

# **SISTEM DE VEDERE ARTIFICIAL BIDIMENSIONALA PENTRU RECUNOASTEREA FORMEI, CULORII SI IDENTIFICAREA POZITIEI SI ORIENTARII PIESERLOR PE CONVEIOR**

**CEBOTARENCO NICOLAE,**

Conducător științific: Prof.dr.ing. **Adrian NICOLESCU**

**REZUMAT:** Pentru tema de cercetare s-a stabilit ca obiect de studiu o aplicația robotizată de ambalare a produselor alimentare, utilizând ca fundație studiul aprofundat în proiectul de diplomă și anume o pereche de doi roboți integrați cu conveioare de transport independent al produselor și ambalajelor, sisteme de vedere artificială integrați într-o linie de producție automatizată a produselor alimentare.

Având toate acestea în vedere, am hotărât să aprofundez acest proiect prin dezvoltarea unei aplicații software care să înlocuiască sistemul video prin recunoașterea obiectelor alimentare pe conveior în baza culorii, formei și poziției a produselor.

## **1 INTRODUCERE**

Fundamentul acestei lucrări a fost elaborat în baza studiilor pentru proiectul de diplomă în anii 2014-2015. În urma studiilor și testelor de atunci s-a identificat o mulțime de erori la nivelul detectării și scanării pieselor pe conveior. În această lucrare am atins performanțe aproape industriale la nivelul sistemului de control video prin recunoașterea culorii, formei și orientarea obiectului în poziție pe conveior.

## **2 STADIUL ACTUAL**

În studiul actual am decis să elaborez acest soft care să poată atinge următoarele puncte majore:

- Verificarea prezentei obiectului pe conveior, indiferent de culoare și forma acestuia.
- Procesarea și detecția formei scanate de pe conveior.
- Detectarea conturului și/sau obiectului excluzând culoarea conveierului.
- Detectarea formei obiectului în dependența de contur.
- Filtrarea culorii pe forma pieselor detectate (identificarea punctelor de interes).
- Determinarea orientării obiectului (în plan XY).
- Colectarea și depunerea orientată a piesei.

---

<sup>1</sup> Specializarea: Master Robotica, Facultatea IMST;  
E-mail: cebotarencu.nicolae@yahoo.com

### **2.1 Verificarea prezentei obiectului pe conveior**

#### *2.1.1 Procesarea și detecția formei scanate de pe conveior.*

Procesarea și detecția formei scanate de pe conveior, se face cu ajutorul camerei web cu care captăm imaginile video. Aceste imagini video sunt împărțite într-o mulțime de cadre pe secunda, unde fiecare cadru din imagine este scanat. În așa fel se poate detecta conturul sau chiar și piesa, excluzând culoarea de fundal al conveierului, cu ajutorul căruia acesta sunt transportate.

#### *2.1.2 Detectarea formei obiectului în dependența de contur.*

Procesarea și detecția formei scanate de pe conveior, se face cu ajutorul camerei web cu care captăm imaginile video. Aceste imagini video sunt împărțite într-o mulțime de cadre pe secunda, unde fiecare cadru din imagine este scanat. În așa fel se poate detecta conturul sau chiar și piesa, excluzând culoarea de fundal al conveierului, cu ajutorul căruia acesta sunt transportate.

Exemplu:

Softul detectează un contur a cărui extreme determină un număr de 3 drepte. Sistemul numără dreptele rezultate și le asociază cu laturile unui triunghi. La fel este și în cazul unui contur cu 4 drepte care le asociază cu o formă rectangulară adică pătrat sau dreptunghi.

