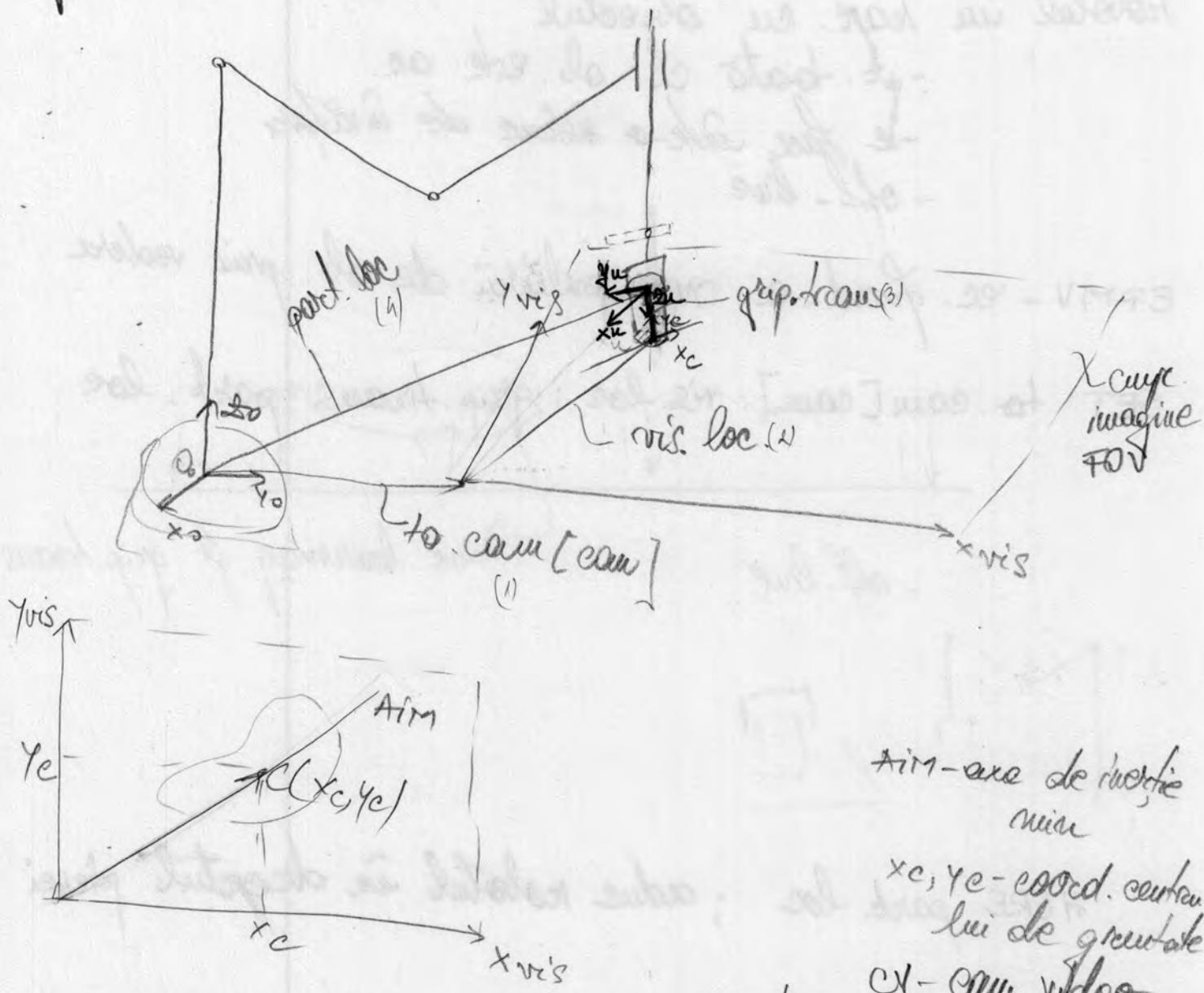
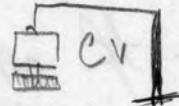


# Legătura vedere - mișcare



aim - axa de mișcare  
mine

$x_c, y_c$  - coord. centru-  
lui de gravitate  
CV - cam. video

SET part.loc = to.cam [cam] : vis.loc : grip.trans (EFM)

(1) to.cam [cam] - ff. camera-robot off-line

- se calc. o sing. dată, în diferentie de apel cu  
virtualitate utilizată cind ca să nu se scrie <sup>relativă</sup> poziție <sub>cam</sub> cu robot.

- se obt. puncte-o sesiune de calibrare a CV.

(2) vis.loc - indică poz. și orientarea ob. în pl. imagine  
locatie

- se obt. pt. fiecare ob. ce treb. accesat de robot  
- VLOCATE

(3). grip.trans - este o f. care arată cum treb. să fie poz. robotel în rap. cu obiectul

- pt. bato cl. ob. este ac.
- se face unk-o sesiune de învățare
- off-line

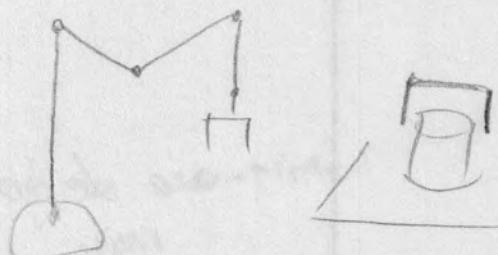
EFMV - ec. fund. a manipulării de ob. prin veștere

SET to.cam [cam] : vis.loc : grip.trans = part. loc.

---

↓ !                    ↓ !                    ↑ ?                    ↓

drive. learning pt. grip.trans



HERE part. loc ; aduc robotul în dreptul prizei

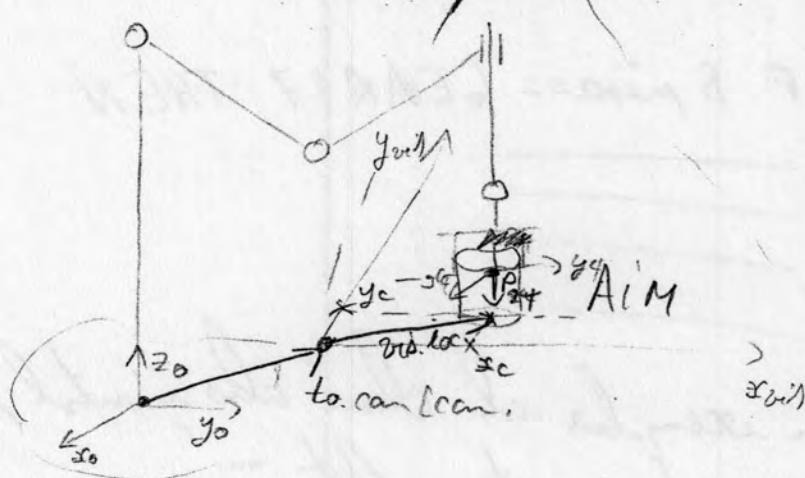
SCR  
curs (S8)

5.04.2012

E. fund. a misc. (EFM)  $\rightarrow$  set. de calib. camera-robot  
SET (part. loc =  $T_0$ . cam (cam))  $\rightarrow$  VLOCATION  $\rightarrow$  t-rat fiz.  
vis. loc. gripl. trans in obiect

Functiile de tip - real

MOVE (s) part. loc



cam = camera virtual.

$\hookrightarrow$  set de pareri

cu g. coroare

interpretare dif

in functie de aplica-

dorita

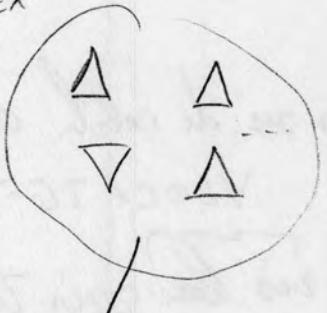
AIM = aceeași înălțime minimă

grip-trans  $\rightarrow$  calc. off-line

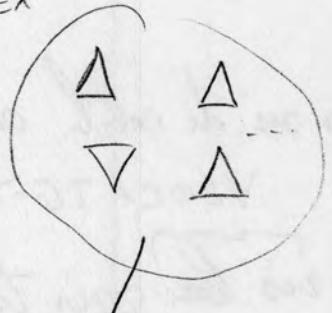
$\rightarrow$  transf. de primire pt. toate obiectele dintr-o clasă

$\rightarrow$  calc. se face în subunitatea de comandă a miscării

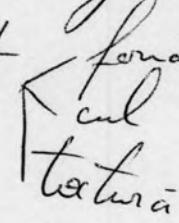
In VA pt rec. (recunoști formă - pattern recognition)

↳ vorbind de obiecte identice care sunt 

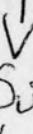
Ex



instante ale  
reprezentării clasei



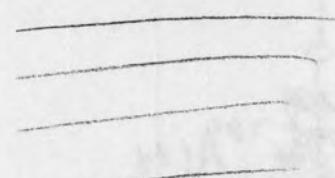
clasei  $\leftarrow$  nume

 Sir de caract ASCII

Ex: "GEAR47"

IF \$piesă == GEAR47 THEN

\$piesă



Prototip, model - un exemplar al clasei care nu este folosit  
a învăța și este folosit să recunoască  
modelul de recunoaștere și modelul de  
localizare

VTRAIN - multiestructură

- ↳ învățarea formă, apetită
- ↳ -II - localizare la

CS / SCR

VTRAIN. FINDER →

VPLAIN. FINDER →

Ex: Într-o scenă am 3 obiecte ob.

