

Laborator 11

Pick and Place folosind vederea

As. ing. Alexandru Dumitrache
As. ing. Raluca Tudorie
www.scr.cimr.pub.ro

Introducere

You may have been studying robot vision too much if...

*...you visit the eye doctor and ask if he can increase
your viewing resolution and turn on anti-aliasing.*

Proiect AdeptSight

- Secvențe de vedere
 - Instrumente de vedere
 - * Acquire Image
 - * **Locator**
 - * Caliper, Arc Caliper
 - * Line Finder, Arc Finder
 - * Image Histogram
 - ș.a.m.d.
- System Devices
 - Cameră (emulată / fixă / montată pe brațul robotului)
 - Controller robot
 - Conveior

Calibrarea sistemului de vedere

In theory there is no difference between theory and practice, but in practice there is.

- *Calibrarea camerei*: permite trecerea de la coordonate discrete, exprimate în pixeli, la coordonate reale, exprimate în milimetri;
 - Se realizează folosind o imagine de referință (Calibration Target) care conține o grilă de puncte egal distanțate.
 - Corectează distorsiunile introduse de lentilă și de poziționarea imperfectă a camerei.
 - * Atunci când camera nu este perfect perpendiculară pe planul de lucru, apar deformări de perspectivă (liniile paralele din planul de lucru se vor vedea neperalele pe imagine).
 - * Lentila poate introduce distorsiuni neliniare, de obicei de tip butoi / pernă (Barrel / Pincushion ¹) sau combinate.

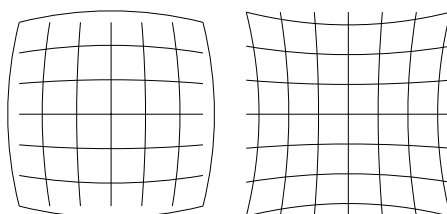


Figura 11.1: Distorsiuni de tip butoi și pernă

- Implementare: transformare matematică neliniară (*Model*), obținută prin regresie, sau Look-Up Table (*LUT*).
- *Calibrarea cameră – robot*: face corespondența dintre coordonatele din imagine (exprimate în milimetri) și coordonatele robot (exprimate de asemenea în milimetri). Este o matrice de transformare HTM 4×4 , similară cu un punct robot. Se calculează folosind un obiect de probă, care este deplasat în spațiul de lucru în poziții cunoscute de robot, iar apoi localizat pe imagine.
- *Offset-ul de prindere (Gripper Offset)*: este poziția robotului, relativă la piesa recunoscută de Locator ². Pentru un model se pot învăța mai multe offset-uri de prindere (Fig. 11.2).

¹Vezi http://en.wikipedia.org/wiki/Image_distortion

²În limbaj ceva mai tehnic, este vorba de poziția punctului condus, exprimată în sistemul de coordonate al modelului piesei (cel afișat cu galben în AdeptSight la învățarea modelului)

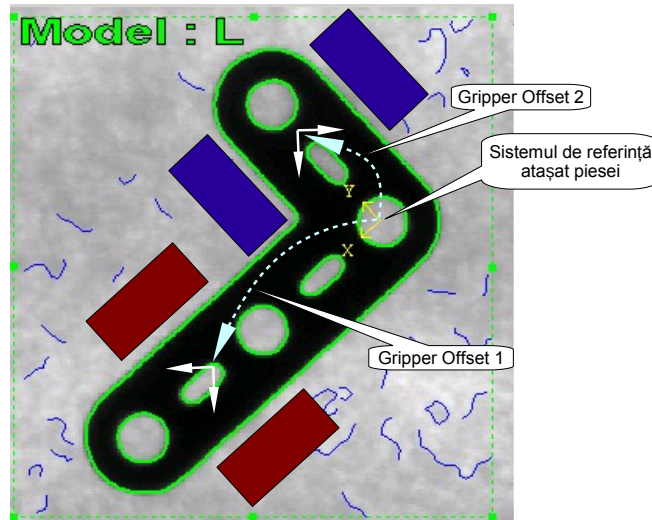


Figura 11.2: Modelul piesei de tip L cu două offset-uri de prindere

Strategii de control vizual

Never trust modern technology. Trust it only when it is old technology.

Look and Move

- Robotul se duce într-o poziție care îi permite camerei să ia o imagine (punctul #loc.img);
- Se ia o imagine, se analizează, și se calculează următoarea mișcare a robotului. În cazul cel mai simplu, se calculează poziția de prindere a unui obiect localizat în imagine.
- Robotul execută mișcarea calculată la pasul anterior, și ciclul se reia.