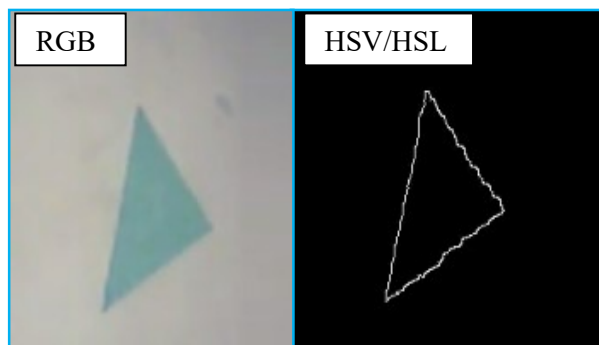


Fig.1 Detectarea conturului unui triunghi

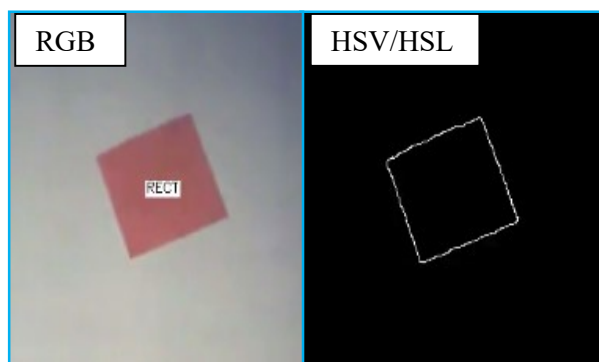


Fig.2 Detectarea conturului unui piese rectangularare

Caz special este in momentul in care sistemul determină distanțe diametral opuse fata de centrul conturului, la fel cu o marja de toleranta de 10% le unește, astfel in cat rezulta un poligon. Aici am aplicat o corecție de forma pentru contururile ce depășesc mai mult de 10 laturi si le determina ca fiind cercuri.

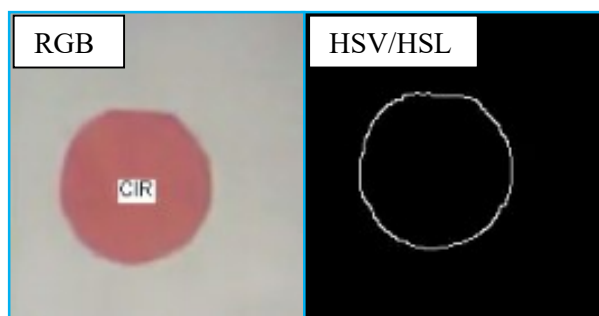


Fig.3 Detectarea unui contur circular

### 3 DETERMINAREA ORIENTĂRII OBIECTULUI IN PLANUL XY

După etapa de Detectarea formei obiectului in dependenta de contur, noi putem determina orientarea obiectului in plan cu ajutorul unei simple teorii geometrice si anume teoria unghiurilor alterne interne.

#### 3.1.1 Detectarea formei obiectului in dependenta de contur.

Prin simpla determinare a unei laturi a pătratului si prin compararea ei cu normala pe care o stabilim, noi putem determina cu cate grade este rotit acest pătrat.

Exemplu:

Rotirea triunghiului

La triunghi determinam mai întâi ce-a mai lunga latura care este si ipotenuza. Apoi identificam distanta ce-a mai mare dintre un punct de pe contur si ipotenuza dar obligatoriu dreapta care unește acel punct trebuie sa fie perpendiculara pe ipotenuza. Astfel am aflat poziția exacta a triunghiului știind ipotenuza si vârful unghiului opus. Intersecția lor ne da si centrul de rotație la care ne raportam atunci când calculam unghiul de rotație fata de nominalul stabilit.

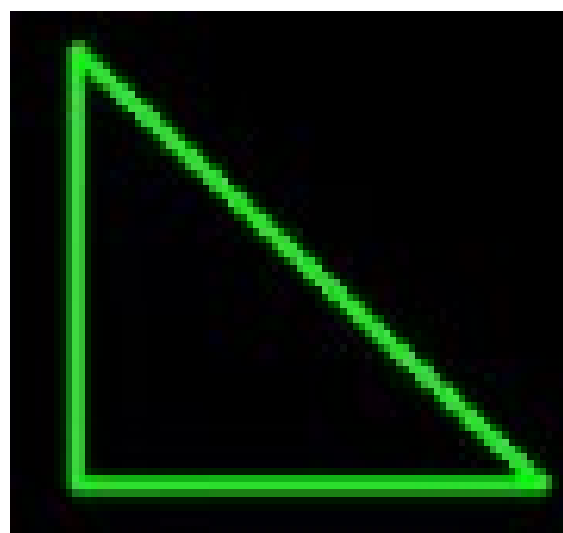


Fig.4 Declararea poziției necesare pentru orientare

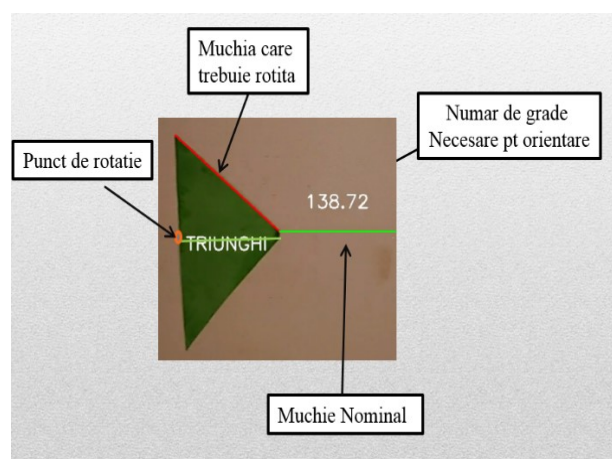


Fig.5 Detectarea poziției si orientarea unui triunghi

### Rotirea pătratului

La Pătrat se determină 2 laturi opuse la vârf, astfel încât să putem găsi diagonala care este de fapt și dreapta pe care o rotim cu câte grade ne sunt necesare pentru a ajunge la nominal

