

Sisteme de recunoaștere a formelor – Lab 4

Transformata distanță (DT). Potrivirea modelelor folosind DT

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 scor 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 scorul de potrivire este mai mic, cu atât obiectul necunoscut este mai asemănător modelului.

2. Fundamente teoretice

2.1. Transformata distanță

Transformata distanță, cunoscută și ca hartă de distanțe sau câmp de distanțe, este o reprezentare a unei imagini digitale. Alegerea termenilor depinde de un punct de vedere, și anume dacă imaginea inițială este transformată în altă reprezentare sau este îmbunătățită cu o informație suplimentară (hartă sau câmp). 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 grayscale, 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 DT 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						

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 metrici sunt:

- Distanța Euclidiană
- Distanța *City block* sau *Manhattan*;
- Distanța „tablă de șah”.

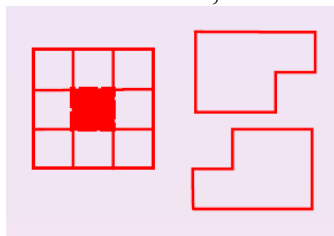
Există mai mulți algoritmi pentru implementarea DT:

- Chamfer DT;
- Euclidian DT;
- Voronoi diagram DT.

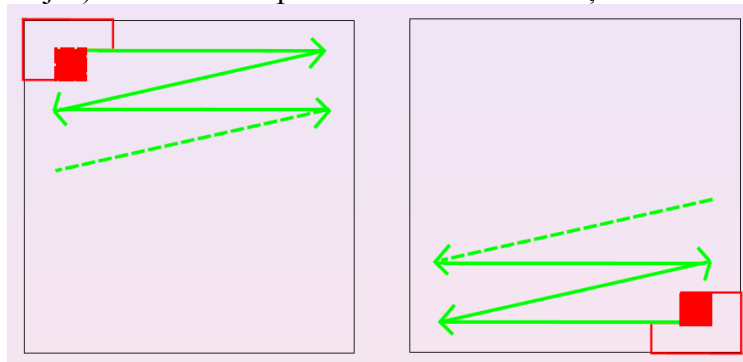
Vom prezenta transformata distanță Chamfer, care este simplă și foarte rapidă, necesitând doar două parcurgeri ale imaginii binare.

O vedere de ansamblu a algoritmului este:

- Se alege o mască de dimensiune 3x3, care se descompune în două părți:



- O dubla 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:



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

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

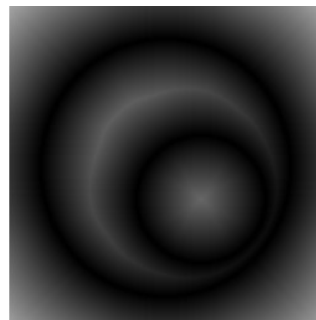
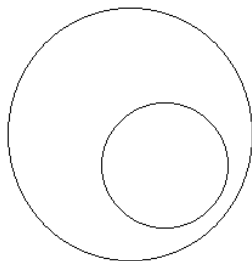
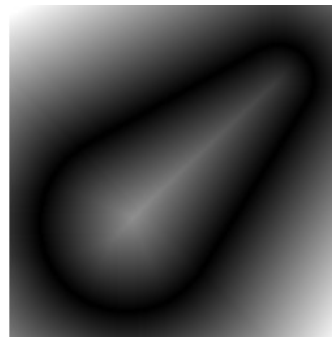
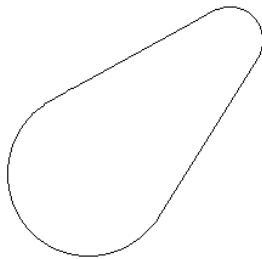
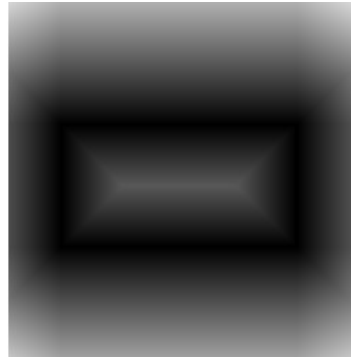
- Inițializarea imaginii DT se face în felul următor:

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

- Precizări:

- Deplasamentul (0,0) cu ponderea 0 aparține măștii;
- Pentru o imagine pe 8 biți pentru infinit se folosește valoarea 255;
- Dacă dorim să obținem o imagine DT unde fiecare pixel aproximează distanța Euclidiană la cel mai apropiat obiect, atunci ponderea $weight(k, l)$ trebuie să aibă valoarea:
 - w_{HV} dacă direcția (k, l) este orizontală sau verticală față de originea măștii
 - w_D dacă direcția (k, l) este diagonală față de originea măștii
 - relația dintre w_{HV} și w_D trebuie să fie $w_D \approx w_{HV} \cdot \sqrt{2}$
 - o alegere bună este $(w_{HV}, w_D) = (5, 7)$ deoarece $7 \approx 5 \cdot \sqrt{2}$

Exemple de imagini DT obținute cu metoda Chamfer, cu valorile $(w_{HV}, w_D) = (2, 3)$. În stânga e imaginea originală, în dreapta imaginea DT.



