

4 Transformata distanță (TD). Potrivirea modelelor folosind TD

4.1 Obiective

În acest laborator vom studia un algoritm care calculează transformata distanță a unei imagini binare (obiect și fundal). Această transformare permite evaluarea unui cost de potrivire a modelului unui obiect cunoscut (de exemplu un contur de pieton) și un obiect necunoscut, pentru a decide dacă obiectul necunoscut este sau nu similar cu obiectul model. Cu cât costul de potrivire este mai mic, cu atât obiectul necunoscut este mai asemănător modelului.

4.2 Fundamente teoretice

4.2.1 Transformata distanță

Transformata distanță, cunoscută și ca hartă de distanțe sau câmp de distanțe, este o reprezentare a unei imagini digitale. Harta asociază fiecărui pixel din imagine informația privitoare la distanța sa față de cel mai apropiat pixel obiect (sau obstacol). În practică, cel mai des întâlnit tip de pixel obiect este un punct de muchie.

Transformata distanță este un operator care se aplică în mod normal doar imaginilor binare. Rezultatul transformării este o imagine cu niveluri de gri, care seamănă cu imaginea intrare, dar în care intensitățile punctelor arată distanța față de cel mai apropiat punct obiect.

În imaginea următoare avem un exemplu de aplicare a transformatei distanță pe o imagine ce conține o formă dreptunghiulară. În imaginea din stânga, pixelii cu valoarea “0” sunt pixeli obiect (puncte de muchie), iar cei cu valoarea “1” puncte de fundal. În imaginea din dreapta, se vede rezultatul aplicării TD folosind metrica “tabla de șah”, unde fiecare valoare codifică distanța față de un punct de muchie.

0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	0	0	0	0	0	0
Imaginea de intrare binară						

0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	2	2	2	1	0
0	1	2	3	2	1	0
0	1	2	2	2	1	0
0	1	1	1	1	1	0
0	0	0	0	0	0	0
Transformata distanță						

Figura 4.1 – stânga: imaginea binară de intrare; dreapta: transformata distanță a imaginii; fiecare poziție conține distanța “tabla de șah” către cel mai apropiat punct de contur (valorile 0). [1]

De obicei transformata/harta este denumită pe baza metricii alese. De exemplu, se poate vorbi de Transformata Distanță Manhattan, dacă metrica folosită este distanța Manhattan.

Alte metrice sunt:

- Distanța Euclidiană:

1.41	1.0	1.41
1.0	0.0	1.0
1.41	1.0	1.41

- Distanța *City block* sau *Manhattan*:

2	1	2
1	0	1
2	1	2

- Distanța „tablă de șah”

1	1	1
1	0	1
1	1	1

Există mai mulți algoritmi pentru implementarea TD:

- Chamfer TD;
- Euclidian TD;
- Voronoi diagram TD.

Cea mai directă metodă de calculare a transformatei distanță este metoda *brute force*. Aceasta presupune calcularea pentru fiecare pixel de fundal distanțelor către fiecare pixel de obiect. Apoi se ia minimumul dintre aceste distanțe pentru fiecare pixel de fundal. Metoda este foarte ineficientă, prin urmare vom studia o soluție mai eficientă, transformata distanță Chamfer, care aproximează distanța Euclidiană, este o metodă simplă și foarte rapidă, necesitând doar două parcurgeri ale imaginii binare.

Algoritmul Transformata Distanță Chamfer

1. Se alege o mască de ponderi 3x3 care are valori proporționale cu distanțele Euclidiene față de elementul din mijloc. Cele mai simple și mai mici astfel de valori sunt 2 pentru deplasarea laterală și 3 pentru deplasarea diagonală. În acest fel distanțele obținute vor fi egale cu aproximativ dublul distanțelor euclidiene reale.