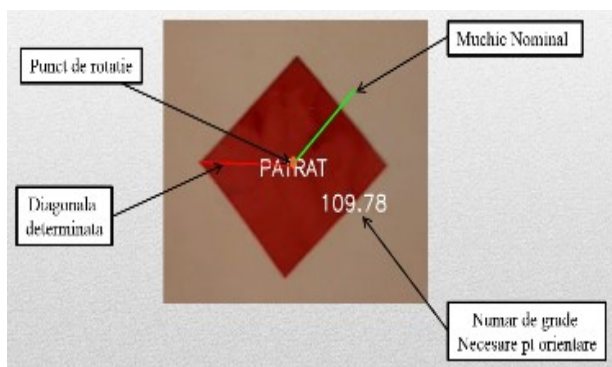


Fig.5 Detectarea poziției și orientarea unui pătrat

### 3.2 Manipularea imaginilor

În această aplicație se vor prezenta diferite forme de reprezentare a imaginilor, modalități de reprezentare a culorilor, cât și două metode de filtrare spațială a imaginilor, mai exact filtrul median și filtrul Gaussian.

### 3.3 Reprezentarea imaginilor

O imagine digitală poate fi reprezentată printr-o funcție  $f(x, y)$ , unde parametrii funcției reprezintă coordonatele imaginii pe cele două direcții ale lui  $R^2$ , așa cum se poate vedea și în Fig. De mai jos. O formă compactă de scriere a unei imaginii digitale monocrome este dată sub forma unei matrici ale cărei elemente sunt denumiți pixeli [1]:

$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \dots & f(0, N-1) \\ f(1, 0) & f(1, 1) & \dots & f(1, N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1, 0) & f(M-1, 1) & \dots & f(M-1, N-1) \end{bmatrix}$$

Unde valorile pe care le poate lua  $f(x, y)$  aparțin unui domeniu finit:

$$0 \leq f(x, y) \leq L - 1.$$

Valoarea lui  $L$  este, în general, 256 pentru imagini gri. Acest proces de reprezentare a

imaginilor, denumit eșantionare, este exemplificat în Fig. De mai jos pentru cazul unei imagini de nivel gri definită pe 8 biți. În cazul imaginilor color,  $f(x, y)$  va conține cele trei culori fundamentale (Roșu, Albastru și Verde), ea putând fi reprezentată matematic în două forme:

– Prin intercalarea pixelilor, unde fiecare linie este o matrice 2D, cu fiecare element reprezentând o listă cu trei valori;

– Prin intercalarea culorilor, unde pe fiecare linie datele sunt separate în matrici 2D, câte una pentru fiecare canal de culoare:

$$f_{RGB}(x, y) = [f_r(x, y), f_g(x, y), f_b(x, y)].$$

Pentru a se putea realiza o standardizare a culorilor a fost introdusă noțiunea de model al culorilor [1]. Un astfel de model poate fi orientat spre hardware (monitoare sau imprimante) sau spre aplicații software, unde scopul final este acela de manipulare a culorilor.

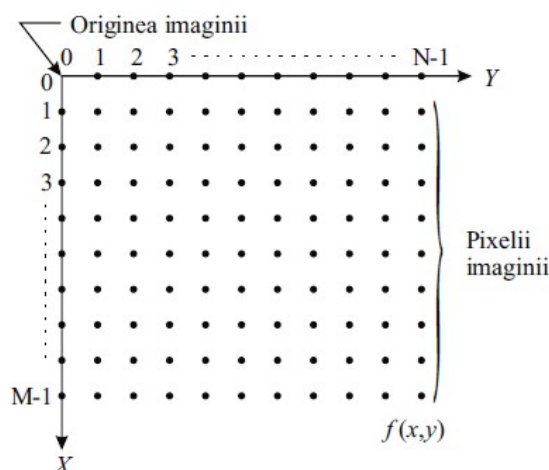


Fig. 6. Coordonatele unei imaginii digitale  $f(x, y)$ .