VTOUCHING

VOVERLAPPING

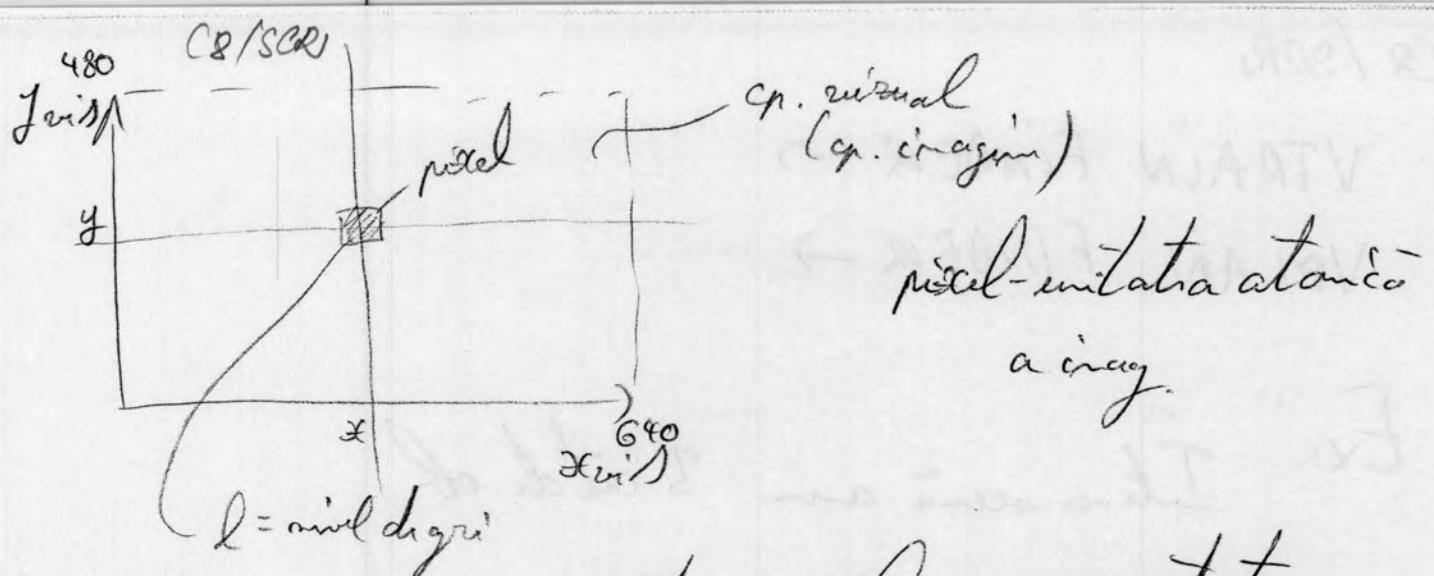
VFINDER - intersectie rec. obiect

Format de reprezentare a imag.

• Imagine analogică  $f(x, y) > 0$   $x, y$  coord. pt. ale unui pixel din imagine  
 $x, y \in \mathbb{R}^+$

• Imagine numerică  $f(x, y) \in \mathbb{Z}^+$

Aceasta imag. numerică se obține prin  
- esantionare pt.  
- quantificare a imag. analogice



Formarea imag.:  $\rightarrow$  captură a lumini proiectate

$$\boxed{f^*} \quad \boxed{8 \text{ bits}} \rightarrow 256 \text{ de nivel digitali}$$

$$l \in [0, 255]$$

$\uparrow \quad \uparrow$   
nume alle

Imagine binarizata

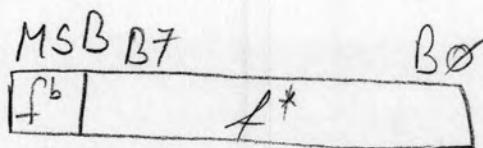
$$\rightarrow f^*(i, j) = \begin{bmatrix} f_{11} & \dots & f_{1m} \\ \vdots & \ddots & \vdots \\ f_{m1} & \dots & f_{mm} \end{bmatrix} \quad \boxed{f^*(i, j)}$$

$$f^b(x, y) = \begin{cases} 0, & f^*(x, y) \leq \text{thr} \\ 1, & f^*(x, y) > \text{thr} \end{cases}$$

$\text{thr} = \text{prag. de binarizare}$   
 $\text{thr} \in (0, 255)$

V. THRESHOLD [cam]  $\rightarrow$  implicit 63

V. 2ND. THRESHOLD [cam]  $\rightarrow$  implicit 0



$$m = \min(V.\text{THRESHOLD}, V.\text{2ND. THRESHOLD})$$

$$M = \max( \dots )$$

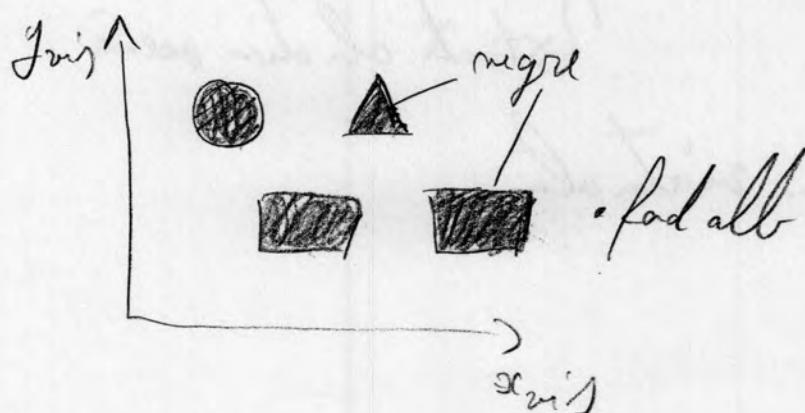
$$f^b(x, y) = \begin{cases} 0, & f^* \in [m, M] \\ 1, & f^* \in [0, m] \cup [M, 255] \end{cases}$$

→ 80% spic di Vision oper. cu imag. binare

Imagini binarizate

VTR AIA. FINDER → multi instrucții care op

cu imag. binar  
(ObjectFinder → principiu)



V. BACKLIGHT

↳ switch

{ 1. fond alb  
0. fond negru

ENABLE      → one - switch  
DISABLE     → one - switch

CS/SCR  
Prel. de imagini

1. Low level image processing

2. H-level image processing

Obs: diferență este separarea obiectelor fără  
de fond

Separare

→ Se crează o BD pt fiecare obiect

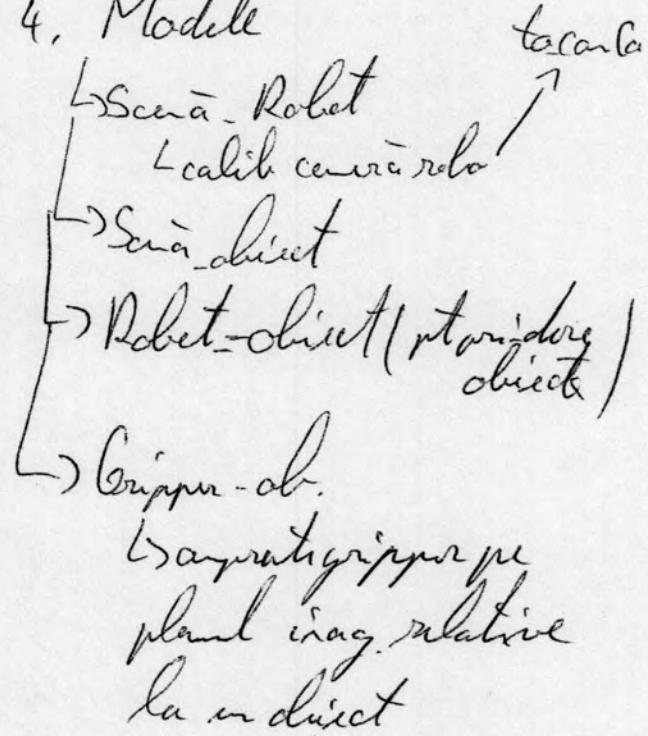
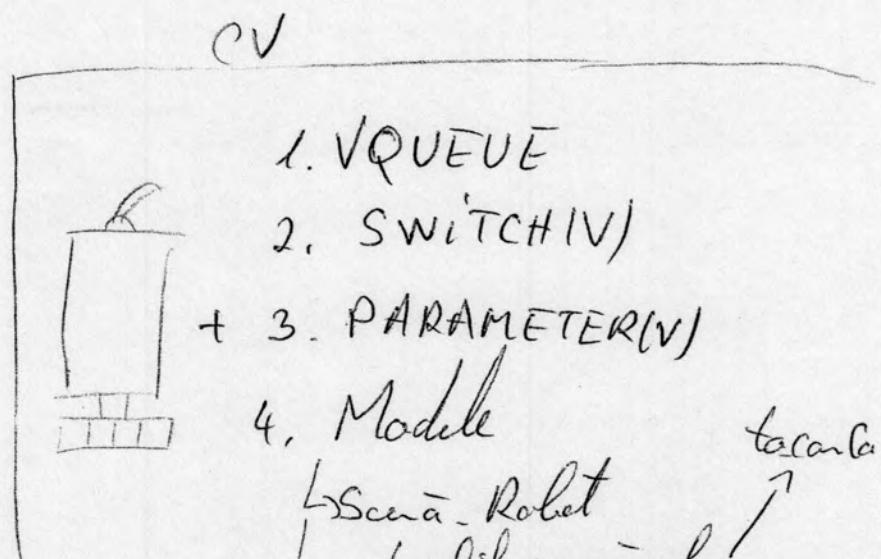
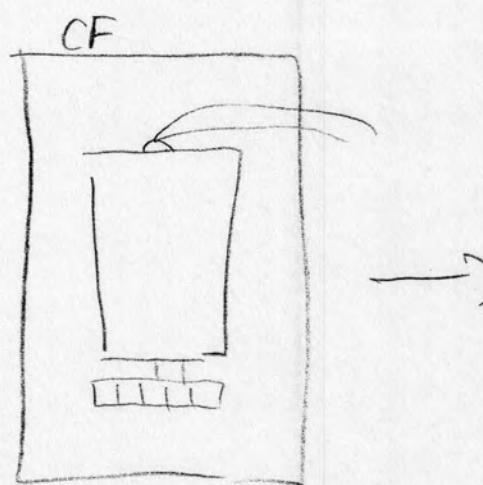
- info. contur
- aria obiect (pixels)
- nr de găuri
- lungime conturului ext
- centru de masă ( $x_c, y_c$ )
- AIM (fiecare obiect)

