Sisteme de recunoastere a formelor

Laborator 2: RANSAC – potrivirea unui cerc la un set de puncte

1. Obiective:

Scopul acestui laborator este aplicarea metodei RANSAC la o noua problema: estimarea parametrilor unui cerc pe baza unui set de puncte 2D.

2. Metoda RANSAC - recapitulare

Obiectiv: Potrivirea robusta a unui model parametric la un set de date S, care contine si puncte zgomot.

Algoritm:

1. Selectie aleatoare a unui esantion de puncte s din multimea S si instantiere a modelului din acest esantion.

2. Determinarea multimii de date S_i care contine puncte situate la o distanta mai mica decat un prag t de model. Multimea S_i este multimea de consens a esantionului, si defineste punctele din S care satisfac modelul.

3. Daca dimensiunea lui S_i (numarul de puncte care satisfac modelul) este mai mare decat un prag T, algoritmul se termina. Optional, se poate re-estima modelul folosind toate punctele din S_i.

4. Daca dimensiunea lui S_i este mai mica decat T, se selecteaza un nou subset si se repeta pasii anteriori.

Dupa N incercari, se selecteaza multimea de consens S_i cu cele mai multe puncte, si (optional) modelul este re-estimat folosind toate punctele din aceasta multime.

3. RANSAC pentru cercuri

Problema de rezolvat este: dandu-se un set de puncte 2D, sa se gaseasca cercul care minimizeaza suma patratelor distantelor de la contur la aceste puncte.

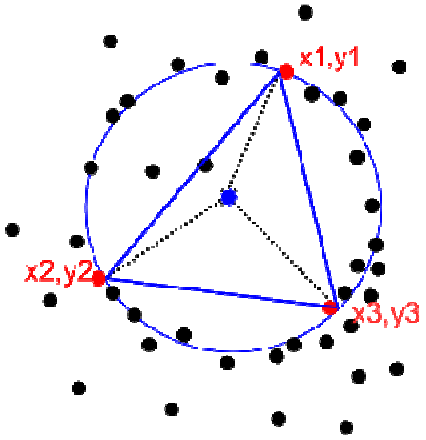
$\varepsilon^2 = \sum_{i=1}^{n} (\sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} - r)^2$ cu conditia ca nici unul dintre puncte sa nu devieze de la cerc cu mai mult de t unitati. Aici (x_c,y_c) reprezinta coordonatele centrului cercului, iar r este raza.

De fapt sunt doua probleme: potrivirea unui cerc la un set de date, si clasificarea datelor ca date (puncte) valide si puncte de tip zgomot. Pragul *t* este ales in functie de zgomotul din setul de date.

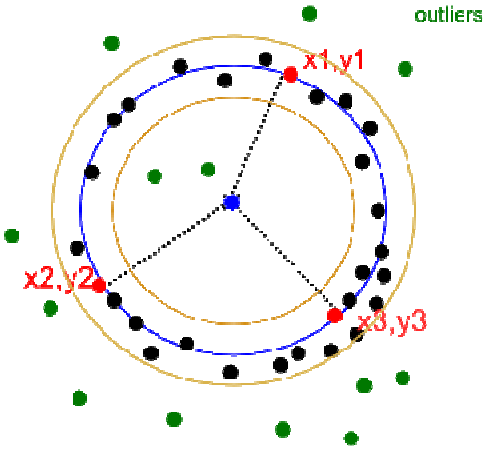
Un cerc este complet definit de trei puncte, care vor fi alese aleator. Multimea suport pentru acest cerc este multimea punctelor care sunt la o distanta mai mica decat *t* de acest cerc. Selectia aleatoare este repetata de mai multe ori, iar cercul cu cea mai mare multime suport este declarat cea mai robusta potrivire.

Algoritmul:

1. Se aleg aleator 3 puncte. (*s=3*). Gasiti cercul care circumscrie triunghiul format de cele trei puncte..



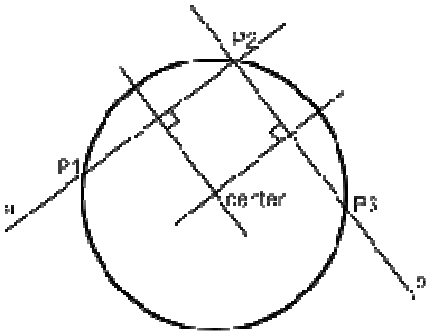
2. Calcularea distantei celorlalte puncte fata de cercul gasit, si numararea celor care sunt la o distanta mai mica decat *t*.



3. Daca numarul de puncte suport este mai mare decat un prag *T*, algoritmul se termina, altfel se repeta pasii anteriori.
4. Dupa un numar de *N* incercari, se alege cercul cu cea mai mare multime suport (consens).

