

4. Detectia punctelor de interes de tip colt

Scop: Detectia punctelor de interes de tip colt pe baza implementarii din OpenCV a metodei Harris de detectie.

4.1. Metoda Harris de detectie a colturilor

Un punct de interes (interest point) este un punct care are o pozitie bine definita in imagine si poate fi detectat in mod robust. El poate fi un minim sau maxim local de intensitate, o terminatie a unei linii, un punct de pe un contur in care curbura are un maxim local etc. Pentru ca un punct de interes sa fie colt sunt necesare analize si validari suplimentare

Un colt se defineste ca fiind un punct in care se intersecteaza cel putin 2 muchii cu orientari diferite respectiv in a carui vecinatate exista cel putin 2 directii dominante (diferite) ale gradientului. Punctele de colt au o localizare bine definita in imagine iar detectia lor necesita folosirea unor metode robuste care sa permita detectia aceluiasi set de colturi in conditii de iluminare diferita, sau daca imaginea obiectul sufera anumite transformari geometrice (rotatie, scalare, transformare de perspectiva).

Exista numeroase metode de detectie a punctelor de tip colt. Dintre acestea se va prezenta metoda Harris. Ea a fost propusa de Moravec [1] si a stat la baza a numeroase variante derivate: Harris & Stephens sau Shi/Kandane & Tomasi [1] care se regasesc si in implementarile din OpenCV.

Pentru localizarea punctelor de colt metoda Harris propune utilizarea matricii de autocorelatie (covarianta a derivatelor) M care contine totii operatorii diferentiali care descriu geometria "suprafetei" imaginii in punctul (x,y) (din punctul de vedere al variatiilor locale de intensitate) [7]:

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_{xy} & I_y^2 \end{bmatrix} = \begin{bmatrix} A & C \\ C & B \end{bmatrix} \quad (4.1)$$

unde:

I_x, I_y sunt componentele orizontale respectiv verticale ale gradientului [2] care in ecuatia (4.1) sunt ridicate la patrat sau inmultite intre ele si sunt ponderate cu o fereastră W de ponderi gaussiene.

Matricea de autocorelatie fiind simetrica fata de diagonala secundara, poate fi diagonalizata prin rotatia axelor de coordonate la forma:

$$M = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}, \quad (4.2)$$

unde:

λ_1, λ_2 sunt valorile proprii ale matricii de autocorelatie

Pe baza matricii de autocorelatie se poate calcula matricea de raspuns $R(x,y)$ care este o masura a „tariei” coltului in fiecare pixel $p(x,y)$:

$$R(x,y) = \det(M) - k * \text{trace}(M) \quad (4.3)$$

unde:

$\det(M) = \lambda_1 * \lambda_2 = A*B - C^2$ este determinantul matricii M

$\text{trace}(M) = \lambda_1 + \lambda_2 = A + B$ este suma matricii M

$k = 0.04 \dots 0.15$ (este o constanta)

Punctele de colt corespund unor locatii in care matricea de raspuns are o valoare pozitiva mare iar valorile proprii λ_1, λ_2 au o valoare mare. Punctele in care functia de raspuns are o valoare negativa iar intre valorile proprii este diferenta semnificativa de valoare corespund unor puncte de

muchie, iar zonele din imagine in care functia de raspuns si valorile proprii au o valoare mica corespund unor zone netede din imagine (fara textura. Muchii sau colturi) – vezi figura 4.1.

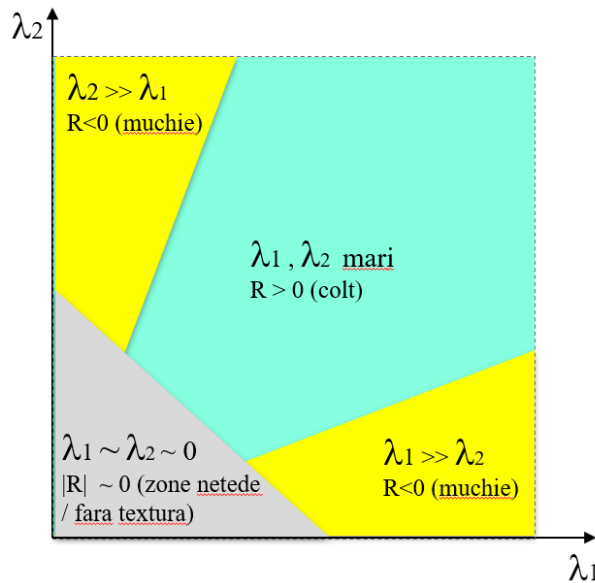


Fig. 4.1. Reprezentare vizuala a semnificatiei valorilor proprii ale matricei de autocorelatie \mathbf{M} si ale matricei de raspuns \mathbf{R} .

