

VISIÓN POR COMPUTADORA

Reporte - Tarea 8: Registro de rostros

Jaime Rangel Ojeda
zS22000513@estudiantes.uv.mx

Maestría en Inteligencia Artificial

IIIA Instituto de Investigaciones en Inteligencia Artificial
Universidad Veracruzana
Campus Sur, Calle Paseo Lote II, Sección 2a, No 112
Nuevo Xalapa, Xalapa, Ver., México 91097

9 de mayo de 2023

1. Objetivo

Registrar 1520 imágenes usando una imagen de referencia y obtener las 99 imágenes que se parezcan a la imagen de referencia (menor error de ajuste).

2. Metodología

- usamos **fitgeotrans(movingPoints,fixedPoints,tformType)** ajusta una transformación geométrica lineal de tipo tformType a los pares de puntos de control movingPoints y fixedPoints.
- **imref2d** almacena la relación entre las coordenadas intrínsecas ancladas a las filas y columnas de una imagen 2D y la ubicación espacial de las ubicaciones de las mismas filas y columnas en un sistema de coordenadas.
- **imwarp(A,RA,tform)** transforma una imagen con referencia espacial especificada por los datos de la imagen A y su objeto de referencia espacial asociado RA. Las salidas son una imagen con referencia espacial especificada por los datos de la imagen B y su objeto de referencia espacial asociado RB.

fórmula para el cálculo de error:

$$\sqrt{\sum_{i=1}^n (\text{fixedPoints}_i - \text{movedPoints}_i)^2}$$

Donde **fixedPoints** y **movedPoints** representan una matriz de 20*2 (20 puntos con posiciones en X y Y)

3. Resultados

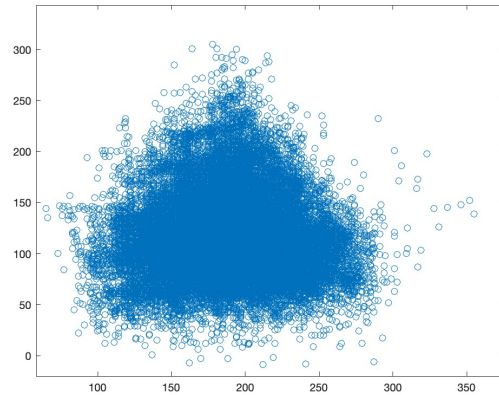


Figura 1: Mapa de distribución de puntos antes del registro de imágenes

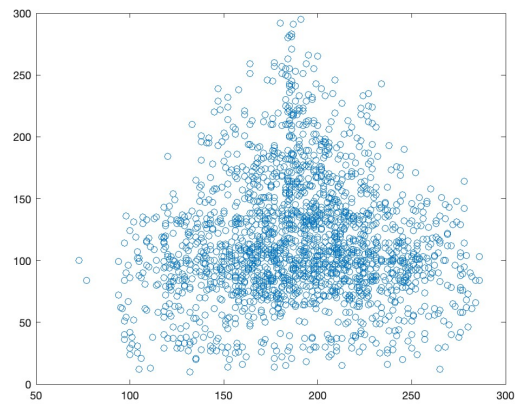


Figura 2: Mapa de distribución de puntos de las 100 imágenes registradas

4. Conclusiones

fitgeotrans Se usa comúnmente en aplicaciones de procesamiento de imágenes y procesamiento de señales para tareas como el filtrado de imágenes, la compresión y la extracción de características.

Genera transformaciones geométricas de una imagen para ajustarla a una imagen de referencia, la función es parte del toolbox de Image Processing.

El registro de transformación geométrica es el proceso de transformar una imagen en la cual está alineada a una imagen de referencia, la transformación puede incluir, rotación, translación, escalamiento y distorsión.

fitgeotrans usa pares de puntos de control para estimar la transformación, haciendo un mapeo de los puntos de control de la imagen de referencia, una vez estimados los puntos se pueden aplicar a una imagen transformada usando **imwarp**

5. Código

```
1 clear all;
2 close all;
3 load data_points.mat
4
5 [r,c,p] = size(data);
6 points = zeros(0,2);
7
8 index = 1;
9
10 for i = 1:r
11     for ip = 1:p
12
13         points(index,1) = data(i,1,ip);
14         points(index,2) = data(i,2,ip);
15         index = index + 1;
16     end
17 end
18
19 plot(points(:,1),points(:,2), 'o')
20
1 clear all;
2 close all;
3
4 load data_points.mat
5 directory = 'Im_Faces';
6 Files=dir(strcat(directory,'/*.pgm'));
7 [rows,columns,points] = size(data);
8
9 %image reference
10 refImage = imread(strcat(directory,'/',Files(1).name));
11
12 %Points
13 index_fixed_points = 1;
14 fixedPoints = zeros(0,2);
15
16 for ip = 1:points
17     fixedPoints(index_fixed_points,1) = data(1,1,ip);
```

```

18     fixedPoints(index_fixed_points,2) = data(1,2,ip);
19     index_fixed_points = index_fixed_points + 1;
20 end
21
22 PosError = zeros(rows, 1);
23
24 for i = 1:rows
25     %iterate for image registration
26     regImage = imread(strcat(directory, '/', Files(i).name));
27
28     %Points
29     movingPoints = zeros(0,2);
30     index_points = 1;
31
32     for ip = 1:points
33         movingPoints(index_points,1) = data(i,1,ip);
34         movingPoints(index_points,2) = data(i,2,ip);
35         index_points = index_points + 1;
36     end
37
38     tform = fitgeotrans(movingPoints,fixedPoints,'affine');
39     Roriginal = imref2d(size(refImage));
40     recovered = imwarp(regImage,tform,'OutputView',Roriginal);
41
42     %error
43     movedPoints = transformPointsForward(tform, movingPoints);
44     distance = sqrt(sum((fixedPoints - movedPoints).^2, 2));
45     PosError(i) = sum(distance);
46
47 end
48
49 [errorPoint, errorIndex] = sort(PosError);
50 %index of the original data with calculate error
51 selectBest = errorIndex(1:100);
52
53 pointsDistribution = zeros(0,2);
54 ipoints = 1;
55
56 for i = 2:100
57     index = selectBest(i);

```

```
58
59     for ip = 1:points
60
61         pointsDistribution(ipoints,1) = data(index,1,ip);
62         pointsDistribution(ipoints,2) = data(index,2,ip);
63         ipoints = ipoints + 1;
64     end
65 end
66
67 plot(pointsDistribution(:,1),pointsDistribution(:,2), 'o')
```