↳ se află VQUEUE

↳ toată ob din scenă

VQUEUE e Can. virtual

CS/SCC



SCR  
curs(S9)

12.04.2012

miercuri  
joi  
vineri

23-25 mai

Hilton

INCOM - Inform. Central

6-7 "studenti"

↳ 350 participanti

in Manufacturing

WWW. incen12.ro

↳ 6 celebriți ↳ Andrew Crik

TC 5.1 → Paul Valckeniers

↳ Belgian

PROSA - Product Resource

Order Staff Agent

Multi-Agent Systems (MAS)

Product - Driven automation

c9/scr

→ Intelligent Products

↳ Duncan McFarlane  
Kary Fransy

→ abordare Holonică

Manufacturing  
Execution  
System

→ 22 mai → workshop de azi

↳ analiză, proiectare MES



Open-source

Camera virtuală

în mod automat

→ set de date asociate unui camera fizic (sus optici,  
îmagine achiziționată)

→ parametri

→ switch-uri (flag-uri) <sup>unic</sup>

→ modele - de calibrare (Spira - Dobot) <sup>(camera)</sup>  
S-R <sup>parametri</sup>

- de recunoaștere (S-O)

↔ Off-line

pt. fixare  
obiect

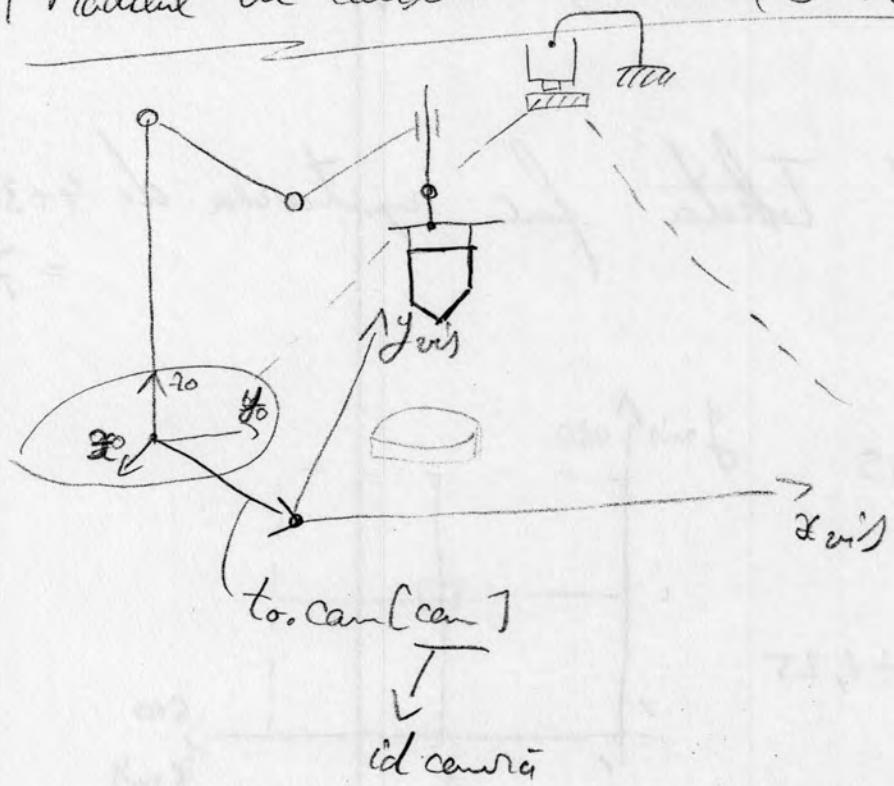
- de printare (R-O) <sup>de evaluare</sup>  
base de ob.

(localizare)

- de extrare colizi (Grippe-Object)

9/50

# Modulul de calibrare C-R/S-R

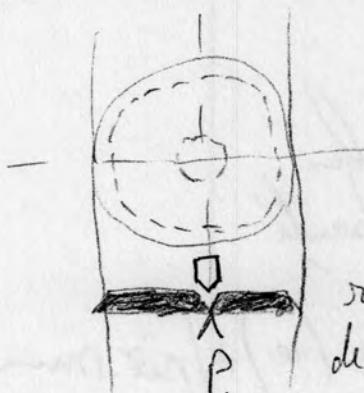


$\text{to.cam[cam]} \rightarrow \text{TRANS}(x, y, z, Rx, Ry, Rz)$

$$x, y, z = R_c \begin{pmatrix} x_{mis}, y_{mis}, z_{mis} \\ x_0, y_0, z_0 \end{pmatrix}$$

Se utilizează un disc circular

$$C(x_c, y_c) + AIM$$



input  
dim. m.  
on

HERE  $r_i \rightarrow$  ref. robot  
DEPARTS 100  
MOVE safe  
VPICTURE (cam, mode, st',  
st 2) how many  
ref. minor

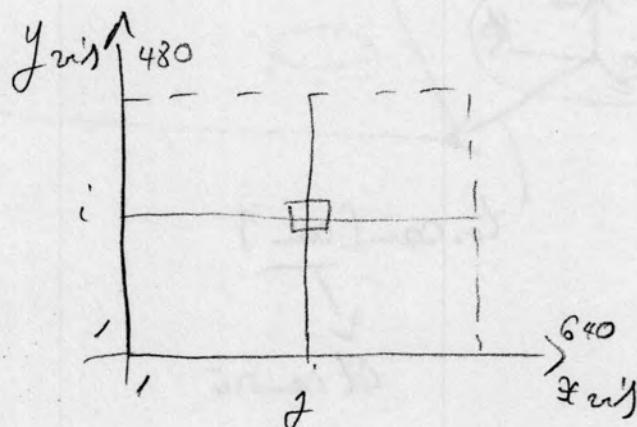
Obs: robotul SCARA are ceva "theta"

9/SCR

V. Consider planul cu 4 cadrele în raport proiectivale totale  
cadrele

- la rob cu ceva înălțime fac ruptarea de  $4+3 \alpha$   
 $= 7$

$\rightarrow$  pix. la. mm - 0,295  
 $\times Y_{scale}$  }  
 $0,75 \div 1,25$



Aria obiectelor

$\rightarrow$  VFEATURE(indice)

$\downarrow$

$\in \overline{f, 50}$

$f^*(i, j)$

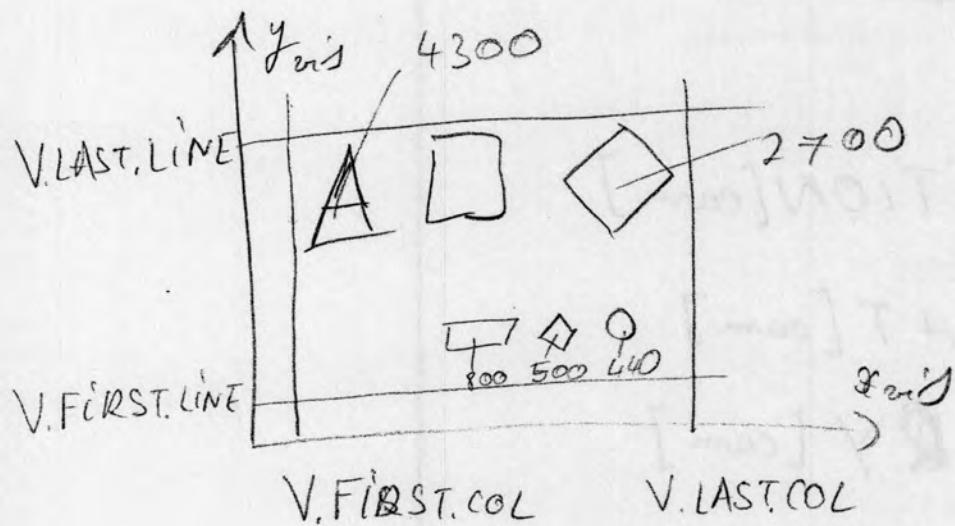
$\rightarrow$  VFEATURE( $10$ ) = aria obiectului  
în pixeli bruni

$$\text{ar. mm}^2 = \text{VFEATURE}(10) * (\text{pix. mm}^{**2}) * \\ XYscale$$

Switch - ~~with~~ ~~parametr~~

- VISION
  - V. RECOGNITION [cam]
  - V. BACKLIGHT [cam]
  - V. BOUNDARY [cam]
  - V. HOLES [cam]
  - V. PERIMETER [cam] → fr. extreccare
  - V. TOUCHING [cam]
  - V. OVERLAPPING [cam]
  - V. DISJOINT [cam]
- 
- V. CENTROID [cam] - surf. calc. coord. centroid  
de mass pt. d. carre apre  
in seria
  - V. 2ND. MOMENTS [cam] \ calc. orient. direttori  
anghiari intre A<sub>1</sub>M<sub>1</sub> e A<sub>2</sub>M<sub>2</sub>

C9/SCR  
PARAM. camera virtuale



V.FIRSTLINE [cam]  
V. LASTLINE [cam]  
V. FIRSTCOL [cam]  
V. LASTCOL [cam]

Ex: PARAMETER name-param = val

V. MIN. AREA [cam]

V. MAX. AREA [cam]

V. MIN. HOLE. AREA [cam]

} filter

Ex:

V. GAIN [cam] → amplif. semialogic

V. OFFSET [cam] → bias. pixel analog mil  
degr 1.

V. THRESHOLD [cam]

V. 2ND. THRESHOLD [cam]

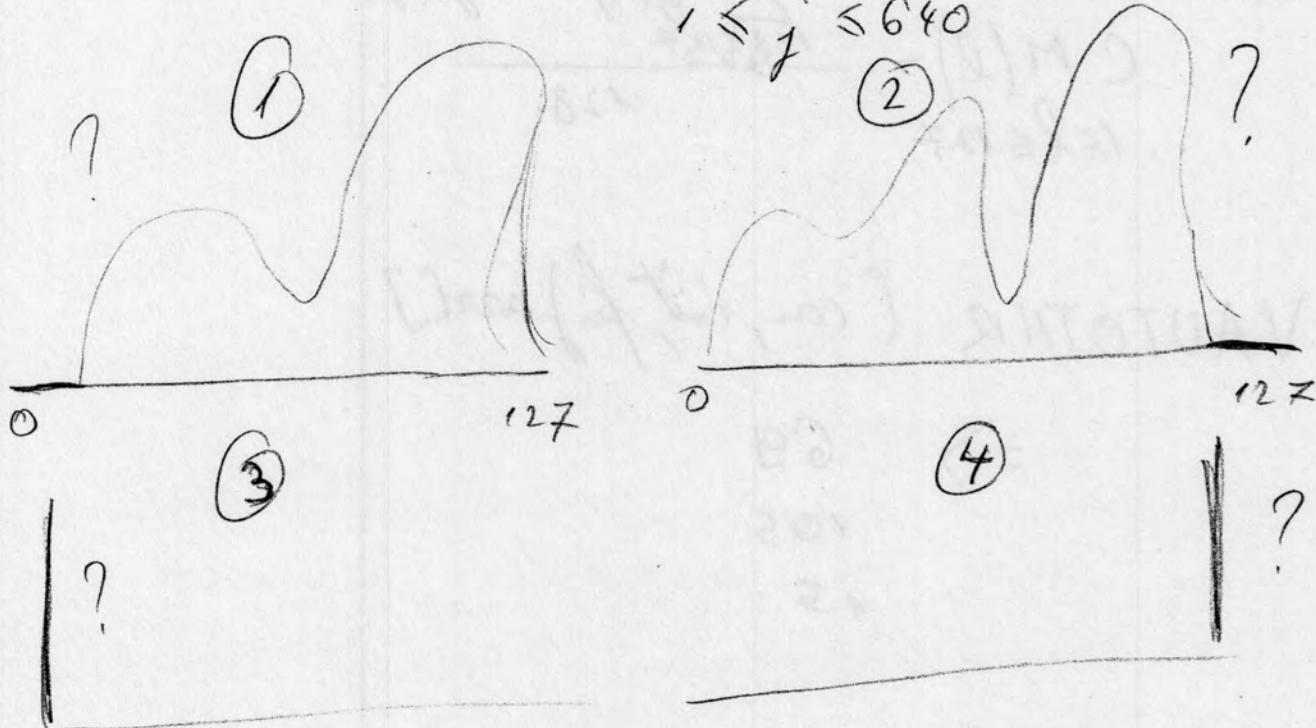
→ histograma imag. en niveles de gris

$$hg: \{0, 1, \dots, 127\} \rightarrow \{0, 1, \dots, 480 \times 640\}$$

$$hg(l) = \sum_{\substack{1 \leq i \leq 480 \\ 1 \leq j \leq 640}} (a_{i,j}) f^{*}(i,j) = l$$

$$1 \leq i \leq 480$$

$$1 \leq j \leq 640$$

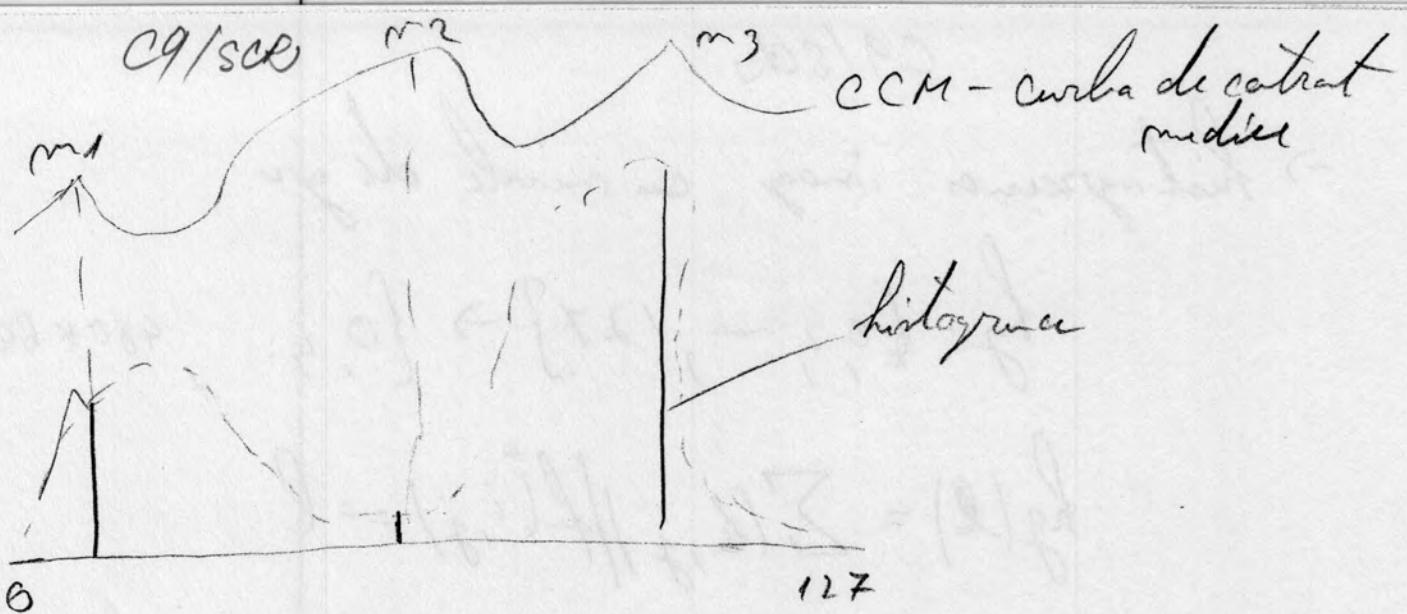


① → n seach VOFFSET

② → n vrgh VOFFSET

③ → n vrgh VGAIN

④ → n seach VGAIN



$$C M(l) = \frac{\sum_{1 \leq z \leq l} h_g(z) - h_g(z)}{128}$$

VAUTOTHR (can, int, fin) arr[]

=> 69

105

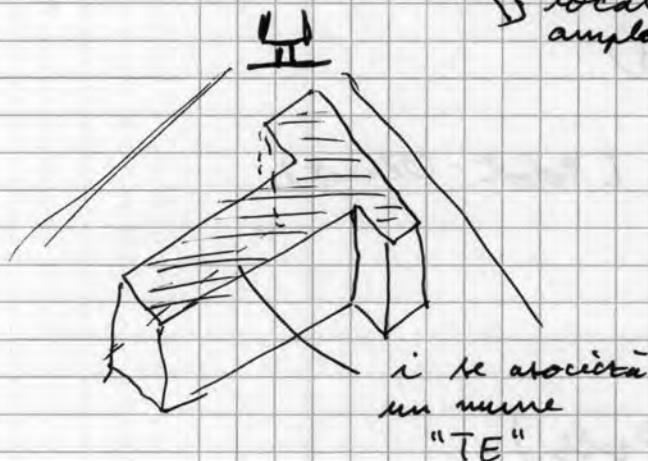
15

## Localizarea obiectelor

- are 2 componente

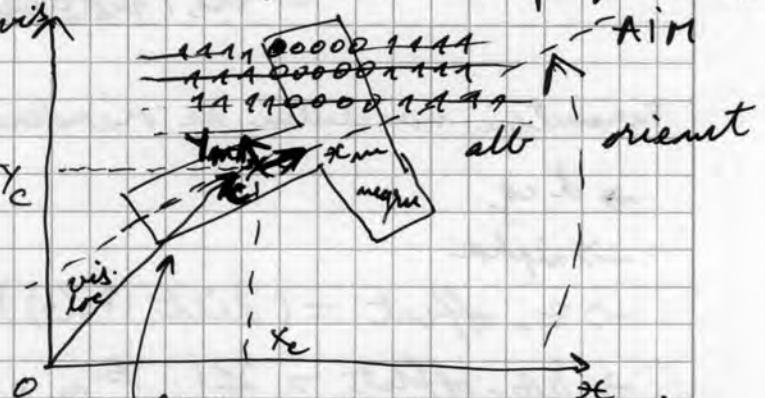
localizare obiect în plan imagine  
(abstracțiere)  $\Rightarrow$  vis. loc

localizare gripez relativ la proiecția  
amploarmentului de prindere  $\Rightarrow$  grip-trans