Metoda Harris de detecie a colturilor poate fi sintetizata prin algorirmul urmator [3]:

Metoda Haris:

1. Pt. fiecare pixel din imagine $p(x, y)$ se calculeaza matricea de autocorelatie \mathbf{M} .
2. Se calculeaza “harta” (matricea) functiei de raspuns $\mathbf{R}(x, y)$ (in fiecare pixel $p(x, y)$).
3. Se filtreaza $\mathbf{R}(x, y)$ pe baza unui prag T (daca $\mathbf{R}(x, y) < T \Rightarrow \mathbf{R}(x, y) = 0$). Pragul T se poate alege prin analiza histogramei valorilor matricii $\mathbf{R}(x, y)$.
4. Se aplica algoritmul de supresie a non-maximelor pe matricea $\mathbf{R} \Rightarrow$ se retin maximele locale din fiecare vecinatate de dimensiune $N \times N$ de la fiecare locatie (x, y) (valorile din \mathbf{R} care nu corespund unor maxime locale se elimina: $\mathbf{R}(x, y) = 0$).
5. Toate punctele ramase ($\mathbf{R}(x, y) > 0$) vor fi colturile detectate.
6. Optional se poate limita numarul maxim de colturi raportate.

4.2. Detectia de colturi folosind functia OpenCV *goodFeaturesToTrack()*

Implementarea a doua variante ale metodewi Harris se regaseste in functia OpenCV *goodFeaturesToTrack*. In functie de valoarea parametrului de intrare *useHarrisDetector* este apelata metoda Harris (true) sau Shi-Tomasi (false). Semnificatiile parametrilor functiei sunt descrise mai jos:

```
// Lista/vector de iesire care va contine coordonatele (x,y) ale colturilor detectate
(output)
vector<Point2f> corners;

// Nr. maxim de colturi luate in considerare. Daca nr. de colturi > maxCorners se vor
considera cele cu raspuns R maxim
int maxCorners = 100;
```

```
// Factor cu care se multiplica masura de calitate a celui mai bun colt (val. proprie minima) pt. metoda Shi-Tomasi respectiv valoarea functiei de raspuns R (Harris)
ex: qualityMeasure = 1500, qualityLevel = 0.01 ⇒ colturile cu valoarea mai mica de 1500*0.01 sunt rejectate:
double qualityLevel = 0.01;
```

Corner quality:

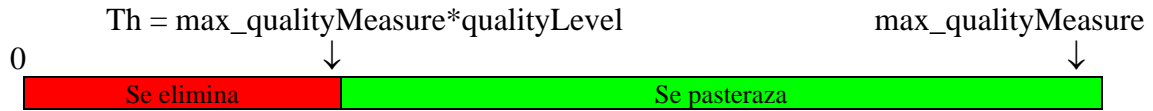


Fig. 4.2. Reprezentare vizuala a modului de calcul a pragului Th folosit la discriminarea bvalorilor matricii de raspun R in puncte de colt / non-colt.

```
// Raza vecinatatii pe care se aplica etapa de supresie a non-maximelor
double minDistance = 10;

// Dimensiunea ferestrei de integrare w folosita in calculul matriciei de autocorelatie M (covarianta a derivatelor)
int blockSize = 3; // 2,3, ...

// Selectia metodei de detectie: Harris (true) sau Shi-Tomasi (false).
bool useHarrisDetector = true;

// Factorul k (vezi descrierea teoretica a metodei)
double k = 0.04;

// Apel functie
goodFeaturesToTrack( Src_image_gray,
                    corners,
                    maxCorners,
                    qualityLevel,
                    minDistance,
                    Mat(), //masca pt. ROI - optional
                    blockSize,
                    useHarrisDetector,
                    k );
```