Visual Servoing

În acest caz, vederea artificială intervine direct în bucla de reglare, înlocuind sau completând datele furnizate de encodere și senzori. Principala problemă este extragerea din imagine a unui set de informații, într-un timp foarte scurt și având o imunitate ridicată la perturbații (robustețe). Informațiile extrase în timp real din imagini sunt folosite pentru a închide bucla. Echipamentele folosite în Visual Servoing trebuie să fie capabile să achiziționeze și să proceseze câteva zeci de imagini pe secundă.

În laborator vom folosi doar *Look and Move*.

Interfața dintre AdeptSight și V^+

If you can't understand it, it is intuitively obvious.

AdeptSight rulează pe PC, sub forma unui server TCP/IP. Programul care rulează pe robot (scris în V^+) trebuie să cunoască adresa IP a PC-ului, pentru a se putea conecta. O vom declara într-o variabilă de tip string:

```
$ip = "172.16.200.20"
```

Vom seta și parametrul VTIMEOUT la 5 secunde. Efectul: dacă serverul AdeptSight nu răspunde în 5 secunde, se generează o eroare. Valoarea implicită este 0 (așteptare la infinit).

```
PARAMETER VTIMEOUT = 5
```

Se pot realiza următoarele:

- Execuția unei secvențe: VRUN;
- Obținerea rezultatelor returnate de instrumentele de vedere: VRESULT și VLOCATION;
- Setarea parametrilor avansați: VPARAMETER.

Execuția unei secvențe de vedere

Programul V^+ cere serverului AdeptSight să execute o secvență de vedere folosind VRUN. Execuția secvenței înseamnă execuția tuturor instrumentelor definite în acea secvență (Acquire Image, Locator etc).

```
VRUN $ip, seq
```

Argumentul \$ip este adresa serverului pe care rulează AdeptSight, iar seq este numărul secvenței de vedere dorite. Secvențele sunt numerotate începând de la 1. La fel sunt numerotate instrumentele în cadrul unei secvențe.

Exemplu: se execută prima secvență de vedere:

```
VRUN $ip, 1
```

Citirea rezultatelor obținute prin analiza imaginii

When all else fails, read the instructions.

Instrumentele de vedere întorc o serie de rezultate (vizibile în fereastra de jos din AdeptSight). Acestea pot fi:

- Rezultate de tip scalar (număr întreg, real, sau variabilă booleană)
 - Se obțin în V^+ cu **VRESULT**
 - Exemple:
 - * grosimea unui obiect, întoarsă de Caliper;
 - * coordonata X a centrului unui arc identificat de Arc Finder.
- Rezultate de tip locație (transformare):
 - Se obțin în V^+ cu **VLOCATION**
 - Exemplu: punctul de prindere al unui obiect recunoscut de Locator.

Sintaxa:

```
rezult.scalar = VRESULT($ip, seq, tool, instance, result, index, frame)
SET rez.loc = VLOCATION($ip, seq, tool, instance, result, index, frame)
```

unde:

- **seq** este numărul secvenței (începând cu 1);
- **tool** este numărul instrumentului de vedere din cadrul secvenței (de la 1);
- **instance** este indexul instanței dorite:
 - Dacă instrumentul Locator găsește 3 obiecte, le vom selecta prin **instance = 1**, apoi 2 și respectiv 3.
 - Dacă instrumentul Caliper are configurate mai multe perechi de muchii (Pair0, Pair1 etc), atunci vom selecta fiecare pereche astfel:
 - * Pair0 → **instance = 1**;
 - * Pair1 → **instance = 2** etc.
- **result** este un cod care identifică rezultatul dorit (se citește din tabel: AdeptSight Online Help - Properties Reference);
- **index** este nefolosit (este tot timpul 1);
- **frame** identifică id-ul frame-ului dorit (folosit la instrumentele de vedere la care am setat Frame Input; dacă nu este cazul, îl lășăm 1).

Argumentele **index** și **frame** sunt opționale.

