

5. Detectia punctelor de interes de tip colt

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

Se va adauga o functie pentru detectia de colturi din imagine. Pentru aceasta se va folosi functia OpenCV *goodFeaturesToTrack*. In functie de valoarea parametrului de intrare *useHarrisDetector* este apelata metoda Harris (true) sau Shi-Tomasi (false).

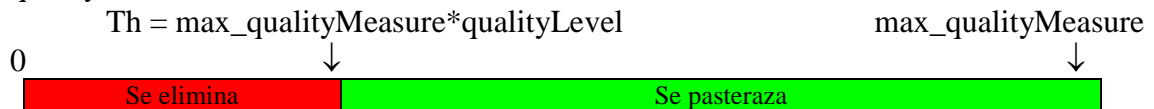
```
// Parametrii functiei

// Lista/vector 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:



```
// Distanta euclidiană minima dintre 2 colturi returnate (functia elimina orice colt vecin cu
coltul curent aflat la o distanta mai mica de 10 pixeli si care are masura de calitate mai
mica ~ metoda Non-Maxima Suppression)
double minDistance = 10;
```

```
// Dimensiunea ferestrei w in care se calculeaza matricea de autocorelatie (covarianta a
derivatelor)
int blockSize = 3; // 2,3, ...
```

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

```
// Factorul k (vezi documentatia curs)
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 sau din regiunea ROI (daca este specificata) dupa urmatoorii pasi:

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. de mai sus). Colturile ramase se sorteaza descrescator.
3. Aplica *non-maximal suppression*: functia elimina orice colt pentru care exista intr-o vecinatate de mai apropiata de `maxDistance` (distanța euclidiană) de un alt colt mai "puternic" (cu $quality_measure$ mai mare).

Algoritmul Harris este prezentat in detaliu in cursul nr. 4 de IOM [1] iar metoda Shi-Tomasi in [2].

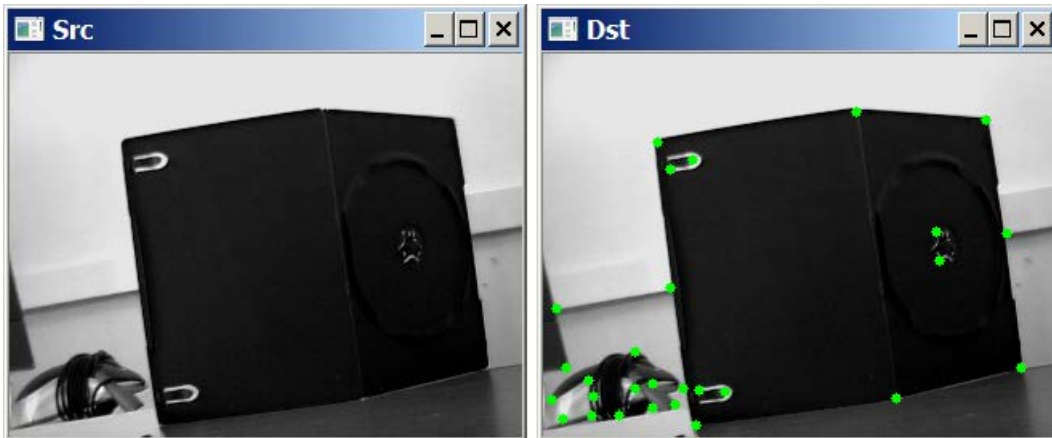


Fig. 1. Ilustrarea rezultatului detectie de colturi folosind functia `goodFeaturesToTrack`.

5.2. Mersul lucrării

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, CV_LOAD_IMAGE_COLOR);`
- clonati sursa in destinatie: `Dst_image = Src_image.clone();` La final. peste aceasta imagine veti desena colturile detectate ...
- sa convertiti sursa intr-o imagine grayscale (`cvtColor`) si apoi sa aplicati un filtru gauss (`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 [3]. Pentru aceasta se initializeaza parametrii necesari:

```
Size winSize = Size( 5, 5 );
Size zeroZone = Size( -1, -1 );
```

```
TermCriteria criteria = TermCriteria( CV_TERMCRIT_EPS + CV_TERMCRIT_ITER, 40, 0.001 );
```