Functia *goodFeaturesToTrack* detecteaza cele mai proeminente ("de calitate" = care au functia de raspuns cea mai mare) colturi din imagine

1. Calculeaza masura de calitate a colturilor in fiecare pixel al imaginii folosind functiile `cornerMinEigenVal()` daca `useHarrisDetector = false` sau `cornerHarris()` daca `useHarrisDetector = true`.
2. Colturile cu valoare $quality_measure > Th$ se pastreaza (zona verde), celelalte (zona rosie) se elimina (vezi fig. 4.2). Colturile ramase se sorteaza descrescator.
3. Se aplica *non-maximal suppression*: functia elimina orice colt pentru care exista intr-o vecinatate cu raza `maxDistance` un alt colt mai "puternic" (cu $quality_measure$ mai mare).

Algoritmul Harris este prezentat in detaliu in cursul nr. 4 dal disciplinei Interactiune Om-Calculator [4] iar metoda Shi-Tomasi in [5].

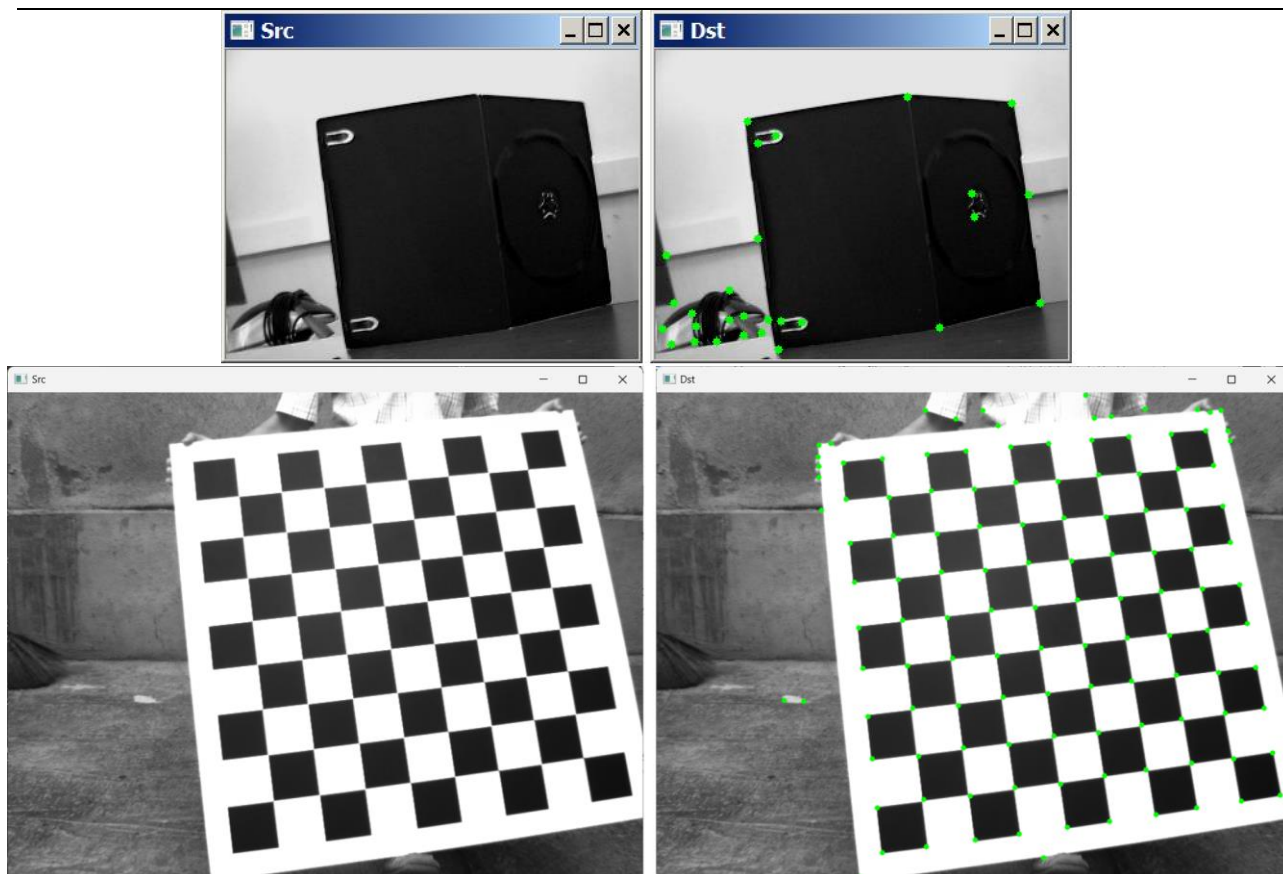


Fig. 4.3. Ilustrarea rezultatului detectie de colturi folosind functia *goodFeaturesToTrack* pe diverse imagini (`useHarrisDetector = true`).

4.3. Activitati practice

1. Se va adauga o functie de procesare pentru detectia colturilor prin apelul metodei *goodFeaturesToTrack*.

Observatie: Pentru a putea detecta colturile indiferent de tipul imaginii sursa (grayscale sau RGB 24) si pentru a putea marca colturile detectate in imaginea destinatie (ex. cercuri de culoare verde) va recomandam sa faceti urmatoarele:

- cititi imagine asursa astfel: `Src_image = imread(fname, IMREAD_COLOR);`
- clonati sursa in destinatie: `Dst_image = Src_image.clone();` La final, peste aceasta imagine veti desena colturile detectate ...
- convertiti sursa intr-o imagine grayscale (`cvtColor`) si apoi sa aplicati un filtru trece-jos gaussians (`GaussianBlur`) inainte de a apela metoda de detectie a colturilor.
- sa vizualizati colturile detectate in imaginea destinatie prin parcurgerea vectorului `corners` si desenarea unor cercuri cu raza 3 si culoarea verde (`Scalar(0, 255, 0)`) in jurul punctelor de colt.