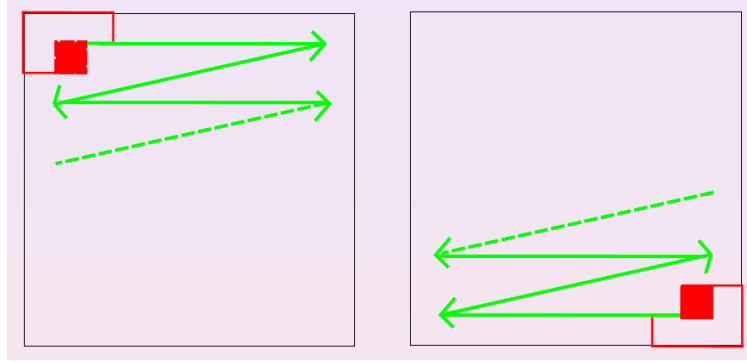
$$\text{ponderi} = \begin{bmatrix} 3 & 2 & 3 \\ 2 & 0 & 2 \\ 3 & 2 & 3 \end{bmatrix}$$

2. Transformata distanță are aceeași dimensiune ca imaginea de intrare și se inițializează cu valori de zero și valori infinite:

$$DT(i, j) = \begin{cases} 0, & (i, j) \in \text{obiect} \\ \infty, & (i, j) \notin \text{obiect} \end{cases}$$

Dacă aplicăm TD pe o imagine binară, unde valoarea 0 înseamnă pixeli obiect și valoarea 255 codifică fundalul și dorim să obținem o imagine TD în tonuri de gri (8 biți/pixel), valoarea ∞ din algoritm ar trebui înlocuită cu valoarea 255.

3. O dublă parcurgere (prima dată sus-jos, stânga-dreapta, a doua oară jos-sus, dreapta-stânga) a imaginii (cu jumătățile corespunzătoare ale măștii, vezi figura de mai jos) este necesară pentru a actualiza distanța minimală. La prima traversare elementul central este comparat cu elementele **galbene** corespunzătoare vecine, iar la a doua traversare, cu elemente **verzi**:



În timpul parcurgerii imaginii sursă (direct și invers), se face următoarea actualizare pe imaginea TD:

$$TD(i, j) = \min_{(k, l) \in Mask} (TD(i + k, j + l) + ponderi(k, l))$$

4.2.2 Potrivirea de modele folosind TD

Dorim să calculăm un cost de similitudine între un model de obiect cunoscut și un obiect necunoscut. Considerăm că ambele obiecte au aceeași dimensiune.

Pașii pentru calcularea costului sunt:

1. Calcularea imaginii TD pentru modelul de obiect cunoscut;
2. [Opțional] Suprapunerea obiectului necunoscut peste imaginea TD calculată anterior prin translatarea obiectului necunoscut astfel încât centrul său de masă să fie același cu centrul de masă al obiectului model. Pentru simplitate, centrele de masă se estimează din punctele de pe contur. Centrul de masă (C_x, C_y) al obiectului descris de conturul Ω se poate aproxima după ecuația:

$$(C_x, C_y) = \left(\frac{1}{N} \sum_{p \in \Omega} p_x, \frac{1}{N} \sum_{p \in \Omega} p_y \right),$$

unde (p_x, p_y) reprezintă coordonatele x și y ale punctului de contur p , iar N este lungimea conturului.

3. Costul de similitudine (potrivire) este media tuturor valorilor pixelilor din imaginea TD care se află sub punctele de pe conturul obiectului necunoscut. Un cost mic de similitudine arată că obiectele sunt similare.



Figura 4.2 – De la stânga la dreapta: conturul obiectului model (frunză); conturul obiectului necunoscut (pieton); pietonul este suprapus peste transformata distanță a frunzei pentru a calcula costul de similitudine.

