

Universidad Veracruzana

MAESTRÍA EN INTELIGENCIA ARTIFICIAL
VISIÓN POR COMPUTADORA

TAREA 8
REGISTRO DE IMÁGENES

Ulises Jiménez Guerrero

29 de abril de 2025

REPORTE TAREA 8

1. Objetivos

- Cargar la base de datos BioID y sus respectivos puntos de control. Visualizar la nube de puntos.
- Realizar un registro sobre las 1520 imágenes, tomando como referencia la primera imagen.
- Seleccionar las 99 imágenes con el menor error de ajuste. Visualizar la nube de puntos para las 100 imágenes seleccionadas.

2. Metodología

2.1. Materiales utilizados

Se utilizó el lenguaje de programación de Matlab, versión académica R2024b. Se requirió del uso del *Image Processing Toolbox* para el registro masivo de imágenes, además del uso de la base de datos BioID [1] para las imágenes de prueba. Esta contiene un conjunto de 1520 imágenes de rostros de personas, junto a un conjunto de 20 puntos de referencia como se observa en la figura 1.

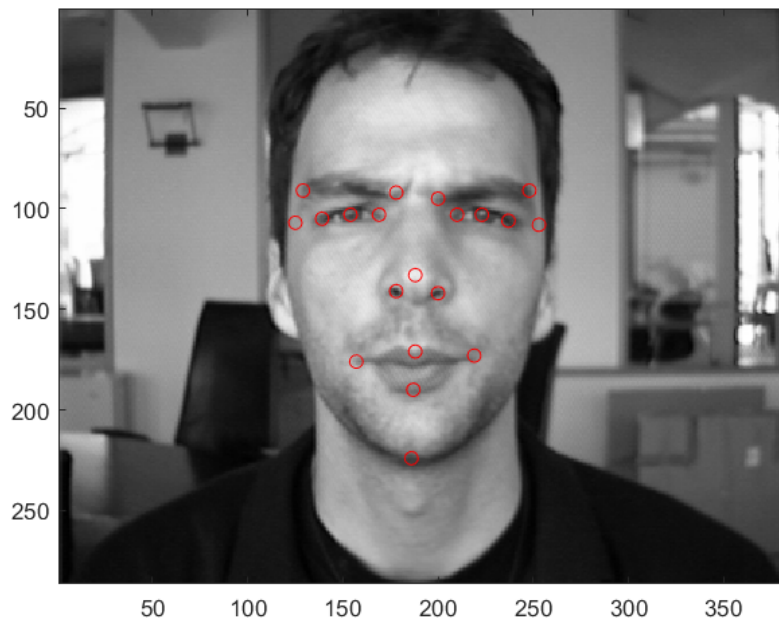


Figura 1: Ejemplo de imagen de la base de datos, mostrando los 20 puntos de referencia.

Como referencia, se tiene al libro [2] para el estudio del proceso de registro de imágenes mediante transformaciones afines.

2.2. Base teórica

El *registro de imágenes* es el proceso de alinear dos o más imágenes de la misma escena, tomadas desde ángulos diferentes. Se tiene una *imagen de referencia* y la imagen a registrar. Se busca encontrar una transformación geométrica sobre la imagen a registrar tal que esta se encuentre alineada con la referencia. El problema a solucionar es encontrar dicha transformación geométrica, la cual debe de estimarse.

Este proceso se lleva a cabo cuando existen diversas imágenes del mismo objeto con características ligeramente diferentes, por ejemplo, un satélite que toma fotografías de la misma ciudad durante días diferentes. Para realizar un análisis significativo sobre estas imágenes, se debe tener en cuenta las diferencias causadas por deformaciones geométricas debidas a cambios en el ángulo, distancia, orientación del objeto, etc.

Para encontrar la transformación geométrica necesaria para el registro, se suelen utilizar *puntos de control*. Estos son puntos presentes en ambas imágenes, cuya posición se conoce con precisión. Se encuentra la transformación geométrica que alinea estos puntos, y se aplica al resto de la imagen a registrar. Si se seleccionaron los puntos de control adecuados, el resultado debería ser la imagen registrada deseada.

La exactitud del registro dependerá de la técnica de aproximación utilizada y de las características de las imágenes. Se pueden realizar mejores registros seleccionando más puntos de control, o técnicas de aproximación más sofisticadas.

2.3. Implementación

Se desarrolló un script compuesto de 4 partes, de forma que cada una se corresponde a uno de los objetivos deseados. Se empieza cargando la base de datos, que ya se encuentra en formato de Matlab. Esta contiene los 20 puntos de referencia de cada una de las 1520 imágenes. Además de esto, se carga la primera imagen, que será la que se utilizará de referencia.

Para la visualización de la nube de puntos, se empieza graficando la primera imagen. Sobre esta, se van graficando cada uno de los 20 puntos de una de las imágenes, iterando sobre todas las imágenes de la base de datos. El color de los puntos es asignado de manera automática por Matlab, sin embargo, la paleta de colores es limitada, por lo que algunos se repiten.

En el registro de las imágenes se empieza seleccionando como referencia los puntos de control de la primera imagen. Después, esta se elimina de la base de datos, para evitar registrar una imagen consigo misma (lo que daría como resultado la transformación correspondiente a la matriz identidad). Se crea una celda de Matlab para guardar los puntos registrados, y se itera para realizar el registro sobre los puntos de cada una de las 1520 imágenes. Para esto, se utilizó la función `fitgeotform2d(puntos_actual, puntos_referencia, 'affine')`. De esta forma, se obtiene una matriz de transformaciones afines para el registro de las imágenes con respecto a los puntos de referencia. Se necesitó utilizar funciones de Matlab para tener una mejor eficacia, dado que el proceso puede ser costoso computacionalmente. Posteriormente, se encontró la transformación de los puntos de referencias de la imagen a registrar mediante el método `trans_matrix.transformPointsForward(puntos_actual)`, y los puntos transformados se guardan en la celda de registros. Esto se repite sobre cada conjunto de puntos en la base de datos.

Para obtener las 99 mejores imágenes, se calcula la distancia promedio en el eje X y en el eje Y para cada uno de los puntos de referencia en las imágenes registradas con respecto a los puntos de control. Posteriormente, se calcula la distancia euclidiana promedio, y esto se toma como el error de ajuste. Estos valores se guardan en una matriz junto a su índice correspondiente. Se ordena la matriz de acuerdo a estas distancias y se recuperan los primeros 99 índices, de forma que se tiene las 99 imágenes con el menor error de ajuste. Se grafican los resultados de manera idéntica al inicio del script.

3. Resultados

En la imagen 2 se muestran las 1520 nubes de puntos de referencia originales. En la imagen 3 se muestran los puntos correspondientes a las 99 mejores imágenes sin registrar, y finalmente en la imagen 4 se muestran estos mismos puntos después del registro.

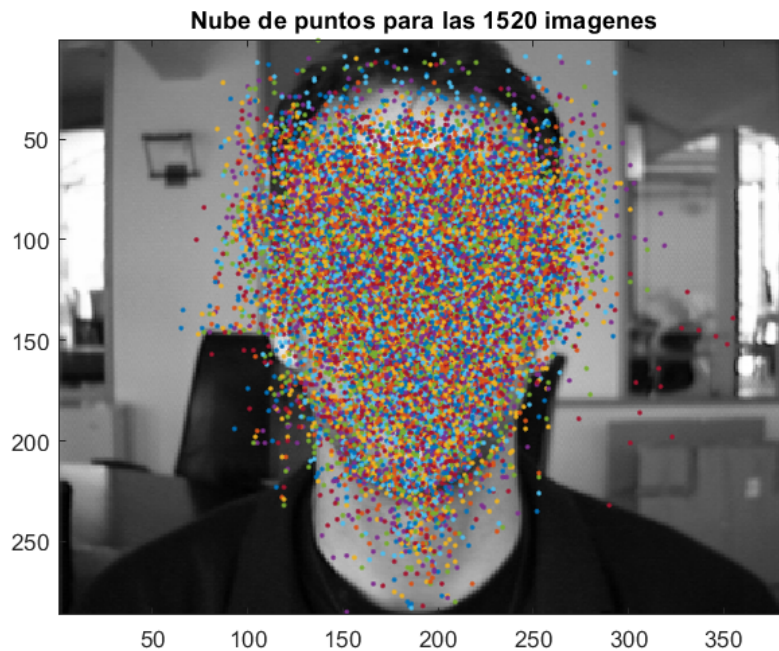


Figura 2: Puntos de referencia de las 1520 imágenes, con la primera imagen como base. Se encuentran tan dispersos que no se logra apreciar ningún patrón.

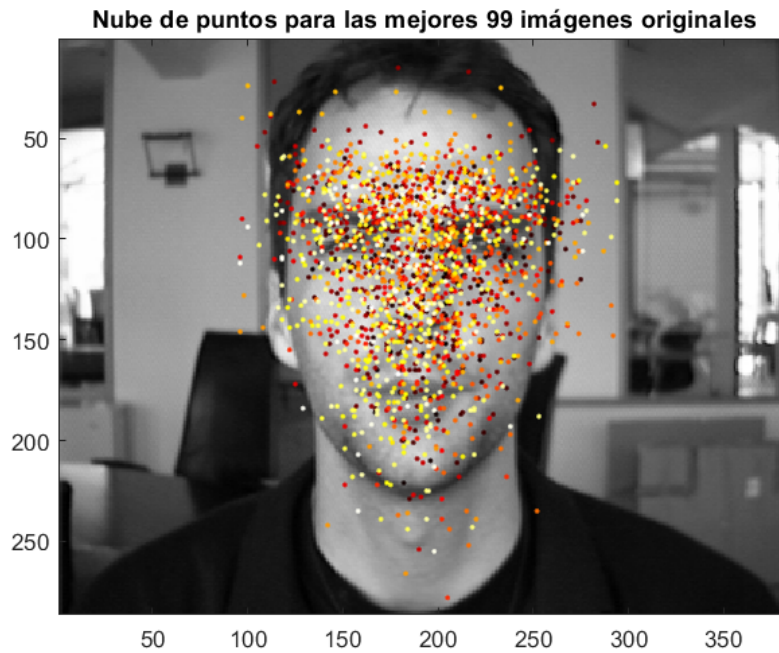


Figura 3: Puntos de referencia de las 99 mejores imágenes originales. Se empieza a ver un patrón sobre el rostro, pero todavía presentan un alto grado de dispersión.

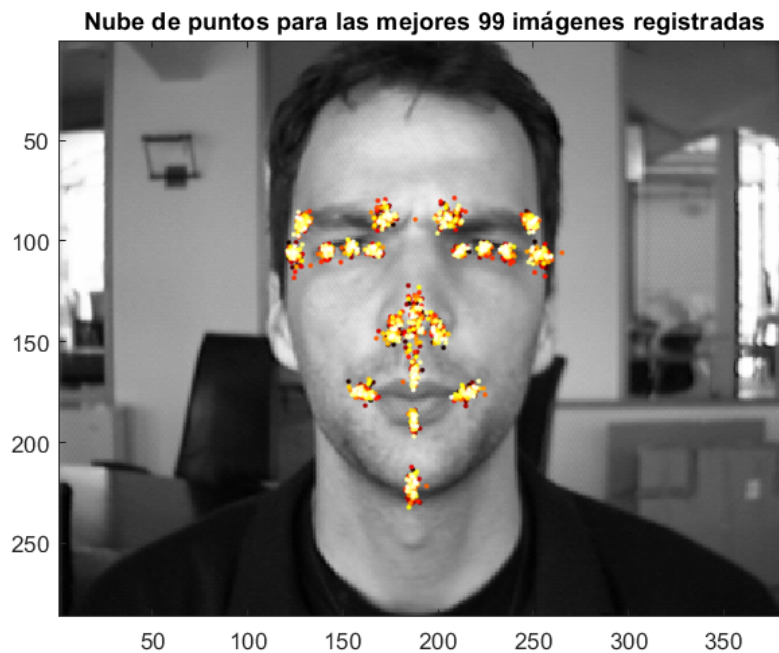


Figura 4: Puntos de referencia de las 99 mejores imágenes registradas. Se nota claramente a qué parte de la cara corresponde cada punto, pero se sigue observando cierta dispersión debido al error en el registro.

4. Conclusiones

Se programó un script que permite el registro de la base de datos conforme a los puntos de referencia pre-etiquetados. Se necesitó utilizar funciones nativas de Matlab para mejorar la eficacia, y aún así el programa presenta dificultades con algunos de los cálculos. Se nota que, incluso después de realizar el registro, se tiene un ligero error, lo que significa que las imágenes resultantes no se encontrarán alineadas de manera perfecta. Sin embargo, el proceso se realizó con transformaciones afines, por lo que queda comprobar los resultados con transformaciones más complejas.

El elegir las imágenes con el menor error de ajuste permite seleccionar aquellas que presentan mayor similitud con la imagen original con respecto a los puntos de referencias seleccionados. Esto se puede utilizar para seleccionar un conjunto de imágenes similares sobre el que aplicar técnicas como PCA, LDA o identificación de objetos.

Referencias

- [1] BioID. The BioID Face Database | BioID. <https://www.bioid.com/face-database/>, November 2022.
- [2] Rafael C. Gonzalez and Richard E. Woods. *Digital image processing*. Pearson, New York, fourth edition, global edition edition, 2017.

5. Anexo

5.1. Código

```
%% Generar nube de puntos para las 1520 imagenes
% Cargar base de datos
load data_points

% Primera imagen como referencia
face1 = imread("Im_Faces\BioID_0001.pgm");

% Visualizacion de la primera imagen
imagesc(face1);
colormap gray

% Visualizaci n de las 1520 nubes de puntos, correspondientes
  a cada cara
no_imgs = 1520;
hold on
for n=1:no_imgs
    plot(squeeze(data(n,1,:)), squeeze(data(n,2,:)), '.')
end
hold off
title("Nube de puntos para las 1520 imagenes")

%% Proceso de registro de imagenes
% Puntos correspondientes a la primera imagen
puntos_referencia = squeeze(data(1,:,:))'; % Se reducen a dos
  dimensiones

% Se eliminan los primeros puntos de los datos, no se necesita
  hacer
% registro sobre ellos
data_reduced = data;
data_reduced(1,:,:)= [];
% Celda para guardar los puntos despu s del registro
registros = cell(no_imgs-1, 1);

for idx=1:(no_imgs-1)
    puntos_actual = squeeze(data_reduced(idx,:,:))';
    % Ajuste de la transformacion afin
    trans_matrix = fitgeotform2d(puntos_actual,
        puntos_referencia, 'affine');
    % Proceso de registro mediante metodo
      transformPointsForward
```

```

        res = trans_matrix.transformPointsForward(puntos_actual);
        % Se guardan los puntos de referencia registrados en la
        % celda de
        % registros
        registros{idx} = res;
end

%% Encontrar mejores 99 puntos
no_mejores_puntos = 99;
% Matriz donde se guardara la diferencia de los puntos
% registrados a los
% puntos de referencia
diferencias = zeros([1519, 2]);

for n = 1:(no_imgs-1)
    % Se selecciona un conjunto de puntos de la celda de
    % registros
    puntos_dif = registros{n};
    % Se calcula la diferencia promedio en la dimension x e y
    delta_x = mean(puntos_dif(:,1) - puntos_referencia(:,1));
    delta_y = mean(puntos_dif(:,2) - puntos_referencia(:,2));
    % Se calcula la distancia euclidean
    dist = sqrt(delta_x^2 + delta_y^2);
    % Se guarda la distancia en la primera columna de la matriz
    % de
    % diferencias
    diferencias(n,1) = dist;
    % Se guarda el indice en la segunda columna de la matriz de
    % diferencias
    diferencias(n,2) = n;
end

% Se ordena la matriz de diferencias en sus filas, en orden
% ascendente de
% acuerdo a la primera columna, que contiene las distancias
diferencias = sortrows(diferencias);

% Se eligen las primeras 99 filas, correspondientes a los 99
% puntos mas
% cercanos a la referencia
puntos_seleccionados = diferencias(1:no_mejores_puntos, 2);

%% Visualizacion de los mejores 99 puntos
% Puntos sin registrar
figure;

```



```

imagesc(face1);
colormap gray
hold on

cmap_puntos = hot(99);
for n=1:99
    idx_actual = puntos_seleccionados(n);
    plot(squeeze(data(idx_actual,1,:)), squeeze(data(idx_actual
        ,2,:)), ...
        ' . ', Color=cmap_puntos(n,:))
end
hold off
title("Nube de puntos para las mejores 99 im genes originales
    ")

%Puntos registrados
figure;
imagesc(face1);
colormap gray

hold on
for n=1:99
    idx_actual = puntos_seleccionados(n);
    puntos_actual = registros{idx_actual};
    plot(puntos_actual(:,1), puntos_actual(:,2), ' . ', ...
        Color=cmap_puntos(n,:))
end
hold off
title("Nube de puntos para las mejores 99 im genes registradas
    ")

```