Exemple de rezultate care pot fi obținute ³:

- De la instrumentul *Locator*:
 - 1310: InstanceCount: numărul de obiecte găsite de Locator;
 - 1312: InstanceModel: modelul obiectului identificat de Locator (tipul piesei recunoscute), returnat ca index 0-based;
 - *1311: InstanceLocation: poziția unui obiect* ⁴;
 - *1400: InstanceLocationGripperOffsetMinimum: poziția unui obiect, cu primul offset de prindere aplicat*;
 - *1401: poziția unui obiect, cu al doilea offset de prindere aplicat*;
 - *1402: poziția unui obiect, cu al treilea offset de prindere aplicat*; ... până la 1499.
 - 1318: InstanceMatchQuality: procentul de potrivire dintre conturul modelului învățat și conturul detectat pe imagine. Instrumentul Locator returnează doar obiectele pentru MatchQuality este mai mare decât „Min Model Recognition (%)“;
 - 1317: InstanceFitQuality: eroarea medie normalizată dintre conturul modelului și conturul de pe imagine.
 - * InstanceFitQuality și InstanceMatchQuality indică gradul de asemănare dintre modelul învățat și obiectul recunoscut.
- De la instrumentele *Caliper* și *Arc Caliper*:
 - 1924: PairSize: distanța dintre cele două muchii detectate.
- De la instrumentul *Line Finder*:
 - 1820: OutputLineAngle
 - 1821: OutputLineStartPointX
 - 1822: OutputLineStartPointY
 - 1823: OutputLineEndPointX
 - 1824: OutputLineEndPointY

Exemplu: obține numărul de obiecte găsite de Locator (al doilea instrument din prima secvență de vedere):

```
nr.piese = VRESULT($ip, 1, 2, 1, 1310)
```

³Rezultatele scrise cu italic sunt de tip locație, iar cele scrise normal sunt de tip scalar.

⁴Rezultatul lui InstanceLocation (1311) este ambiguu. Dacă nu există calibrarea cameră-robot, poziția obiectului este dată în coordonate imagine. Dacă există calibrarea, dar nu este definit nici un offset de prindere, este întoarsă poziția în coordonate robot (World), însă punctul nu poate fi utilizat pentru prinderea obiectului. Dacă este definit cel puțin un offset de prindere, se întoarce poziția cu primul offset aplicat (aceiași rezultat ca 1400). Concluzie: vom folosi doar 140x pentru a obține poziții din care obiectul poate fi manipulat.

Configurarea instrumentelor de vedere din V^+

Every configuration option is a place where the program is too stupid to figure out for itself what the user really wants.

Fish (Friendly Interactive Shell) User Documentation

Setările pot fi citite/scrise folosind VPARAMETER.

Pentru setarea valorii unui parametru ⁵:

```
VPARAMETER (sequence_id, tool_id, parameter_id) $ip = value
```

Pentru citirea valorii unui parametru:

```
value = VPARAMETER ($ip, sequence_id, tool_id, parameter_id)
```

Toți parametrii sunt de tip scalar (numere reale sau întregi).

Parametri de interes pentru Locator:

- 519: MaximumInstanceCount: numărul maxim de instanțe (obiecte) întoarse de Locator;
- 530: InstanceOrdering: ordinea în care sunt returnate obiectele.
 - Câteva valori posibile:
 - * 2: hsLeftToRight
 - * 3: hsRightToLeft
 - * 4: hsTopToBottom
 - * 5: hsBottomToTop
 - * 6: hsQuality - sortare după gradul de asemănare (Quality) ⁶
- 557: MinimumModelPercentage: este „Min Model Recognition (%)“.

Exemplu: se configurează instrumentul Locator (al doilea instrument din prima secvență) astfel încât să returneze maxim trei obiecte:

```
VPARAMETER (1, 2, 519) $ip = 3
```

⁵În paranteză mai pot fi specificate încă două argumente opționale: `index_id` și `object_id`, pe care nu îi vom folosi la laborator.

⁶Obiectele sunt sortate întâi după Match Quality. Dacă există obiecte cu aceleași valori pentru Match Quality, acestea sunt sortate după Fit Quality.

Exemplu de program

Dacă constructorii ar face casele în felul în care programatorii concep programe, atunci prima ciocanitoare care ar veni ar distruge civilizația.

A doua lege a lui Weinberg [din Murphy]

În AdeptSight se definește o secvență cu instrumentele Acquire Image și Locator. Se învață modelul unei piese de tip **L** și se configurează instrumentul Locator astfel încât să nu confunde piesele **L** cu cele de tip **I** sau **T**. Se învață un offset de prindere pentru modelul învățat, folosind wizardul.