$x_m, y_m$  - sistem coord obiect

$o_m x_m$  - se aliniază cu AIM

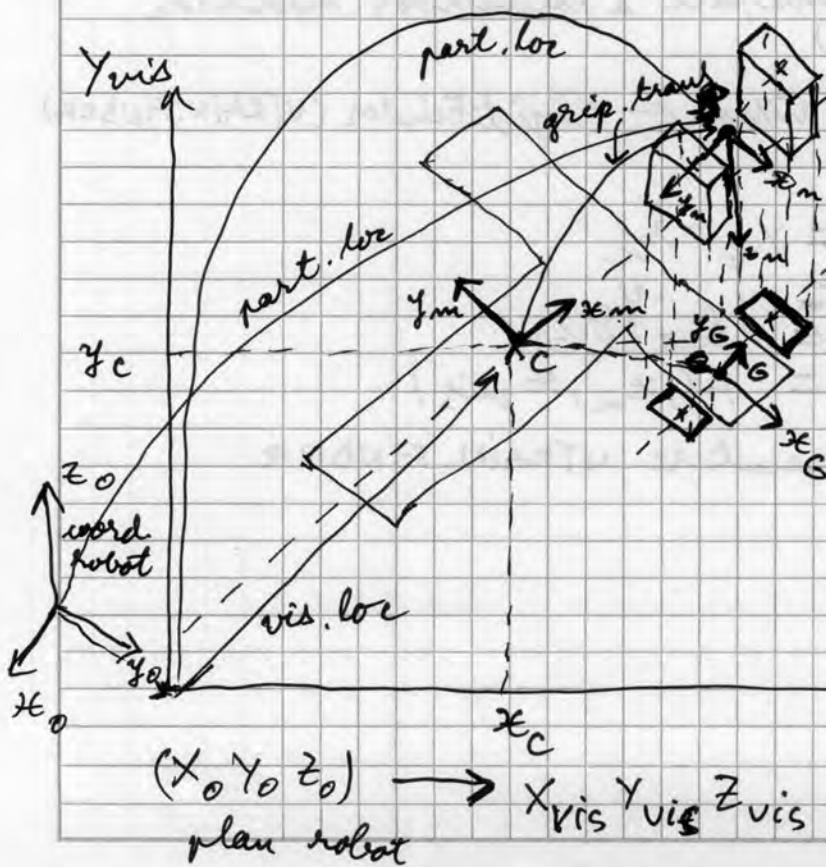


grip-trans  $\Rightarrow$  model de prindere  
(model robot - obiect)

C - centru de masă,  $(x_c, y_c)$

AIM - aria de inelță minimă

orient =  $\angle (Ox_{vis}, AIM)$



$$G = P \cdot P^\top$$

$$X_G Y_G$$

$$C(x_c, y_c) = C \cdot M$$

$$d_{cg} = \text{dist}(C, G)$$

$$\alpha = \angle(d_{cg}, M)$$

SET part.loc = to.cam [cam] : vis.loc : grip.trans

part.loc = amplasament robot ~~before~~ relativ la SR robot  
 $(x_0 \ y_0 \ z_0)$

part.vis = amplasament robot relativ la SF imagine  
 $(x_{vis} \ y_{vis} [z_{vis}])$

Parametrii modelului de prindere (Robot-Object)

$\rightarrow d_{cq}$

$\rightarrow \alpha$

$\rightarrow z\_offset = \text{dist}(P, G)$

$\rightarrow rz\_offset = f(x_m, x_{vis})$

$d_{cq}, \alpha, z\_offset, rz\_offset = f(C, AIM, "stil prindere")$

VTRAIN.FINDER  $\rightarrow$  învăță modelul de recunoaștere  
 ↓  $\rightarrow$  învăță stilul de prindere  
 $\text{mod} = 14 \text{ sau } 5?$

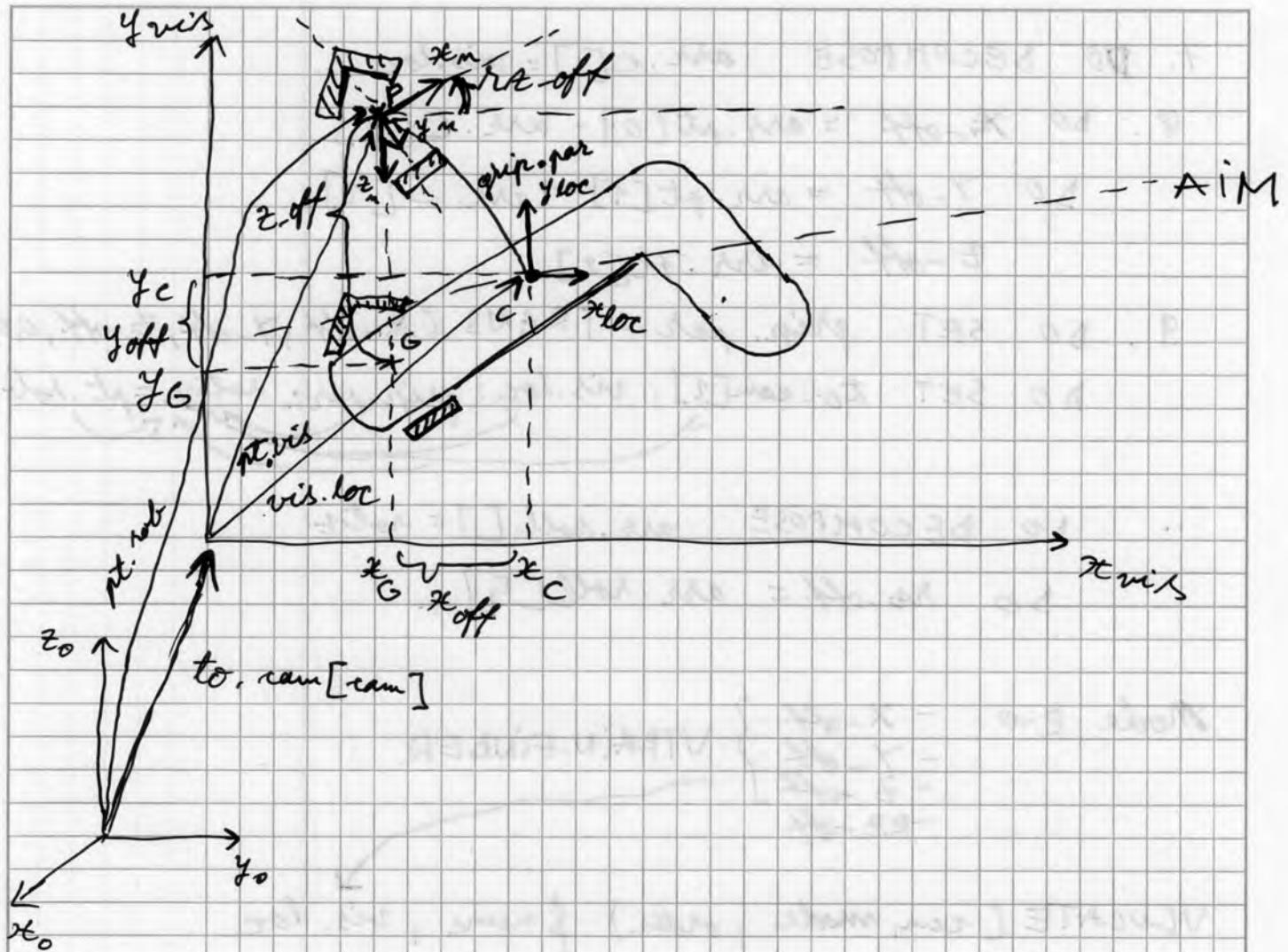
Descrierea tehnicii de învățare a parametrilor modelelor de prindere (Robot-Object)

- exemplificare pentru Adept Vision  $\leftarrow$  ObjectFinder (VTRAIN.FINDER)

Model R-O

$$\begin{array}{l} \cdot d_{cq} \\ \cdot \alpha \\ \cdot z\_off \\ \cdot rz\_off \end{array} \rightarrow \begin{cases} z\_off & = x_G - x_c \\ y\_off & = y_G - y_c \\ z\_off & = \text{dist}(P, G) \\ rz\_off & = f(x_m, x_{vis}) \end{cases}$$

output executare VTRAIN.FINDER



Secvență de învățare a parametrilor modelului R.O.

1. MCP → placare manuală a robotului în amplasamentul dorit  
de grădere

cam = 1

DO CLOSEI

DO OPENI

DO CLOSEI

2. HERE pt.rob ← înregistrează pără robot

3. DO OPENI

DO DEPARTS 100

DO MOVE outside

4. DO SET. to.cam[1]: pt.vis = pt.rob ← se află pt. vis - cu  
cam(Cx, y, z, Rx, Ry, Rz)

5. DO DECOMPOSE arr. pt[1] = pt.vis ← introduce cele 6 componente  
pt.vis în arr

6. VPICTURE(1, -1) - 1, 1 ← how many

DO VLOCATE(1, 2) "?", vis.loc [xc, yc, 0, ..., 0] ← se returnează  
în vis.loc  
amplasamentul obiectului

7. DO DECOMPOSE arr.c[] = vis.loc

8. DO X-off = arr.pt[0] - arr.c[0]

DO Y-off = arr.pt[1] - arr.c[1]

Z-off = arr.pt[2]

9. DO SET grip.par = TRANS (X-off, Y-off, Z-off, 0,0,0)

DO SET to.cam[1]: vis.loc : grip.par : rot.z = pt.rob

.. DO DECOMPOSE arr.roll[] = rot.z

DO RZ-off = arr.roll[5]