## 4 FILTRAREA IMAGINILOR

Filtrarea imaginilor este folosită pentru a se reduce zgomotul din imaginile de intrare, fiind definită în domeniul spațial ca o operație de convoluție :

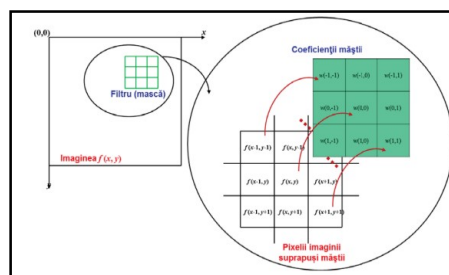


Fig. 7. Reprezentarea operației de filtrare spațială a unei imaginii

Filtrul median este reprezentat de acea mască în care toți coeficienții sunt egali:

$$w(x, y) = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}.$$

## 4.1 Limbajul de programare C++, Codul sursa

### 4.1.1 Captura ecran

```

13 Mat img_in, img_out;
14 string strImagePath;
15
16 // Incarcarea imaginii de pe disk
17 if (argc == 2)
18     strImagePath = argv[1];
19 else
20     strImagePath = "../pcrai.jpg";
21
22 img_in = imread(strImagePath.c_str(), CV_LOAD_IMAGE_UNCHANGED);
23
24 // Verificarea incarcarii corecte a imaginii
25 if (!img_in.data)
26 {
27     cout << "Imaginea " << strImagePath << " nu a putut fi incarcata" << endl;
28     return -1;
29 }
30
31 // Conversia din imagine in culori in imagine gri
32 cvtColor(img_in, img_out, CV_RGB2GRAY);
33 imshow("Culoare -> Gri", img_out);
34 waitKey();
35 destroyAllWindows();
36
37 // Conversia din spatiul de culoare RGB in spatiul HSV
38 cvtColor(img_in, img_out, CV_RGB2HSV);
39 imshow("RGB -> HSV", img_out);
40 waitKey();
41 destroyAllWindows();
42
43 // Accesarea pixelului avand coordonatele (10, 10)
44 Vec3b pxValue = img_in.at<Vec3b>(10,10);
45 cout << "(10,10): R = " <<
46     (int)pxValue[0] << " G = " <<

```

## Descrierea funcțiilor principale ale programului:

```
32 void cvtColor(InputArray src, OutputArray dst, int code)
```

Convertește o imagine dintr-un spațiu de culoare în altul.

- src: imaginea de intrare;
  - dst: imaginea de ieșire;
  - code: codul conversiei de culoare:
- CV\_BGR2HSV, CV\_RGB2HSV, CV\_HSV2BGR, CV\_HSV2RGB;
  - CV\_BGR2HLS, CV\_RGB2HLS, CV\_HLS2BGR, CV\_HLS2RGB;
  - CV\_BGR, CV\_RGB, CV\_Lab2BGR, CV\_RGB.

### 5. Detectarea Formelor

În lucrare de față este utilizată metoda dezvoltată de Viola și Jones [3], ulterior completată de Rainer și Mazdt [4], în care au fost propuse implementări ale unor noi algoritmi pentru detecția de obiecte în timp real. Avantajul

acestei noi modalități de detecție este acela că algoritmi folosiți au un grad de detecție foarte ridicat și pot fi aplicați pentru orice tip de obiect. Una dintre aplicațiile unde pot fi utilizate metodele propuse este aceea a detectării fizionomiei și caracteristicilor faciale. Metoda pentru detectarea fizionomiei are la bază clasificatorul AdaBoost [5], și caracteristici de tip Haar [3, 4], fiind cunoscută și sub denumirea de clasificator Haar [2].

Algoritmul pentru detecția de obiecte propus de Viola și Jones [7] are la bază caracteristici, motivația alegerii acestora în detrimentul alegerii directe a pixelilor fiind dată de faptul că, pe de o parte, caracteristicile pot fi codificate în așa fel încât să conțină într-un singur loc mai multe informații iar, pe de altă parte, viteza de procesare este superioară utilizării directe a pixelilor. În Fig.5 sunt prezentate caracteristicile utilizate de algoritmul pentru detectarea fizionomiei.