Se învață punctul robot #loc.img din care vor fi luate imaginile.

Programul robot:

```
.PROGRAM pickplace.sight()
  GLOBAL #loc.img
  GLOBAL $ip
  AUTO loc.piesa
  AUTO nr.piese

  ; Adresa serverului AdeptSight
  $ip = "172.16.200.20"
  PARAMETER VTIMEOUT = 5
  SPEED 20 ALWAYS
  MOVE #loc.img
  BREAK

  ; Executa prima secventa de vedere
  VRUN $ip, 1

  ; Citeste numarul de piese recunoscute de Locator
  ; result = 1310 => instance count
  nr.piese = VRESULT($ip, 1, 2, 1, 1310)

  FOR i = 1 TO nr.piese
    ; result = 1400 => locatia piesei, cu Gripper Offset aplicat
    SET loc.piesa = VLOCATION($ip, 1, 2, i, 1400)
    CALL pick.place(loc.piesa, loc.piesa);
  END
  MOVE #loc.img
.END
```


Verificarea amprentei gripper-ului

Logic is a systematic method of coming to the wrong conclusion with confidence.

Este posibil ca două piese să fie foarte apropiate. În acest caz, dacă robotul încearcă să prindă una din piese, el se va lovi cu gripper-ul de cealaltă. Putem detecta această situație folosind instrumentul Image Histogram.

Histograma este un grafic care arată distribuția nivelelor de gri în imagine. Luminozitatea unui pixel în AdeptSight ia valori de la 0 (negru) la 255 (alb). Se numără câți pixeli corespund cu fiecare nivel de gri, și se trasează graficul din Fig. 11.3 ⁷. Pe histogramă putem vedea dacă imaginea este *subexpusă* (prea întunecată, valori ridicate în partea stângă), *supraexpusă* (prea luminoasă), sau dacă contrastul este prea mare sau prea mic.

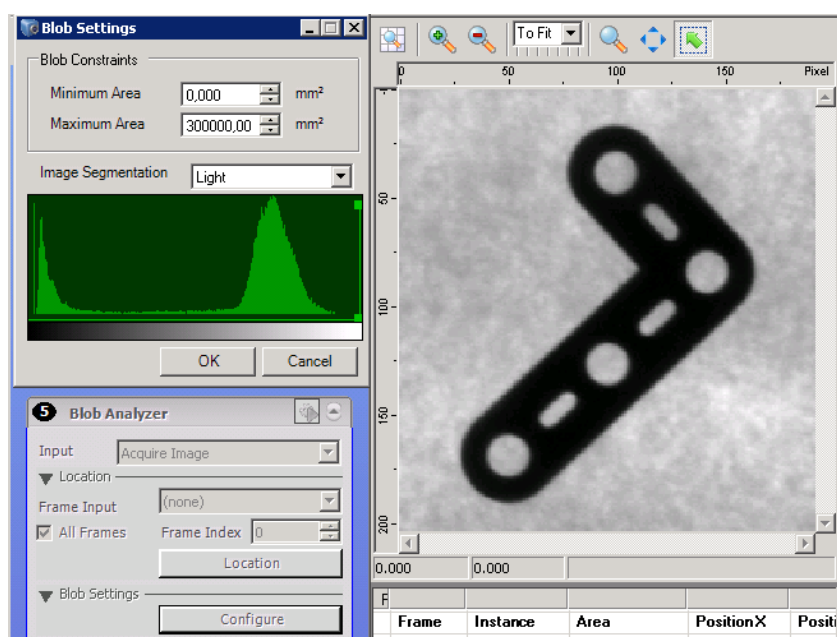


Figura 11.3: Exemplu de histogramă (fereastra de configurare Blob Analyzer)

⁷Asta se întâmplă când inginerii scriu programe: instrumentul Image Histogram nu știe să afișeze grafic histograma imaginii; în schimb, Blob Analyzer știe ☺

Vom poziționa câte un instrument Image Histogram pentru fiecare deget al gripper-ului (Fig. 11.4); acesta va acoperi cursa completă a degetului respectiv (adică, dacă un deget se deplasează 10 milimetri la închidere/deschidere, iar lățimea lui este de 15 milimetri, locația inspectată de Image Histogram va avea 25 mm). În căsuța *Frame Input* se va selecta *Locator*.