Mode R-O  
- X-off  
- Y-off  
- Z-off  
- RZ-off } VTRAIN.FINDER

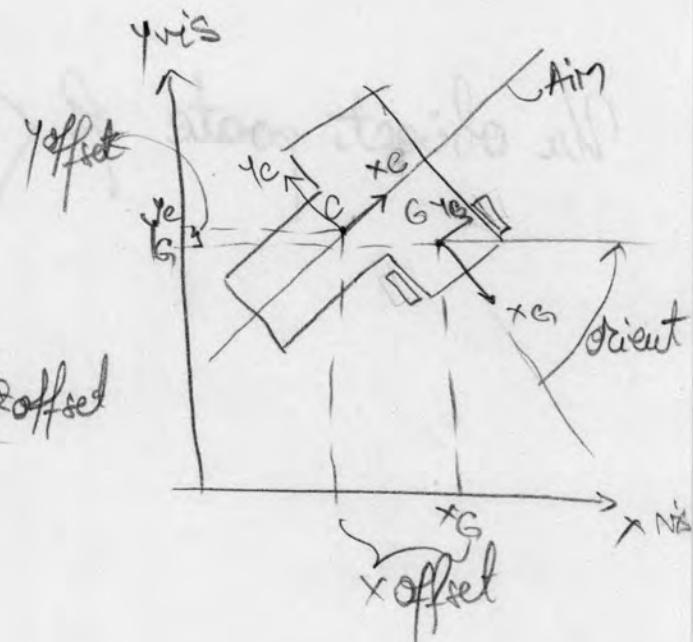
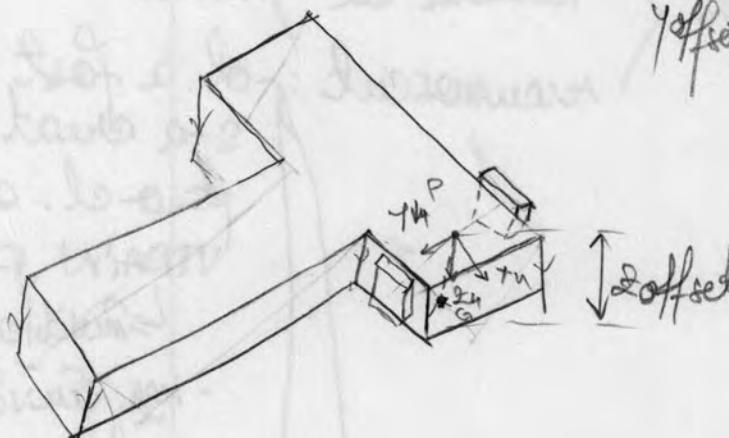
VLOCATE (cam, mode, order) \$name, vis. loc

verificata in VFEATURE (1)-data VLOCATE s-a executat cu succes

## Localizare de obiecte

VLOCATE (cam, mode, order) \$name, vis.loc.

- fct. e doar instructiune de tip. ce <sup>x<sub>obj</sub>, orient</sup> localizeaza pt. sele
- realizeaza cu succes a ac. ciestr. permite localizarea ob. în pl. imag => obt. pozitie și orientare și de control atasat robotului pe pl. imag. cf. anii stîl de studiere anterior învățate.



$x\text{-offset}$   
 $y\text{-offset}$   
 $z\text{-offset}$   
 $\{$   
 $2\text{-offset}$ 
} VTRAIN.FINDER mode = 5

vis.loc - var. de tip transf. relativă din care 3 param. sunt 0.

$\{$   
 $x_1, y_1, z_1, R_x, R_y, R_z$   
 $\uparrow \quad \uparrow$   
 $x_G \quad y_G$   
 $\}$   
 orient

? = 0 pt. redată pt. că redarea nu vede în sp.

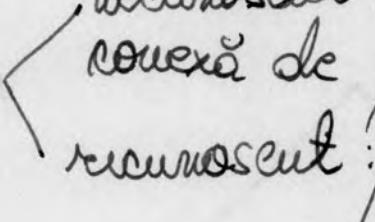
- și se va utiliza din pick-place

a) Cam - id - ul cam. virtuală pe care noi o folosim  
- p. și cam. fizică pentru def. și cam. virtuală

Dmocle = 3 biti

=  $B_2, B_1, B_0$   $\xrightarrow{\text{MSB}} B_0 = \begin{cases} 0, \text{wait} \\ 1, \text{no-wait} \end{cases}$   $\rightarrow$  dc. se impune ca să progr.  
să fie autorizat să făc.  
dc. să-a terminat locali-

$B_1 = \begin{cases} 0, \text{Final any} \\ 1, \text{Final particular} \end{cases}$   $\xrightarrow{\text{găsește (+) piesă sau}}$  sarea sau nu.  
 $\xrightarrow{\text{găsește (+) piesă, sau}}$  unele piese particiale

Un obiect poate fi   
necunoscut : blob, pante, regiune  
coperă de pixeli

recunoscut : ob. a fost învățat  
s-a creat un model  
pt. o cl. de ob.

VTRAIN, FINDER, modele  
învățare offline

- rez. învățate și  
obt. cu VFINDER

nume : - fil. de char.

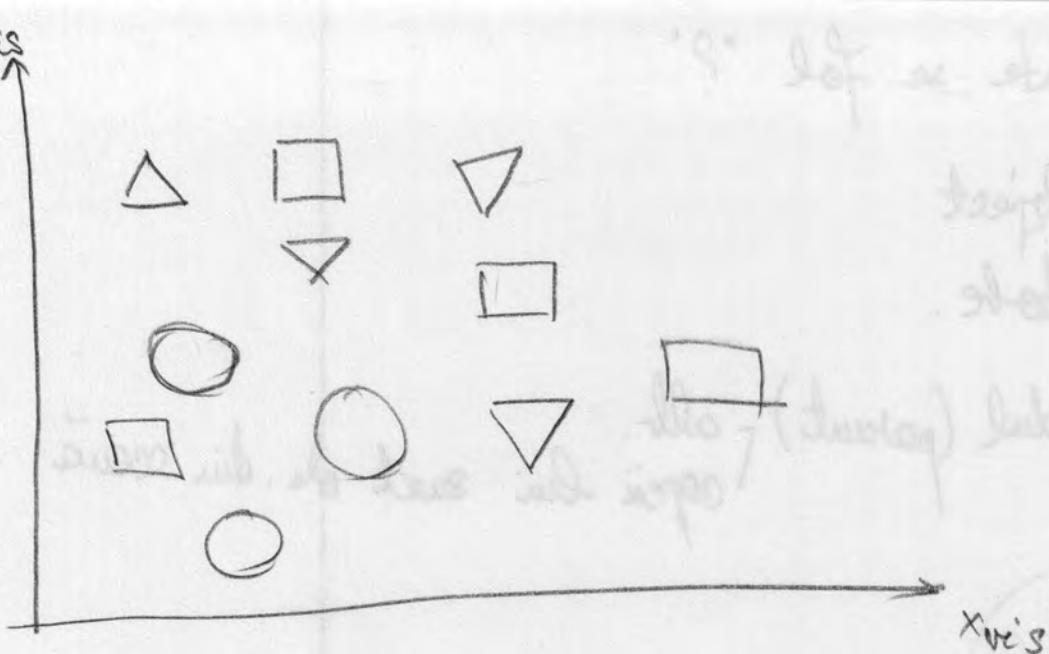
"FLANSA 4"

a fost recunoscută și insta-  
ta a ob. (un model)

O alternativă la învățarea ob. modelului este urmă-  
rea de caracteristici și discriminarea între ob.

△ - triunghi  
□ - patrat  
○ - disc

} ob. din pl. vederii  $\Rightarrow$  se doarță împărțirea pe  
conținutul de piese de ac. tip.



Find any - mod de localizare a unui ob. recunoscut sau  
rec. recunoscute care îndeplinește condi. de  
moment ORDER

Find particular - se localizează un ob. dintr-o cl., respectând  
ocultarea

⇒ implică și nume

any : (\$name . et. ce) vor. în  
cerc. și. va returna o listă de ob. loc.  
particular : "nume . et. ce" reprez.  
numele cl.

Find any → golike scene

VLOCATE( . . . ) \$name,

IF VFEATURE(1) THEN

IF \$name == "PATRAT" THEN

≡ instr. ce tratează patratul

ELSE

IF \$name == "TRIUNGHI" THEN

≡ instr. ce tratează triunghiul

ELSE

IF \$name == "DISC" THEN

≡ instr. ce tratează discul

END

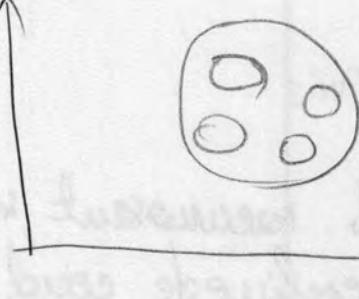
END

-f. ob. neînățăte se fol "?"

-> B<sub>2</sub> { 0, object  
1, hole.

! imp. este fondul (parent) alb.  
copiii lui sunt ob. din scenă

yvis



1+4 NLOCATE (1+ob și 4+găuri)

order - st. o ordine predefinită de localizare a ob.