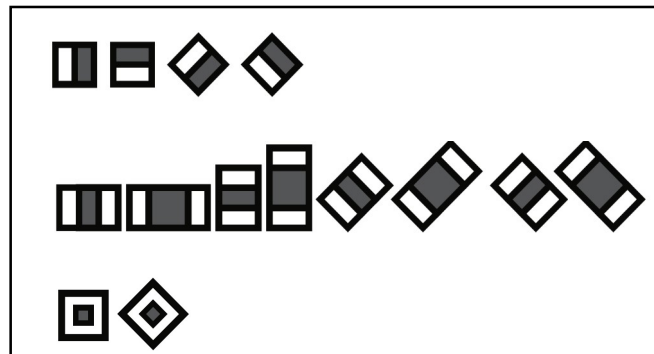


Fig.8,1.Caracteristici de tip muchie, 2.Caracteristici de tip linie, 3.caracteristici centrale înconjurate

### 6. Verificarea prezenței obiectului pe conveyer

Procesarea și detecția formei scanate de pe conveyer, se face cu ajutorul camerei web cu care captăm imaginile video. Aceste imagini video sunt împărțite într-o mulțime de cadre pe secunda, unde fiecare cadru din imagine este scanat.

În așa fel se poate detecta conturul sau chiar și piesa, excluzând culoarea de fundal al conveyerului, cu ajutorul căruia acesta sunt transportate.

### 7. Detectarea formei obiectului în dependența de nivelul de saturație de gri

Se face cu ajutorul unui principiu matematic. În momentul etapei de verificare a prezenței obiectului pe conveyer, se determină așa numite puncte de interes. Aceste puncte de interes nu sunt altceva decât extremele unui contur detectat prin transformarea imaginii în Alb/Negru. Culoare

din imaginile RGB în HSL/HSV, care sunt imagine binară, unde se deduce forma piesei.

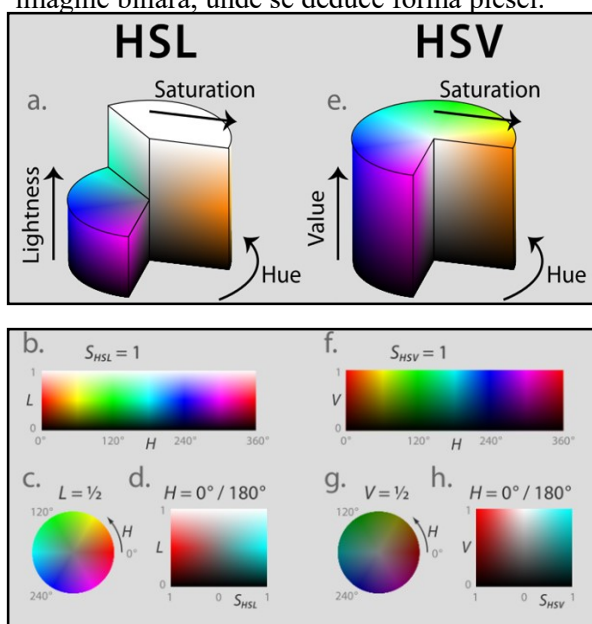


Fig.9 Modul de conversie a pixelilor

## 5 CONCLUZII

În urma metodelor aplicate asupra optimizării cadrelor din imaginile video, putem observa o îmbunătățire remarcabilă a stabilizării imaginii video cât și a vitezei de recunoaștere a acesteia. Față de prima versiune a acestui sistem acum putem detecta poziția mult mai exactă a întregului obiect, nu doar centru cum era în prima fază a studiului de licență datorată faptului că știm cu exactitate conturul pieselor și orientarea lor în plan.

## 6 BIBLIOGRAFIE

- [1] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 3rd Ed. New Jersey, USA: Prentice-Hall, Inc., 2006.
- [2] G. Bradski and A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library. Sebastopol, Canada: O'Reilly Media, 2008.
- [3] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," Proc. of the 2001 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, Vol. 1, Kauai, USA, 2001, pp. 511–518.
- [4] R. Lienhart and J. Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection," Proc. of the 2002 Inter. Conf. on Image Processing ICIP, New York, USA, 2002, pp. 900–903.
- [5] Y. Freund and R. Schapire, "Experiments with a New Boosting Algorithm," Proc. Of The Thirteenth Inter. Conf. on Machine Learning, San Francisco, Canada, 1996, pp. 148–156.
- [14] M. Isard and A. Blake, "Condensation - conditional density propagation for visual Tracking," International Journal of Computer Vision, Vol. 29, pp. 5–28, 1998.
- [7] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," Proc. of the 2001 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, Vol. 1, Kauai, USA, 2001, pp. 511–518.