4.3 Detalii de implementare

Citirea imaginii de intrare ca și o imagine cu niveluri de gri:

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

Inițializarea imaginii TD cu imaginea de intrare:

```
Mat td = src.clone();
```

Accesarea pixelilor dintr-o vecinătate de tip 8:

```
int di[8] = {-1,-1,-1,0,0,1,1,1};
int dj[8] = {-1,0,1,-1,1,-1,0,1};
int weight[8] = {3,2,3,2,2,3,2,3};
```

```
for(int k=0; k<8; k++)
    uchar pixel = img.at<uchar>(i+di[k], j+dj[k]);
```

4.4 Activitate practică

1. Implementați Transformata Distanță Chamfer. Calculați și vizualizați imaginea TD pentru imaginile de intrare: *contour1.bmp*, *contour2.bmp*, *contour3.bmp*. Rezultatele trebuie să coincidă cu cele prezentate în text. Pixelii de obiect sunt negri iar fundalul este alb.
2. Calculați imaginea TD pentru *template.bmp*. Evaluați costul de similitudine între imaginea model și cele două obiecte necunoscute: *unknown_object1.bmp* – pieton, *unknown_object2.bmp* – frunză. Costul de similitudine este media valorilor din imaginea TD de pe pozițiile punctelor de contur ale obiectului necunoscut.
3. Calculați costul de similitudine prin inversarea rolurilor de obiect necunoscut și obiect model.
4. Translați obiectul necunoscut astfel încât centrul său să corespundă cu centrul obiectului model și recalculați costurile de potrivire.
5. Opțional, implementați Transformata Distanță Euclidiană adevărată. De ce este diferită față de Transformata Distanță Chamfer?

4.5 Exemple de rezultate

Exemple de rezultate de imagine TD folosind metoda Chamfer și matricea de ponderi sugerată.

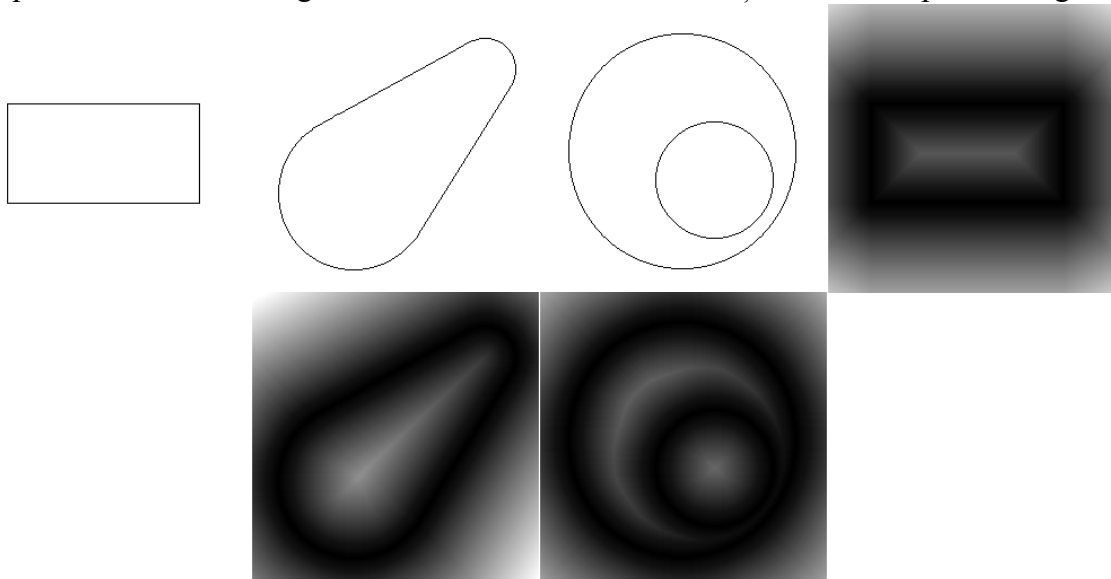


Figura 4.3 – Rândul de sus: imagini binare; rândul de jos: transformata distanță Chamfer

4.5 Referințe

- [1] Wikipedia The Free Encyclopedia – *Distance Transform*,
http://en.wikipedia.org/wiki/Distance_transform
- [2] Compendium of Computer Vision – *Distance Transform*,
<http://homepages.inf.ed.ac.uk/rbf/HIPR2/distance.htm>