0 → aici o ordine și putem să uim scriam

1 → always the biggest

2 → always the smallest

3 → always the leftmost

4 → rightmost

5 → downmost

6 → upmost

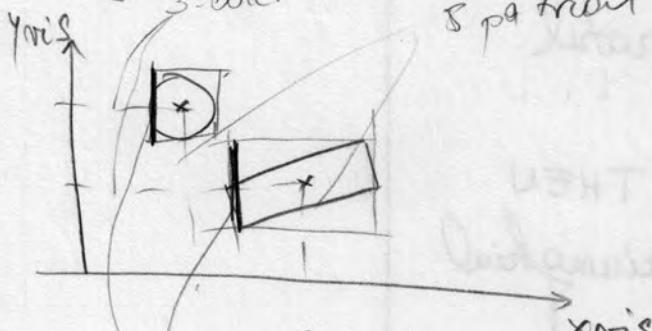
d.p. d.v. al. C(x<sub>c</sub>, y<sub>c</sub>)

7

8 → la fel ca 3÷6 d.p. d.v. al sim

9

10



de în cadrare  
Vale căreia sunt  
lucru axele x vis, y vis  
de atie min.

la 7-10 se văz. multe vîngătări

Bere)

Dură calc. off-line al param. modelului de proiecte  
(x-offset, y-offset, rez-offset, z-offset), primii 3 fiind introduse în sis. ast. când recunoașterea se face pe baza de modele învățate

VTRAIN.FINDER(cau, mode, moshe, arg, arg2, arg3)

\$model-name, ibc = val, val2, val3

- cau - nr. cau. virtuală
- mode - și moduri:  $\begin{cases} 5 & \text{pt. introducerea celor 3 modele} \\ 14 & \end{cases}$

• duode-ignorat

• arg  
arg2  
arg3  
) depend de mode

• \$model-name

• ibc = image buffer  $\rightarrow$  fie subsp. a pl. imag

• val  $\leftarrow$  x offset  
val2  $\leftarrow$  y. offset  
val3  $\leftarrow$  rez offset

f. moshe = 5

• pt. mode = 14  $\rightarrow$  val / -5, 5  $\rightarrow$  deviația AIM  
val2

## Recursoarele obiectelor

- 2 moduri - după, trăsăteri: inițiere a unei sesiuni de învățare  
a trăsăturii obiectelor.  
după model → învățare a modelului  
→ VTRAIN.FINDER mode = 3

1. Selectie trăsăturilor care identifică cel mai bn. un ob.  
(care) discrimină cel mai bn. o cl. de ob. de celelalte

$$\text{aria} = S$$

$$\text{perimetru} = P$$

$$wz. \text{găuri}$$

fact. de rotație - r

$$\rightarrow \text{pt. arie} \quad S = \pi R^2$$

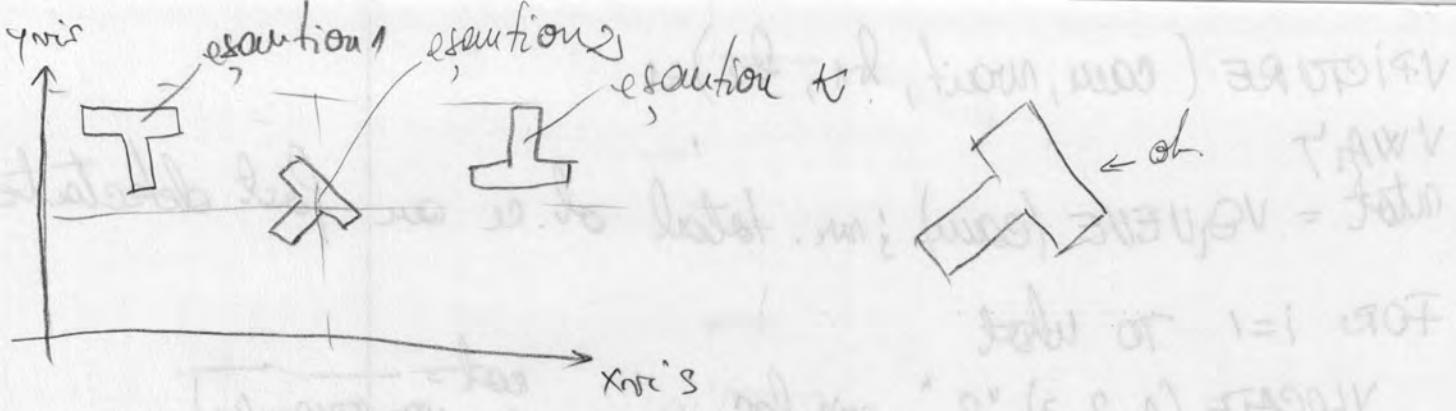
$$P = 2\pi R$$

$$r = \frac{P^2}{4\pi S}$$

Pt. fiecare reprezentant (cl)

→ cel puțin 6 ori

cauză	1	2	3	n
arie	404	390	410	408
perimetru	184	186	179	182
wz. găuri	4	4	4	4
factor de rotație	0,96	0,95	0,98	0,98



VPICTURE (cam, <sup>wait</sup>buf1, buf2) mode, 1  
-1

- | VF(10) - arcie. - in pixels
- | VF(41) - perim
- | VF(17) - arc. găuri

DO VLOCATE (1,2,3) "?", vels loc.

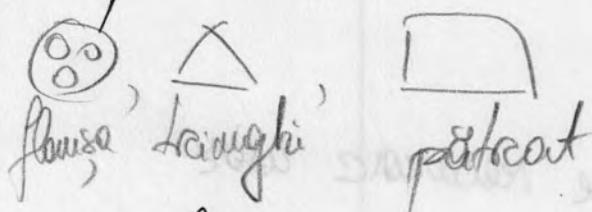
DO TYPE VFEATURE(1)

DO TYPE VFEATURE(10)

DO arcl.um = VFEATURE(10)\*Yscale\*\*2 \* XYratio

DO f.round = (VFEATURE(41)\*\*2)/(4\*pi\*arcl.um)

Γ - se reține cel mai mare val. dintre de la fiecare frâță-  
heră, precum și min



disc | în urmă model.

sterniu.  $f = 390$

arcmax.  $f = 410$

pozsterniu.  $f = 179$

pozmax.  $f = 184$ .

urq.  $f = 4$

frat min.  $f = 0,95$

frat max.  $f = 0,98$

VPICTURE (cam, wait, b<sub>1</sub>, f<sub>2</sub>) -1

VWAIT

atot = VQUEUE (cam) ; nr. total ob. ce au fost detectate

FOR i=1 TO atot

VLOCATE (1, 2, 3) "?", vis. loc.

IF VFEATURE (1) THEN

OK.f = ((ar >= ar\_min) AND (ar <= ar\_max)) AND  
(per >= per\_min) AND (per <= per\_max)) AND  
holes == 4 AND (rot\_s = rot\_min) AND (rot <= rot\_max)

IF OK.f THEN

; creeaza ob. de tip flansa

SET place

CALL pick-place ()

nr.f = nr.f + 1

ELSE

OK.f = ...

END

END

END

- tehnica se utilizeaza pt. ob. sp. ce te recurgere usor

Interactarea de modelului

Reconoscere după „model”

→ Object Finder (V+)

Etape de parcurs

1. Învățare model (creare/generare model) - offline
2. Planificare a modului în care se va efectua căutarea - offline  
(identificarea)
3. Identificare propriu-zisă obiecte (Search) - online

### ① Generare model

- constă în generarea unui model de recunoaștere pt fiecare clasă de obiecte
- model din imaginea cu mințiuri de gri
- parametrii  $\rightarrow$  ai modelului
  - $\rightarrow$  de căutare
- se face se un singur cadru / sample
- 2 etape  $\rightarrow$ 
  - 1: Detectare contururi
  - $\rightarrow$  2: Generare și refinare set de trăsături (pe baza conturilor)  $\rightarrow$  Edge features

Criterii impuse:

1. obiectul să fie reprezentativ pt clasa din care face parte (nu defecte, etc.)  
 $\rightarrow$  în zona valorilor medii
2. imaginea să fie lipsită de războaie
  - fond / fundal
  - obiecte

### 3. Fond uniform

Operații

1. Definire AOI  $\rightarrow$  VDEF. AOI  
area of interest

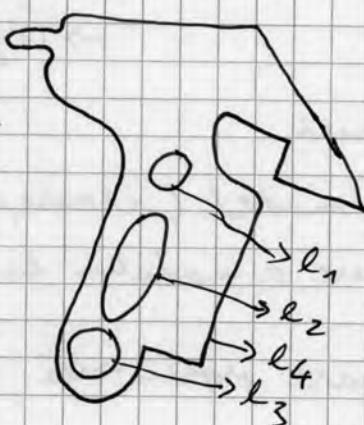
$\downarrow$   
ibr image buffer region

2. Multi-instrucție VTRAIN. FINDER



Edge features

sistemul  
elimină  
contururi false



- se rețin → contururile închise
- care au o lungime minimă imposibilă
- acele contururi care au contrast bun între foreground și background

contururi atomicice

- linii drepte
- arce de cerc

sistemul generează pt fiecare obiect o lungime  
ponderată:

$$L_{p_i} = \sum_{i=1}^n l_i \cdot w_i$$

$$|L_c - L_{p_i}| \leq E_s \cdot L_p$$

lungime contur candidat

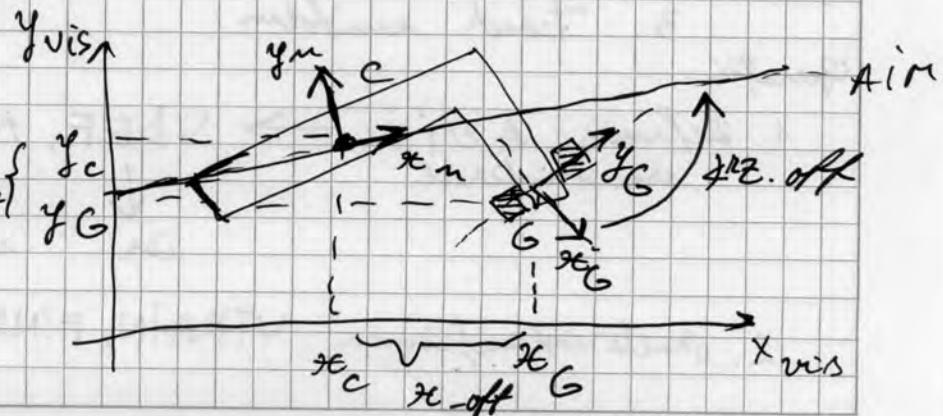
$$\bar{E} = 1 - p ; p = \text{verify percentage}$$

Etape pt verificarea concordanței

- Outline → reține un nr de contururi candidat  
→ grosieră
- Detaliu → de detaliu