2.2. Potrivirea de modele folosind DT

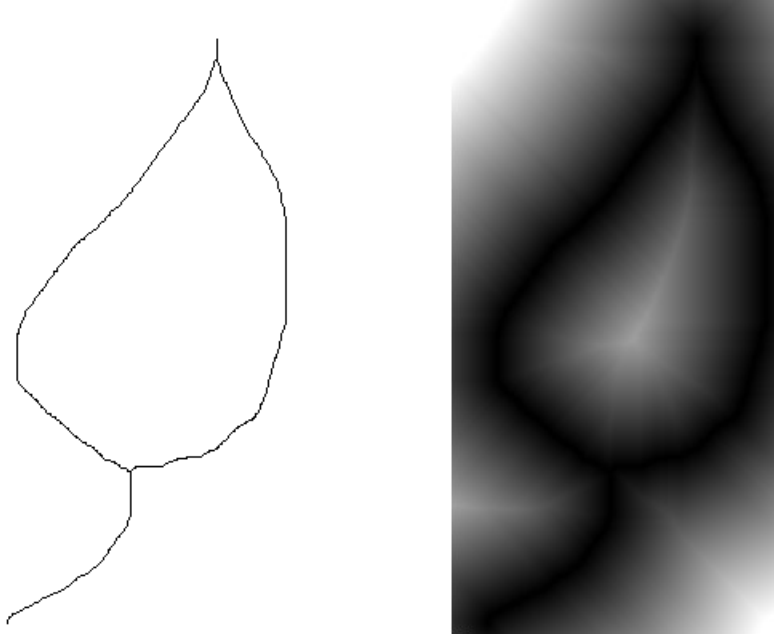
Dorim să calculăm un scor de potrivire între un model de obiect cunoscut și un obiect necunoscut. Considerăm că ambele obiecte au aceeași dimensiune.

Pașii pentru calcularea scorului sunt:

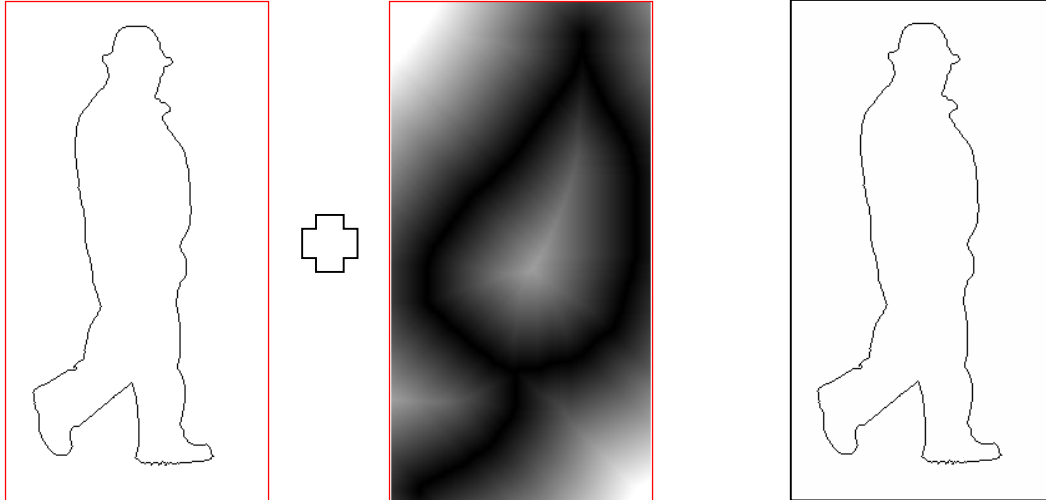
- Calcularea imaginii DT pentru imaginea de model;
- Suprapunerea obiectului necunoscut peste imaginea DT calculata anterior;
- Scorul de potrivire este media tuturor valorilor pixelilor din imaginea DT care se află sub punctele de pe conturul obiectului necunoscut.

