

C6

to.cam = transformare relativa care descrie pozitia si orientarea (sistemul de coordonate atasat planului imaginii fata de SC atasat RI); leaga lumea imaginii, exprimata in pixeli, de lumea robotului, exprimata in mm;

vis.loc = amplasamentul atasat obiectului(piesa localizata in scena), este SC atasat in centru de masa; calculul in timp real;

grip.tras = amplasamentul SC din varful robotului, in functie de unde se afla piesa; stilul de prindere impus; leaga SC din varful robotului de cel din baza;

piesa.loc = reprezinta amplasamentul din varful robotului fata de pct fix(piesa); compunerea celor 3;

look and move = folosita in peste 90% din aplicatii din cauza simplitatii;

Imagine analogica = reprezinta o functie $f(x,y) > 0$ de intensitatea luminoasa unde x, y sunt coordonate spatiale ale pixelului(pct in care analizam intensitatea);

Imagina digitala = $f^*(x,y)$ se obtine prin esantionare spatiala si cuantificare a semnalului spatial; binarizare :

```
{  
    prag de binarizare = default = 63, se foloseste pentru a obtine max de info din img analizata;  
}
```

Principiul RVA(Robot cu Vedere Artif) => 3 etape ce asigura comanda miscarii robotului:

1. Planificarea(cu ajutorul VA)
2. Generare traiectorie
3. Urmare traiectorie

Metode de tip RVA:

1. look and move => majoritatea aplicatiilor
se bazeaza pe VA in planif. miscarii
foloseste PI(prelucr de imag.)
detecteaza, recunoaste si localizeaza piesele
2. visual servoing => aplicatii speciale
se bazeaza pe reactie de imagini(extrage trasaturi)

C7

=====

EFM = interactiunea dintre 2 comp principale(comanda miscarii si planificarea vizuala a miscarii) => look and move;

Camera fizica = obiectul fizic, traductor;

Camera virtuala = concept informational(set de date); fiecare camera fizica poate avea cate 2 camere virtuale; La adept avem 4 fizice cu 8 virtuale;

Iluminare = functionare buna este determinata de un sistem de iluminat in proportie de 65%;
luminarea trebuie sa fie invarianta, de aceea trebuie izolata camera la lumina naturala;
luminarea trebuie sa fie uniforma, adica jaluzele trase si cat mai multe neoane aprinse;
luminarea poate fi frontala(gen neoane) sau in diascopie(lumina vine de jos, gen piesele sunt puse pe sursa de lumina);

Metoda de recunoastere explicita = bazata pe valorile numerice ale unor trasaturi definit pentru clase de obiecte(gen o trasatura poate fi numarul de gauri);

Metoda de recunoastere implicita = bazata pe invatarea unui model(VPICTURE);

DUALitatea = ma folosesc atat de camera fizica cat si de camera virtuala pt a imbogati informatia

Componente CV =>

1. switches(comutatoare enable/disable)
2. parameters
3. CF + modele(se vad prin nsite param invatati)
4. VQUEUE(coade de info)

1 si 2 au carac offline

3 se foloseste pt elarning

4. det o informatie creata in timpul exec

Modele de invatare => 1. Model Calibrare Camera

-2 tipuri de param intrinseci si extrinseci

2. Model Scena Obiect

-recunoastem modele de obiecte care apartin unor clase

C8

=====

Modalitati de a folosi VA:

{

1. Look and move(facem o poza a scenei si miscam RI)

2. Visual Servoing(procesarea imaginii intra in bulca de reglare)

}

Sisteme de vedere => 1. AdeptVision(placa de pe controllerul robot)

2. Adept Sight(controllerul RI se conecteaza la un server)

Pasi de lucru AdeptSight

1. ajustare aparam fizici camera(focalizare, etc.)

2. calibrare camera(px->mm, cu foaia de tinte de la lab)

3. calibrare camera robot(virtuala)

4. invatam modele de piese(ne folosim de poza piesei si ii atasam un SC)

5. invatam offset de prindere(aflam grip.trans)

Semnale

intrare: 1001-1512

iesire: 1-512

software: 2001-2512

C9

=====

Camera virtuala:

{

1. Switches

2. Parameters

3. Modele

4. VQUEUE

}

Modele:

{

1. Model camera - robot aka. calibrare -- obligatoriu!!;

2. Model scena - robot aka. recunoastere de obiecte;

{

explicita = bazata pe parametrii(arie, perimetru, nr de gauri, etc);

implicita = bazata pe modele ale claselor de obiecte;

}

3. Model robot - obiect aka. de prindere(exprima amplasamentul unui SC);

4. Modelul gripper - scena aka. evitare coliziuni;

}

VQUEUE = zona de memorie in care se depune informatia despre toate obiectele detectate si eventual recunoscute;

Histograma = contine nivelurile de gri calculate;

Detectare coliziune: -se gen modelul care elaga amprentele griperului de obiect

- daca nu exista riscuri de coliz. nu invatam modelul
- daca avem coliziune invatam modelul de prindere(mai multe modele robot-

obj)

C10

=====

Metoda de recunoasterea a obiectelor explicit:

```
{
    1. Se selecteaza setul de trasaturi care face discriminarea intre clase;
    2. Se invata valorile numerice ale trasaturilor pentru fiecare clasa;
    3. Se retin valorile min si max pentru trasaturi;
}
```

Metoda de recunoastere a obiectelor:

```
{
    1. Training = invatare de modele pentru fiecare clasa; etapa se face offline printr-o comanda monitor;
    2. Planning = planificarea recunoasterii; offline pentru fiecare model fara schimbarea iluminarii;
    3. Search = recunoasterea; in timp real;
}
```

Conditii pt invatarea corecta a obiectelor:

- obj selectat trebuie sa repr cat mai corect clasa
- se da un nume clasei(un string)
- fondul trebuie sa fie uniform
- imag. luata sa aibe un contrast bun
- calibrarea robot-camera sa fie facuta altfel nu putem face recunoasterea
- iluminare uniforma

Parametrii cautati pt cautare(search):

- nivelul de detectare al domeniilor
- contrastul
- recognition level(1-10)
- param. Conformity
- procentaj de verificare
- restrictii de cautare(determinante de amplasament)

C11

=====

Cauze pt nerecunoasterea obiectelor:

- nu s-a invatat modelul cum trebuie
- ob. nu se vad complet
- ob. se suprapun(solutie -> activam switch-ul care trebuie)
- ob. se ating(solutie -> activam switch-ul care trebuie)

Moduri de detectie:

- coarse(imagini neclare)
- detail(imagini clare)