- recunoaștere  
- localizare

$$\{G\} = \text{proj} \left[ \{P\} \right]_{(x_{vis}, y_{vis})}$$



VTRAIN.FINDER (cam, mode, dmode, arg1, arg2, arg3) \$ name -  
model, ibr = val, val2, val3

pt recunoastere : mode = 3  $\rightarrow$  def param de recunoastere asociati  
 $\hookrightarrow$  arg = 3 : defineste "verify percentage"

arg, arg2, arg3  $\hookrightarrow$  valori  $> 0$ ; def. param de search  
 $\rightarrow$  valori  $< 0$ ; def. param de model

ex : VTRAIN.FINDER(1, 3, ...)  $\stackrel{\text{"FLANS", ibr}}{=} 95$  ( $0 - 100\%$ )

VTRAIN.FINDER(1, 3, ...) = 8 ( $1 - 10$ )

VTRAIN.FINDER(1, 3, ...) = 0  $\{0, 1\}$

$0 \rightarrow$  lata sistemul sa decida criteriile  
pt filtrare / miciunile de filtrare  
la OUTLINE si detail

mode = 5  $\rightarrow$  pentru localizare

VTRAIN.FINDER(1, 5, ...) "L", ibr = X-off,  
Y-off, Z-off

mode = 14  $\rightarrow$  pentru restrictii de  
cantare

VTRAIN.FINDER(1, 14, ...) name = alfa\_min, alfa\_max

alfa\_min, alfa\_max =  $\neq$  min/max dintre  
A(i) si X  
 $\rightarrow$  restrictie de orientare

## 2. Planificare

VPLAN.FINDER (cam, mode, dmode, ambiguity)

\$ fmodels[], \$ bmodels[]

ambiguity → diferențe între corpurile cu detaliu → ce pot conduce la  
ambiguitate  
fmodels → foreground models

→ tabelou/listă de modele de obiecte învățate

bmodels → background models

presupunem "FLANSA", "LIG", "TT" învățate

... \$ fmodels[0] = "FLANSA"

\$ fmodels[1] = "LIG"

\$ fmodels[2] = "TT"

\$ fmodels[3] = " " → capital letter

→ când program este rulat, și obiect va fi comparat și  
match-uit cu modelele specificate

## 3. Recunoaștere (Search) → online

VFINDER (cam, mode, dmode, how-many) îl se

how-many → ~~număr~~ cete obiecte să recunoască

→ "-1" → oricât de multe

recenta este

~~VFINDER (cam, mode, b1, b2)~~

VPICTURE (cam, wait, b1, b2) mode

VFINDER (cam, mode, dmode, how-many) îl se

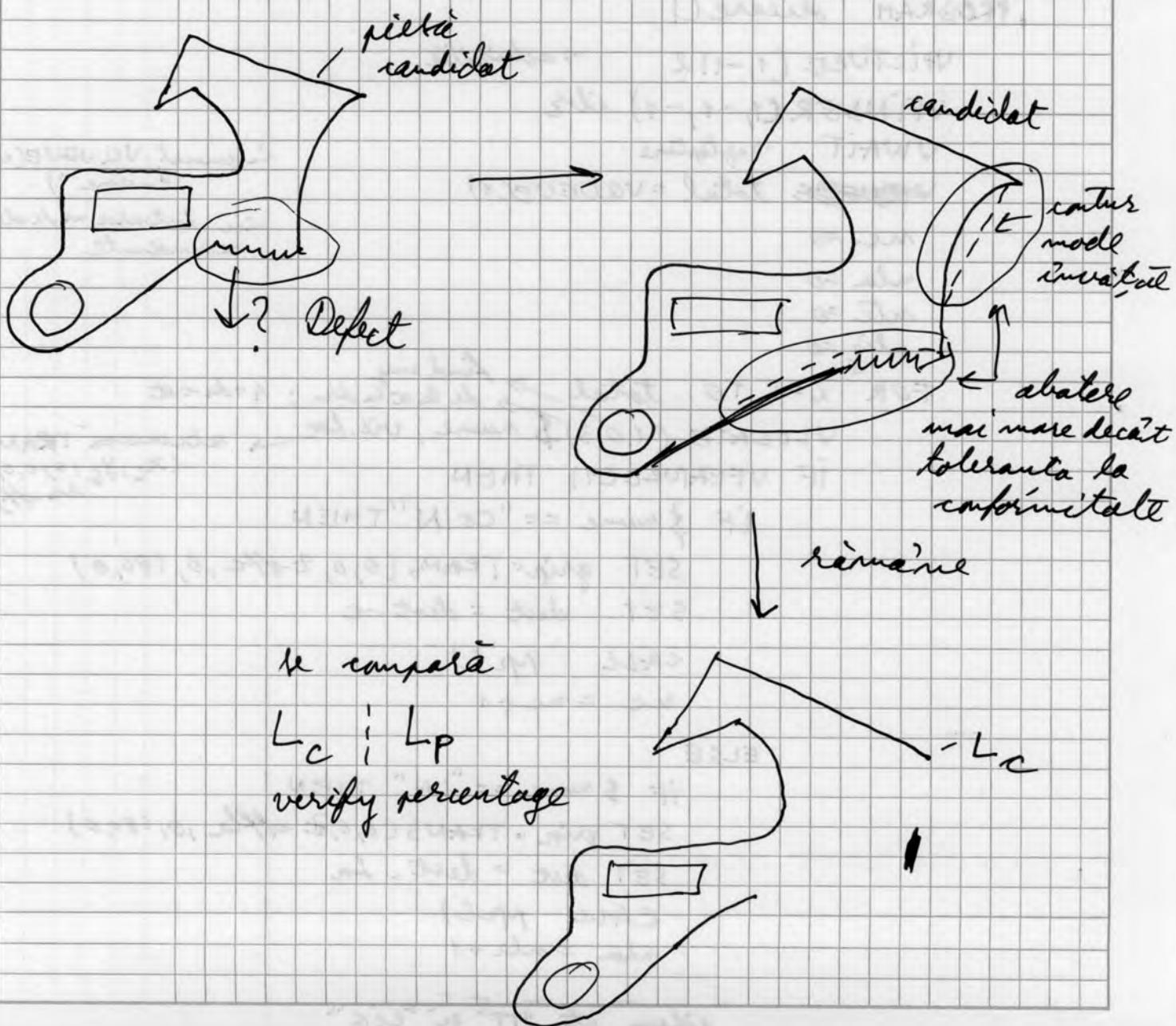
VWAIT

mode  
-1, 0, 1, 2  
2 = achiziție only

și obiecte fară modele de recunoaștere, trebuie să le da un  
nume defaut: "?"

## Parametrii pt Search

1. Contour detection parameters
  - OUTLINE (1-10)
  - DETAIL (1-10) $\text{UTRAINFINDER}(-, 3, --, -15) = 0 \rightarrow \text{automat}$
2. Contrast parameters
3. Search constraints (orientarea AIM)
  - alfa\_min
  - alfa\_max
4. Recognition level (1-10)
5. Conformity tolerance
6. Verify percentage



## Programe

1. Am învățat un nr de 4 modele: "COIN", "LA", "TE", "LIG"  
 și dorește mutarea lor la :  $\downarrow \downarrow \downarrow \downarrow$   
 pt fiecare destinație avem  $x\_off, y\_off, z\_off, rx\_off$   
 distanța de pick and place :  $d_{zpi}, d_{apl}$   
 așteptare de la comanda deplasare gripper până la  
 urm. comandă: 0,5 sec  
 viteză maximă 100% cu excepția pick, place  $\rightarrow 25\%$   
 punct tare învățat  
 se încrește cu numărul virtuală<sup>1</sup>

### PROGRAM oricare()

VFEATURE(1,-1) 2  $\rightarrow$  achiziție

VFINDER(1,-1,-1) idz

VWAIT  $\rightarrow$  așteptare

~~VLOCATE~~ total = VQUEUE(1)

nc = 0

nla = 0

nte = 0

alig = 0

FOR i=1 TO total  $\rightarrow$  find any de la start la dr ; 4=dr  $\rightarrow$  st

VLOCATE(1,0,3) \$ nume, vis.loc  $\rightarrow$  returnată TRANS

IF VFEATURE(1) THEN

IF \$nume == "COIN" THEN

SET grip = TRANS(0,0,z-offc,0,180,0)

SET dest = dest - c

CALL MRC()

nc = nc + 1

ELSE

IF \$nume == "LA" THEN

SET grip = TRANS(0,0,z-offla,0,180,0)

SET dest = dest - la

CALL MRC()

nla = nla + 1

in general VQUEUE(com, "nume")

aici întoarce nr de obiecte  
reunibile

idem pt "TE" și "LIG"

END ← pt if VFEATURE  
END ← pt far  
MOVE rate  
• END

. PROGRAM MR()

PARAMETER HAND.TIME = 0.5  
SPEED 100 ALWAYS  
SET pick = to.cam[1]: vis.loc: grip  
APPRO pick, dapi  
BREAK  
SPEED 25  
MOVES pick  
CLOSEI  
DEPART dapi  
APPRO dest, dapl  
BREAK  
SPEED 25  
MOVES dest  
OPEN i  
DEPARTS dapl  
RETURN  
• END

2. Același lucru, întră grupat pe tipurile de piele

. PROGRAM ordine ()  
VPICTURE(1,-1) 2  
VFINDER(1,-1,-1) idr  
VWAIT  
totc = VALUEVEC(1, "COIN")  
totla = VALUEVEC(1, "LA")  
totte = VALUEVEC(1, "TE")

```
catdig = VALUEUE(1, "LIG")
mc = 0
    *  
    nldig = 0
FOR i=0 TO totc → "find-particular"
    VLOCATE(1, 2, 3) "COIN", vis.loc
    IF VFEATUREC1) THEN
        SET dest = dest.s
        SET grip = TRANS(0, 0, z-off, 0, 180, 0)
        CALL nrc()
        mc = mc + 1
    END
END
--- for -uri pt relatale tipuri de piele
MOVE rate
o END
```