Exemplu: Considerăm că obiectul model este o frunză (contur), și obiectul necunoscut este un pieton (contur):

- Calculăm imaginea DT a frunzei:



- Suprapunem pietonul peste imaginea DT a frunzei; calculăm scorul de potrivire ca media valorii tuturor pixelilor din imaginea DT a frunzei care cad sub pixelii conturului pietonului.



Înainte să calculați scorul translați obiectul necunoscut astfel încât centrul său de masă să coincidă cu centrul 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. Detalii de implementare

Citirea imaginii de intrare ca și o poză grayscale:

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

Inițializarea imaginii DT cu imaginea de intrare:

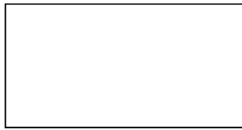
```
Mat dt = src.clone();
```

Accesarea pixelilor dintr-o vecinătate de tip 8:

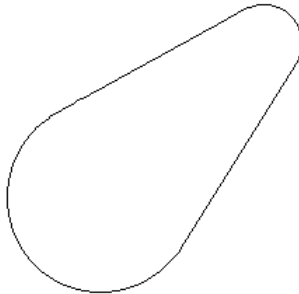
```
int di[8] = {-1,-1,-1,0,0,1,1,1};
int dj[8] = {-1,0,1,-1,1,-1,0,1};
int weight[8] = {0,1,0,1,1,0,1,0};
for(int k=0; k<8; k++)
    uchar pixel = img.at<uchar>(i+di[k], j+dj[k]);
```

4. Activitate practică

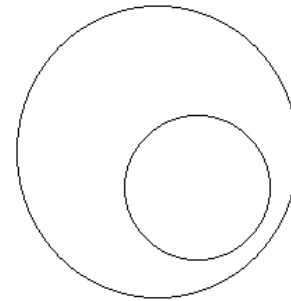
1. Implementați Transformata Distanță Chamfer. Calculați și vizualizați imaginea DT pentru imaginile de intrare: *contour1.bmp*, *contour2.bmp*, *contour3.bmp*. Rezultatele trebuie să coincidă cu cele prezentate în text. Pixelii de obiect sunt negri iar fundalul este alb.



contour1.bmp

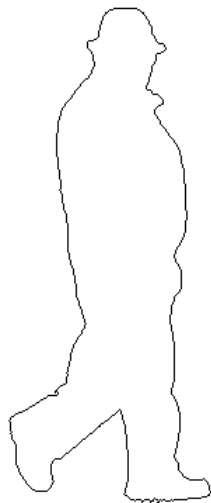


contour2.bmp

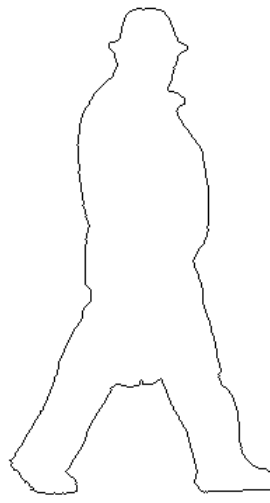


contour3.bmp

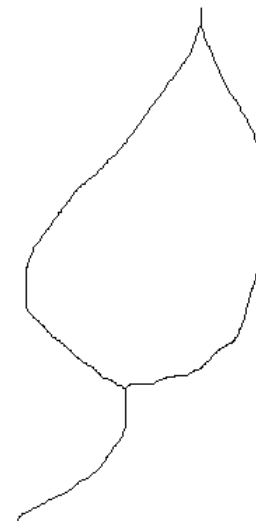
2. Calculați imaginea DT pentru *template.bmp*. Evaluați scorul de potrivire între imaginea model și cele două obiecte necunoscute: *unknown_object1.bmp* – pieton, *unknown_object2.bmp* – frunză. Scorul de potrivire este media valorilor din imaginea DT de pe pozițiile punctelor de contur ale obiectului necunoscut.



template.bmp



unknown_object1.bmp



unknown_object2.bmp

3. Calculați scorul de potrivire 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 scorurile de potrivire.

5. Opțional, implementați Transformata Distanță Euclidiană adevărată. De ce este diferită față de Transformata Distanță Chamfer?

5. Referinte

[1] Wikipedia The Free Encyclopedia – *Distance Transform*,

http://en.wikipedia.org/wiki/Distance_transform

[2] D. Coeurjolly – *Distance Transform*, University Claude Bernard Lyon

http://www.cb.uu.se/~tc18/subfields/distance_skeletons/DistanceTransform.pdf

[3] Compendium of Computer Vision – *Distance Transform*,

<http://homepages.inf.ed.ac.uk/rbf/HIPR2/distance.htm>