Sisteme de recunoastere a formelor – Laborator 2

RANSAC – potrivirea unei linii la un set de puncte

1. Obiective

Scopul acestei lucrari de laborator este de a introduce o metodă de potrivire robustă la erori mari denumită RANSAC.

2. Fundamente teoretice

Random Sample Consensus (RANSAC) - consensul eşantioanelor aleatoare - este o paradigmă pentru potrivirea unui model la date experimentale, întrodusă de Martin A. Fischler şi Robert C. Bolles în 1981 [1].

Dupa cum au declarat Fischler şi Bolles "Procedura RANSAC este opusul tehnicilor convenţionale de filtrare: în loc să folosim cât mai multe date posibile pentru a obţine o soluţie iniţială, şi apoi să eliminăm punctele invalide, RANSAC foloseşte un set iniţial de date cât mai mic posibil şi apoi măreşte acest set cu date valide atunci când este posibil."

Algoritmul RANSAC este descris în cele ce urmează [2]:

Obiectiv: Potrivirea robusta a unui model la un set de date S care conţine puncte zgomot.

Algoritm:

1. Selecţie aleatoare a unui eşantion de puncte s din mulţimea S şi instanţiere a modelului din acest eşantion.

2. Determinarea mulţimii de date S_i care conţine puncte situate la o distanţă mai mică decât un prag t de model. Mulţimea S_i este mulţimea de consens a eşantionului, şi defineste punctele din S care satisfac modelul.

3. Dacă dimensiunea lui S_i (numărul de puncte care satisfac modelul) este mai mare decât un prag T, algoritmul se termină. Opţional, se poate estima din nou modelul folosind toate punctele din S_i.

4. Dacă dimensiunea lui S_i este mai mică decât T, se selectează un nou subset şi se repetă paşii anteriori.

5. După N încercări, se selectează mulţimea de consens S_i cu cele mai multe puncte, şi (opţional) modelul este reestimat folosind toate punctele din această mulţime.

2.1. RANSAC pentru linii

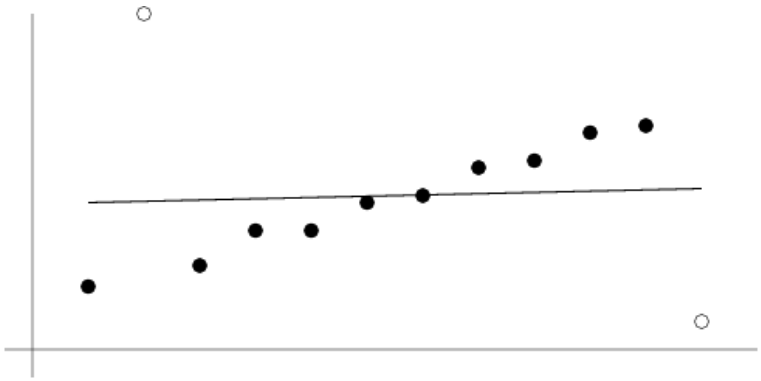

Figura 1-a

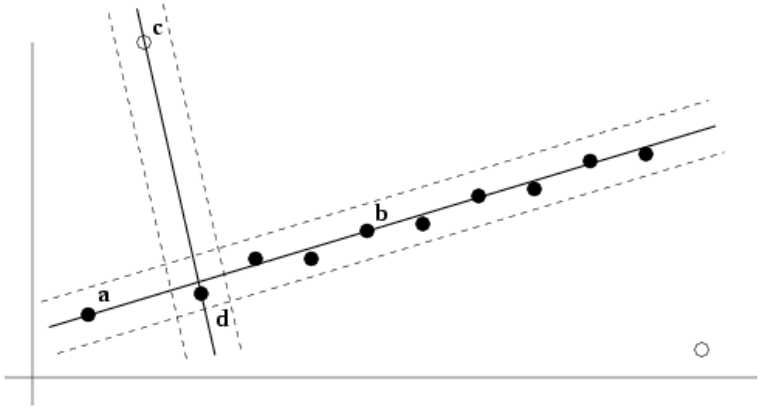

Figura 1-b

Problema, ilustrată în figura 1-a, este următoarea: fiind dată o mulţime de puncte 2D, găsiţi linia care minimizează suma distanţelor perpendiculare (regresie ortogonală), în condiţiile în care nici unul din punctele valide nu deviază de la linie cu mai mult de *t* unităţi. Există de fapt două probleme: potrivirea unei linii la date, şi clasificarea datelor în puncte valide şi puncte zgomot. Pragul *t* este selectat în funcţie de zgomotul propriu măsurătorii.