4. Fundamente matematice

4.1. Gasirea parametrilor unui cerc ce trece prin trei puncte *P1, P2, si P3 dintr-un plan.*



Coordonatele centrului se calculeaza astfel:

- Doua drepte se formeaza din doua perechi de puncte. Prima (*a*) trece prin punctele *P1* si *P2*. Dreapta *(b)* trece prin urmatoarele doua puncte, *P2* si *P3*.
- Ecuatiile celor doua drepte sunt: $y_a = m_a(x - x_1) + y_1$ si $y_b = m_b(x - x_2) + y_2$, unde *m* este panta dreptei,: $m_a = \frac{y_2 - y_1}{x_2 - x_1}$ si $m_b = \frac{y_3 - y_2}{x_3 - x_2}$
- Centrul cercului este intersectia dintre mediatoarele segmentelor P_1P_2 si P_2P_3.
- Perpendiculara pe o dreapta de panta *m* are o panta *-1/m*, astfel ca ecuatiile mediatoarelor segmentelor P_1P_2 SI P_2P_3 sunt:

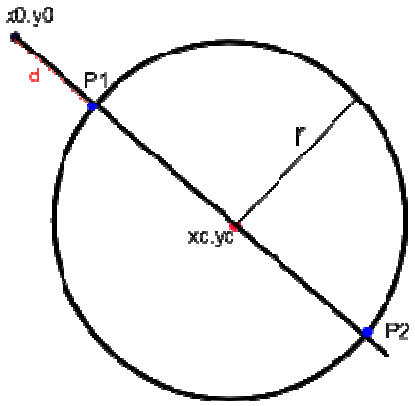
$$y_a^{'} = -\frac{1}{m_a}\left(x - \frac{x_1 + x_2}{2}\right) + \frac{y_1 + y_2}{2}$$

$$y_b^{'} = -\frac{1}{m_b}\left(x - \frac{x_2 + x_3}{2}\right) + \frac{y_2 + y_3}{2}$$

- Cele doua drepte se intersecteaza in centrul cercului, deci daca sistemul de mai sus se rezolva pentru *x* se gaseste: $x_c = \frac{m_a m_b (y_1 - y_3) + m_b (x_1 + x_2) - m_a (x_2 + x_3)}{2(m_b - m_a)}$
- Folosind valoarea lui *x*, putem calcula *y*: $y_c = -\frac{1}{m_a}\left(x_c - \frac{x_1 + x_2}{2}\right) + \frac{y_1 + y_2}{2}$

Raza cercului se calculeaza usor, tinand cont ca punctul P_1 este pe cerc, iar coordonatele centrului (x_c, y_c) sunt cunoscute. $r = \sqrt{(x_c - x_1)^2 + (y_c - y_1)^2}$

4.2. Calculul distantei dintre un punct (x_0, y_0) si un cerc.



Daca se considera dreapta definita de punctul (x_0, y_0) si centrul cercului, aceasta dreapta intersecteaza cercul in punctele P_1 si P_2. Distanta este segmentul format de punctul (x_0, y_0) si cel mai apropiat punct de pe cerc, P_1.

$$d = \left| \sqrt{(x_c - x_0)^2 + (y_c - y_0)^2} - r \right|$$

5. Notiuni de implementare – desenarea cercului

```
void CDibView::OnProcessingRANSACCircle()
{
	BEGIN_PROCESSING();
	/*********************************************
	TODO:
	 Write down the code for RANSAC circle

	*********************************************/

	/* Drawing a circle on the image */
	CDC dc;
	dc.CreateCompatibleDC(0);
	CBitmap ddBitmap;
	HBITMAP hDDBitmap = CreateDIBitmap(::GetDC(0),
			&((LPBITMAPINFO)lpS)->bmiHeader, CBM_INIT, lpSrc,
			(LPBITMAPINFO)lpS, DIB_RGB_COLORS);

	ddBitmap.Attach(hDDBitmap);
	CBitmap* pTempBmp = dc.SelectObject(&ddBitmap);
	CPen pen(PS_SOLID, 1, RGB(255,0,0));
	CPen *pTempPen = dc.SelectObject(&pen);

	/* draw a circle having radius r and center a point of
	coordinates (x,y)*/
	int x = 100;
	int y = 90;
	int r = 20;

	dc.MoveTo ( (int)(x + r), dwHeight-1-y );
```

```
        dc.AngleArc(x, dwHeight-1-y, r, 0, 360);

        dc.SelectObject(pTempPen);
        dc.SelectObject(pTempBmp);
        GetDIBits(dc.m_hDC, ddBitmap, 0, dwHeight, lpDst,
                  (LPBITMAPINFO)lpD, DIB_RGB_COLORS);

        END_PROCESSING("RANSAC-circle");
}
```

6. Activitate practica:

Folosind mediul Diblook, implementati algoritmul descris. Folositi imaginea *circle.bmp* pentru datele de intrare. Punctele sunt desenate cu negru (valoarea 0 pentru pixel). Desenati rezultatul obtinut peste imaginea de intrare. Incercati cu alte imagini desenate de dumneavoastra. Parametrii pentru imaginea data sunt: $w = 0.5$, $p = 0.99$; pentru t puteti incerca mai multe variante, precum 10, 12, 15 pixeli. (vezi laboratorul 1 pentru explicatii).

7. Bibliografie:

[1] Alexander Hornberg: *Handbook of Machine Vision*, 2006
[2] Robert C. Bolles, Martin A. Fischler: A RANSAC-Based Approach to Model Fitting and Its Application to Finding Cylinders in Range Data, 1981
[3] Richard Hartley, Andrew Zisserman: *Multiple View Geometry in Computer Vision,*2003