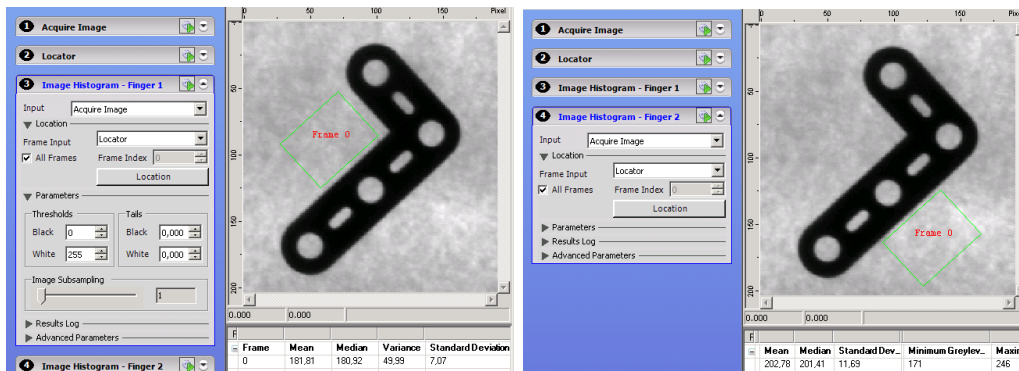


Figura 11.4: Poziționarea celor două instrumente Image Histogram

Image Histogram va calcula următorii indicatori statistici, care ne pot ajuta să aflăm dacă degetul va lovi într-o piesă sau nu:

- 1500: Mean
- 1501: Median
- 1503: StandardDeviation
- 1506: MinimumGreylevelValue
- 1507: MaximumGreylevelValue
- 1508: GreylevelRange

Pe imaginea din Fig. 11.4 putem observa că, atunci când amprenta degetului se află pe fundal, indicatorul Mean ia valori relativ ridicate (culoare deschisă, valori mai mari decât 150), iar StandardDeviation are valori mici (sub 15). Dacă amprenta se află pe o porțiune închisă la culoare, atât Mean cât și StandardDeviation au valori mici. Dacă amprenta se află parțial pe o piesă, parțial pe fundal, StandardDeviation va avea valori mari. În acest caz, validarea primei amprente pentru piesa cu numărul *i* se poate face astfel:

```

mean = VRESULT($ip, 1, 3, 1, 1500, 1, i)
stdev = VRESULT($ip, 1, 3, 1, 1503, 1, i)
valid = (mean > 150) AND (stdev < 15)
IF valid THEN
    ...
END

```

În laboratorul 8 (AnyFeeder) a fost folosită o soluție ușor diferită. Imaginea a fost reglată astfel încât fundalul (luminos) să fie văzut alb, iar piesele – gri sau negre (imagine supraexpusă și contrast ridicat). Testul pentru validarea amprenteii a fost simplificat, fiind necesară doar verificarea valorii minime a nivelului de gri prezent în regiunea de interes. Dacă această valoare era mai mică decât 255, însemna că în regiunea verificată exista cel puțin un pixel gri, adică amprenta era suprapusă pe una din piesele de pe masă, deci poziția de prindere era declarată invalidă.

Secvența de cod din laboratorul 8:

```
...
amprenta1 = VRESULT($ip, 1, 3, 1, 1506, 1, i)
amprenta2 = VRESULT($ip, 1, 4, 1, 1506, 1, i)
IF ((amprenta1 < 255) OR (amprenta2 < 255)) THEN
    blocate = TRUE
ELSE
    ; poziția de prindere OK, se execută Pick and Place
    ...
END
```

Problemă propusă

Stivuire

Se dorește recunoașterea pieselor de tip **I** folosind camera fixă, și așezarea acestora într-o stivă verticală. Piesele vor fi preluate din scena de lucru de la stânga la dreapta. Observații:

- Poziția „safe“ coincide cu #loc.img;
- Înălțimea unei piese este $h = 4.25 \text{ mm}$;
- Axele stivei verticale au dimensiunea de 30 mm;
- Viteza maximă de lucru va fi de 30%;
- Viteza de deplasare pe verticală a robotului va fi de 15%;
- Gripper-ul are nevoie de 0.5 secunde pentru închiderea respectiv deschiderea completă;
- Aplicația începe și se termină în #loc.img cu gripper-ul deschis.
- După terminare, robotul va activa semnalul 0001 și îl va menține activ timp de o secundă.

Baftă în sesiune! ☺