2. Se va vizualiza rezultatul metodelor de detectie a colturilor din *goodFeaturesToTrack* pentru diferite valori ale parametrilor de intrare: `maxCorners`, `qualityLevel`, `minDistance`, `blockSize`, `useHarrisDetector = true/false`, `k`.

3. Se va adapta functia implementata pentru punctul 1 pentru a calcula coordonatele rafinate ale colturilor detectate cu precizie de sub-pixel si se vor afisa intr-un fisier text. Metoda de calcul a

coordonatelor colturilor cu precizie de sub-pixel este descrisa in tutorialul [6]. Pentru aceasta se initializeaza parametrii necesari:

```
Size winSize = Size( 5, 5 );
Size zeroZone = Size( -1, -1 );
TermCriteria criteria = TermCriteria( TermCriteria::EPS + TermCriteria::COUNT, 40, 0.001 );
```

Se calculeaza locatiile rafinate ale colturilor astfel:

```
cornerSubPix(Src_image_gray, corners, winSize, zeroZone, criteria );
```

si se scriu intr-un fisier text coordonatele (corners[i].x, corners[i].y) cu precizie de 2 zecimale.

4. Se va integra metoda Harris (vezi functia *cornerHarris_demo*) [7] ca o functie noua de procesare in OpenCVApplication. Aceasta functie calculeaza functia de raspuns **R**, ii normalizeaza valorile in intervalul(0 .. 255) si apoi afiseaza colturile a caror valoare normalizata a raspunsului este peste un prag.

Cum interpretam rezultatul (vezi figuta 4.2):

- `dst = Mat::zeros(src.size(), CV_32FC1)` - va contine functia de raspuns $R(x,y)$
- aceasta valoare se normalizeaza in interv. 0..255 si se pune intr-o imagine grayscale (1 canal): `dst_norm_scaled`
- Zonele netede din imagine ($R(x,y) \approx 0$) sunt mapate in nuante de gri
- Punctele de muchie ($R(x,y) < 0$) sunt mapate in nuante inchise (negru)
- Punctele de colt ($R(x,y) > 0$) sunt mapate in nuante deschise (alb)

Observatie: se poate observa ca unele colturi lipsesc ($R < thresh$) iar altele sunt incercuite de mai multe ori (puncte vecine care au functia de raspuns peste prag). Pentru a elimina raspunsurile multiple pt. un colt puteti aplica „metoda de supresie a non-maximelor” [4].

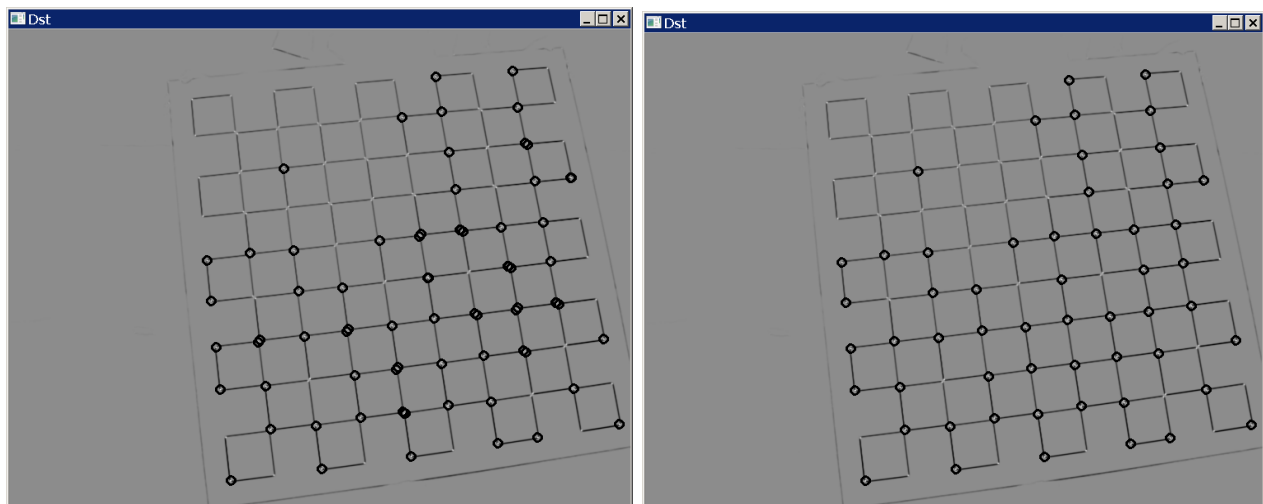


Fig. 4.4. Ilustrarea rezultatului implementarii *cornerHarris_demo* ($thresh=200$) fara supresia non-maximelor (stanga) si cu supresia non-maximelor pe o vecinatate de dimensiune 11x11 (dreapta).

5. Adaugati o functie de procesare care sa realizeze detectia de colturi pe secvente video sau secvente live captate de la webcam. Folositi ca si sablon de procesare functia `testVideoSequence()` din exemple existente in proiectul *OpenCVApplication* si integrati procesarea facuta la punctul 1 din mersul lucrarii.

4.4. Bibliografie

- [1] Metode de detectie a punctelor de tip colt, http://en.wikipedia.org/wiki/Corner_detection, citat 2024.
- [2] S. Nedevschi, T. Marita, R. Danescu, F. Oniga, R. Brehar, I. Giosan, C. Vancea, R. Varga, Procesarea Imaginilor - Îndrumător de laborator, editia a 2-a, Editura U.T. Press, Cluj-Napoca, <https://biblioteca.utcluj.ro/carti-online-cu-coperta.html>, 2023.
- [3] A. Koschan, M. Abidi, Digital Color Image Processing, Wiley & Sons, 2008, cap 6, pag 143 - 144.
- [4] Interactiune Om-Calculator, Note de curs, Curs 4: <http://users.utcluj.ro/~tmarita/HCI/C4.pdf> .
- [5] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, June 1994.
- [6] The OpenCV Tutorials, (\doc\opencv_tutorials.pdf), 6.7 Detecting corners location in subpixeles. https://docs.opencv.org/4.9.0/dd/d92/tutorial_corner_subpixels.html
- [7] The OpenCV Tutorials, Release (\doc\opencv_tutorials.pdf), 6.2 Harris corner detector. https://docs.opencv.org/4.9.0/d4/d7d/tutorial_harris_detector.html
- [8] The OpenCV Tutorials, Release (\doc\opencv_tutorials.pdf), 6.5 Shi-Tomasi corner detector, https://docs.opencv.org/4.9.0/d8/dd8/tutorial_good_features_to_track.html