se calculeaza locatiile rafinate ale colturilor:

```
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 incerca integrarea metodei Harris (vezi functia *cornerHarris_demo*) [4] ca o functie noua de procesare in *OpenCVApplication*. Aceasta functie calculeaza functia de raspuns R , o normalizeaza (0 .. 255) si apoi afiseaza colturile a caror valoare normalizata a raspunsului este peste un prag.

Cum interpretam rezultatul (vezi desen slide 18 / C4):

- $dst = \text{Mat}::\text{zeros}(\text{src.size}(), \text{CV_32FC1})$ - va contine functia de raspuns $R(x,y)$
- aceasta valoare se normalizeaza in interv. 0..255 si se pune intr-o imagine grayscale (1 channel): 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” (vezi C4).

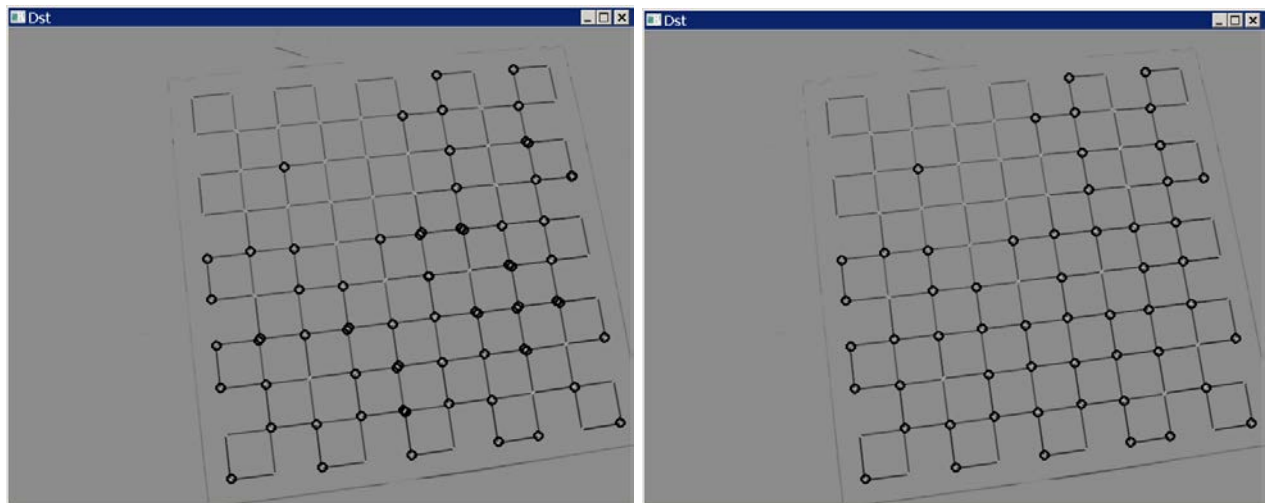


Fig. 2. Ilustrarea rezultatului folosind functia *cornerHarris_demo* ($thresh=200$) fara supresia non-maximelor (stanga) si cu supresia non-maximelor pe o vecinatate de dimensiune 11x11 (dreapta).

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 *OpenCV Application* si integrati procesarea facuta la punctul 1 din mersul lucrarii.

Referințe

[1] Interactiune Om-Calculator, Note de curs, Curs 4: <http://users.utcluj.ro/~tmarita/HCI/C4.pdf> .

[2] 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.

[3] The OpenCV Tutorials, (\doc\opencv_tutorials.pdf), 6.7 Detecting corners location in subpixeles. https://docs.opencv.org/3.4.11/dd/d92/tutorial_corner_subpixels.html

[4] The OpenCV Tutorials, Release (\doc\opencv_tutorials.pdf), 6.2 Harris corner detector. https://docs.opencv.org/3.4.11/d4/d7d/tutorial_harris_detector.html

[5] The OpenCV Tutorials, Release (\doc\opencv_tutorials.pdf), 6.5 Shi-Tomasi corner detector. https://docs.opencv.org/3.4.11/d8/dd8/tutorial_good_features_to_track.html

[6] https://docs.opencv.org/3.4.11/d9/d97/tutorial_table_of_content_features2d.html