri detectate”.
- **Foaie aliniată:** Se aplică transformarea perspectivei. Se afișează fereastra „Foaie Aliniată”.

## 6 Rezultate și observații

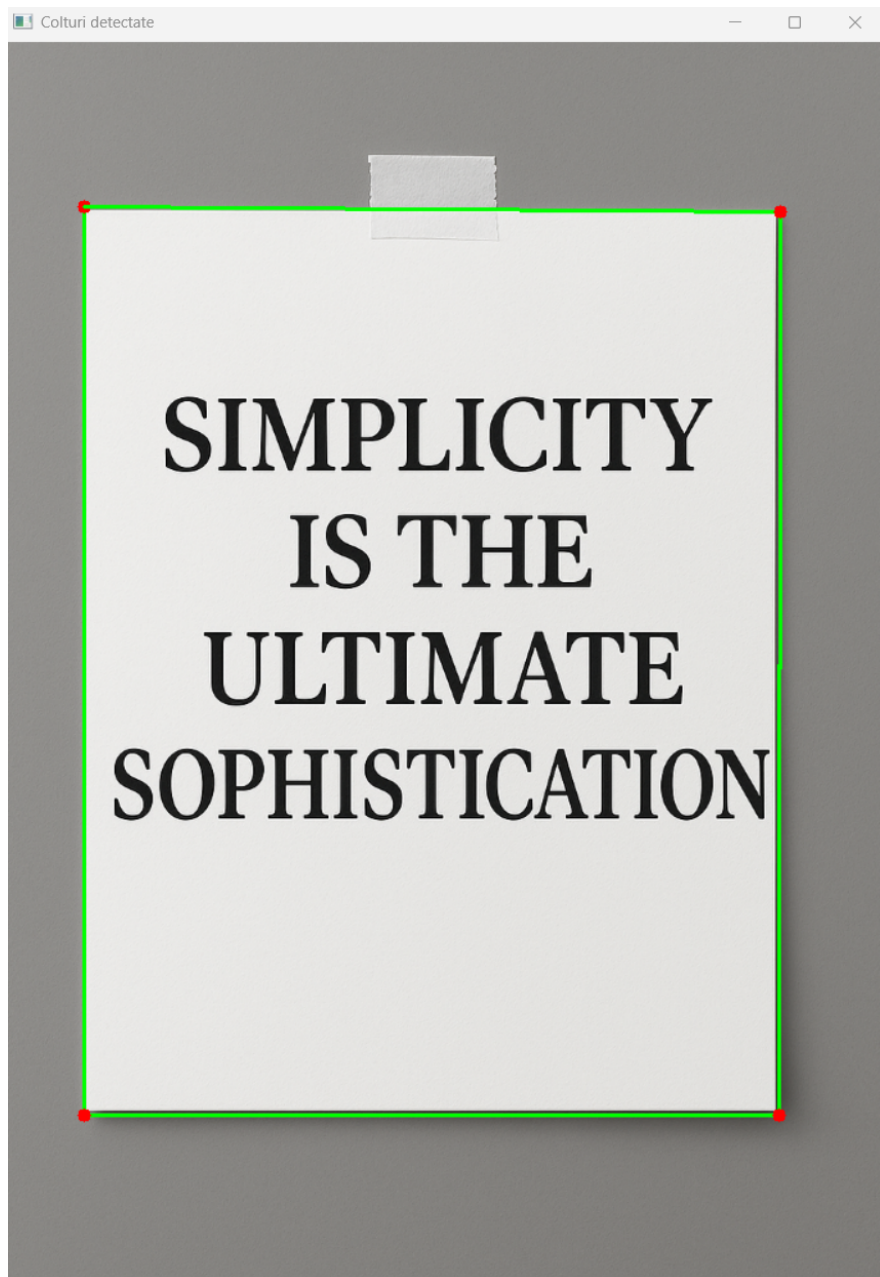
- Codul oferă rezultate similare cu o implementare clasică OpenCV.
- Este robust în cazul imaginilor cu contrast bun între hârtie și fundal.
- Pot apărea erori în detectare dacă fundalul este neuniform sau foaia este parțial ascunsă.

## 7 Bibliografie / Surse

- Curs CSE576 – University of Washington:  
*White Paper Alignment using Line and Corner Detection*  
<https://courses.cs.washington.edu/courses/cse576/05sp/papers/MSR-TR-2004.pdf>
- Documentație OpenCV: <https://docs.opencv.org/>

## 8 Exemple vizuale

- Imagine originală cu colțuri detectate:



- Imagine aliniată după transformare:



- Exemplu 1:



- Exemplan 2:

