
Table of Contents

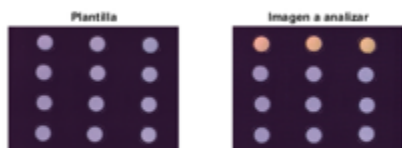
H1. Control de calidad mediante la diferencia de imágenes (David Morais & Alejandro Ruiz)	1
1. Procesamiento de las imágenes para encontrar los centros de las pastillas y analizar alineamiento	1
2. Obtención de la matriz de transformación afín	3
3. Transformación geométrica de la imagen	3
4. Diferencia de imágenes alineadas y resaltado de las pastillas defectuosas	4

H1. Control de calidad mediante la diferencia de imágenes (David Morais & Alejandro Ruiz)

Leer las imágenes

```
im1 = imread("Blispac1.tif");
im2 = imread("Blispac2.tif");

figure;
subplot(1,2,1); imshow(im1); title('Plantilla'), impixelinfo;
subplot(1,2,2); imshow(im2); title('Imagen a analizar'), impixelinfo;
```



Pixel info (X,Y) Pixel Value

1. Procesamiento de las imágenes para encontrar los centros de las pastillas y analizar alineamiento

La idea es pasar las imágenes a escala de grises para posteriormente binarizarlas separando así el fondo (color negro) de las pastillas (color blanco). Una vez binarizadas calculamos los centros y con su diferencia observamos si están alineadas o no

```
% Convertimos las imágenes a escala de grises
gray1 = rgb2gray(im1);
gray2 = rgb2gray(im2);

% Umbral para segmentar las pastillas
level = graythresh(gray1); % Se ajusta de forma automática
```

```

bw1 = imbinarize(gray1, level);
bw2 = imbinarize(gray2, level);

% Imágenes binarizadas de las pastillas
figure;
subplot(1,2,1); imshow(bw1); title('Imagen 1 binarizada'), impixelinfo;
subplot(1,2,2); imshow(bw2); title('Imagen 2 binarizada'), impixelinfo;

% Establecemos el umbral mínimo para el área en píxeles que debe tener una
% pastilla
minArea = 500; % De esta forma eliminamos posible ruido al calcular el
centro de las pastillas

% Filtramos las regiones según el área y obtenemos su centro.
% regionprops recibe una imagen binaria y almacena los centros y áreas de
% las superficies con valor positivo.
stats1 = regionprops(bw1, 'Centroid', 'Area');
stats2 = regionprops(bw2, 'Centroid', 'Area');

% Eliminamos las regiones con área menor al umbral, así solo quedan pastillas
stats1 = stats1([stats1.Area] > minArea);
stats2 = stats2([stats2.Area] > minArea);

centroides1 = cat(1,stats1.Centroid);
centroides2 = cat(1,stats2.Centroid);

% Ordenamos columnas de pastillas por valor de y descendiente para poder
% acceder luego por id al comparar pastilla a pastilla
for i = 0:2
    idini = i * 4 + 1;
    idfin = i * 4 + 4;

    centroides1(idini:idfin, :) = sortrows(centroides1(idini:idfin, :), 2);
    centroides2(idini:idfin, :) = sortrows(centroides2(idini:idfin, :), 2);
end

% Imágenes de las pastillas marcando su centro
figure;
imshow(bw1), impixelinfo;
hold on;
for k = 1:length(stats1)
    plot(stats1(k).Centroid(1), stats1(k).Centroid(2), 'ro'); % Mostramos
los centros con un círculo de color rojo
end
title('Centroides de las pastillas de la imagen 1');

% Igual para la imagen 2
figure;
imshow(bw2), impixelinfo;
hold on;
for k = 1:length(stats2)
    plot(stats2(k).Centroid(1), stats2(k).Centroid(2), 'go'); % Mostramos
los centros con un círculo de color verde
end

```

```

title('Centroides de las pastillas de la imagen 2');

% Comprobamos la alineación de las imágenes
% Para ello observaremos si la diferencia de todos los centros es nula o
% realmente mínima.

% Inicializamos un umbral de píxeles de diferencia entre centros
umbral = 5;
alineadas = true;

% Comparamos las distancias de cada par de centros
for k = 1:length(stats1)
    centroide1 = centroides1(k);
    centroide2 = centroides2(k);

    dist = norm(centroide1 - centroide2);
    if dist > umbral
        alineadas = false;
        break;
    end
end

if alineadas
    disp("ALINEADAS");
else
    disp("NO ALINEADAS");
end

```

NO ALINEADAS

2. Obtención de la matriz de transformación afín

Resolvemos el sistema de ecuaciones `fitgeotform2d` es una función que recibe una matriz de puntos a mover, una matriz de puntos de referencia y el tipo de transformación geométrica lineal. Devuelve la matriz de transformación afín

```
T = fitgeotform2d(centroides2, centroides1, "affine");
```

3. Transformación geométrica de la imagen

Aplicamos la transformación afín a la imagen 2

```

im2Alineada = imwarp(im2, T, 'OutputView', imref2d(size(im1))); % Aquí T
es la matriz afín calculada

% Visualizamos la imagen alineada
figure;
subplot(1,3,1); imshow(im2); title('Imagen Desalineada'), impixelinfo;
subplot(1,3,2); imshow(im2Alineada); title('Imagen Alineada'),
impixelinfo;
subplot(1,3,3); imshow(im1); title('Imagen Original'), impixelinfo;

```



Pixel Info (X, Y) Pixel Value

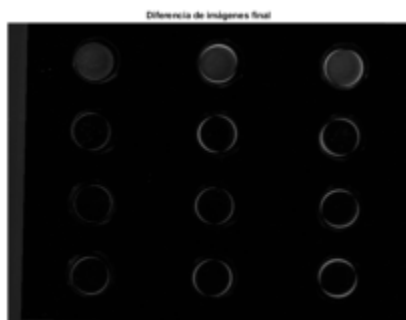
end

4. Diferencia de imágenes alineadas y resalta- do de las pastillas defectuosas

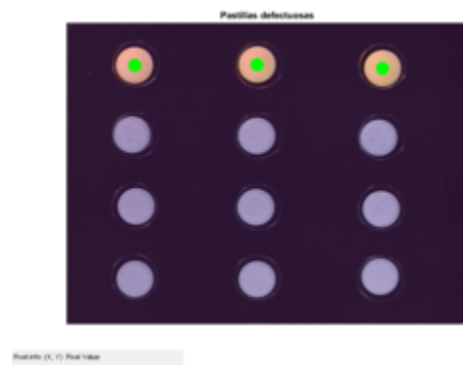
Hacemos la diferencia de las dos imágenes y pasamos de rgb a gray para trabajar únicamente con un nivel de gris

```
diff = imabsdiff(im1, im2Alineada);
diff = rgb2gray(diff);
figure, imshow(diff), title("Diferencia de imágenes final"), impixelinfo;

% Finalmente, marcamos las pastillas defectuosas con un círculo verde en la
% imagen analizada
figure;
imshow(im2), impixelinfo;
hold on;
for k = 1:length(centroides1)
    x1 = uint16(centroides1(k,1));
    y1 = uint16(centroides1(k,2));
    x2 = uint16(centroides2(k,1));
    y2 = uint16(centroides2(k,2));
    if (diff(y1,x1) > 10)
        plot(x2, y2, 'go', "LineWidth", 10); % Mostramos los centros con un
        círculo de color rojo
    end
end
title("Pastillas defectuosas"), impixelinfo;
```



Pixel Info (X, Y) Pixel Value



Published with MATLAB® R2024b