Primul pas este selectarea a două puncte aleator, iar aceste puncte vor defini o linie. Mulţimea suport (consens) pentru această linie este dată de punctele care sunt la mai puţin de o distanţă prag de această linie. Selecţia aleatoare este repetată de mai multe ori, şi linia care are cele mai multe puncte suport este considerată potrivirea cea mai robustă. Intuitiv, dacă unul din cele două puncte alese pentru potrivirea liniei este parte a zgomotului, linia estimată nu va avea o mulţime suport foarte mare.

Dacă măsurăm calitatea unei linii prin mărimea mulţimii support, avem avantajul de a favoriza cele mai bune linii de la început. Astfel, linia (a,b) din figura 1-b are o mărime a suportului de 10, pe când linia (c,d) are un suport de doar 2. Putem deduce că *c* sau *d* este un punct zgomot.

Parametrii algoritmului sunt:

1. **Pragul de distanţă:** Se alege pragul *t* astfel încât un punct este valid cu o probabilitate dată. Pentru aceasta avem nevoie de forma distribuţiei de probabilitate pentru distanţa unui punct valid la model. În practică acest prag este ales empiric.

2. **Numărul de eşantioane:** Numărul de eşantioane *N* este ales pentru a asigura, cu o probabilitate *p*, că cel puţin unul din eşantioane nu conţine zgomot. De obicei *p* este ales ca 0.99. Fie *q* probabilitatea ca un punct selectat să fie valid. Atunci probabilitatea că toate cele *s* puncte selectate să fie valide este q^s. Evenimentul complementar este că cel putin un punct să fie invalid şi are probabilitatea $1-q^s$. Dacă facem *N* selecţii probabilitatea ca să avem cel puţin un outlier în fiecare este $(1 - q^s)^N$ care trebuie să fie egala cu *1-p,* putem exprima $N = log(1 - p)/log(1 - q^s)$.

3. **Mărimea pragului pentru mulţimea consens:** O regulă simplă este că algoritmul să se finalizeze în momentul în care dimensiunea mulţimii consens este similară cu numărul de puncte valide presupuse a fi în setul de date, dându-se proporţia dintre puncte valide şi zgomot. De exemplu, pentru *n* puncte $T=qn$. Pentru potrivirea liniei din Figura 1, o estimare conservatoare este $q = 0.8$, deci $T = 0.8 \cdot 12 = 9.6$.

3. Fundamente matematice

Ecuaţia unei linii care trece prin două puncte distincte (x_1, y_1) şi (x_2,y_2) este dată de:

$$(y_1 - y_2)X + (x_2 - x_1)Y + x_1y_2 - x_2y_1 = 0$$

Distanţa unui punct (x_0, y_0) la o dreaptă definită de aX+bY+c = 0 este:

$$d = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}}$$

4. Consideraţii practice

Deschiderea unei poze cu nivele de gri:

```
Mat img = imread("filename", CV_LOAD_IMAGE_GRAYSCALE);
```

Crearea unei imagini cu nivele de gri:

```
Mat dst(height, width, CV_8UC1); //8bit unsigned 1 channel
```

Accesarea pixelului de la rândul *i* coloana *j*:

```
uchar pixel = img.at<uchar>(i,j); //unsigned char type
```

Un punct negru la poziţia `(i,j)` din imagine corespunde la un punct cu coordonatele `x=j, y=i`:

```
if (img.at<uchar>(i,j)==0){
     Point p; p.x = j; p.y = i;
}
```

Modificarea pixelului de pe rândul *i* coloana *j*:

```
img.at<uchar>(i,j) = 255; //white
```

Desenarea unei linii care trece prin două puncte *(x1,y1)* şi *(x2,y2)*:

```
line(img, Point(x1, y1), Point(x2, y2), Scalar(B,G,R));
```

Afisarea imaginii:

```
imshow("title", img);
waitKey();
```

5. Activitate practica

1. Construiţi lista de puncte prin găsirea tuturor punctelor negre din imaginea de intrare.
2. Calculaţi parametrii necesari *N* şi *T* pornind de la valorile *t=10, p=0.99, q=0.8, s=2*. Pentru *points1.bmp* folosiţi *q=0.3*.
3. Aplicaţi algoritmul RANSAC:
    a. Alegeţi două puncte diferite;
    b. Determinaţi ecuatia dreptei care trece prin cele două puncte alese;
    c. Calculaţi distanţa tuturor punctelor la dreaptă;
    d. Număraţi punctele valide (inliers);
    e. Memoraţi parametrii liniei *(a,b,c)* dacă linia curentă are cele mai multe puncte valide până acum;
    f. Stabiliţi condiţiile de oprire (mulţime de consens suficient de mare sau număr maxim de iteraţii).
4. Desenaţi linia optimală găsită de metodă.

6. Bibliografie

[1] Robert C. Bolles, Martin A. Fischler: *A RANSAC-Based Approach to Model Fitting and Its Application to Finding Cylinders în Range Data*, 1981

[2] Richard Hartley, Andrew Zisserman: *Multiple View Geometry în Computer Vision,* 2003