
Títol: Control de qualitat mitjançant la imatge diferència

Table of Contents

Pas 1: Obténir la imatge diferència entre el patró i la imatge Blispac2	1
Pas 2: Obténir la matriu de transformació afí (T)	2
Pas 3: Calcular la transformació afí	4
Pas 4: Obténir la imatge diferència després de l'alineació	4

Autors: Jordi Muñoz Florensa, Joan Carles Veny Martí

Pas 1: Obténir la imatge diferència entre el patró i la imatge Blispac2

```
img1 = imread('Blispac1.tif'); % Imatge patró
img2 = imread('Blispac2.tif'); % Imatge amb error

% Calculem la diferència absoluta entre les dues imatges per veure quina
% diferència hi ha sense cap processament
img_diff = imabsdiff(img1, img2);

subplot(1,3,1), imshow(img1), title('Imatge Blispac1');
subplot(1,3,2), imshow(img2), title('Imatge Blispac2');
subplot(1,3,3), imshow(img_diff), title('Diferència de imatges');
```



Pas 2: Obtenir la matriu de transformació afí (T)

```
% Per tal de fer la detecció automàtica vam trobar la funció imfindcircles
% que busca cercles que tinguin un radi compres entre els valors de
% l'interval i retorna els centres i el radi dels cercles.
[centers1, radii1] = imfindcircles(img1, [20 50]);
[centers2, radii2] = imfindcircles(img2, [20 50]);

% Com que la funció imfindcircles no ens retorna els cercles aparellats,
% hem decidit realitzar aparellar els cercles que estiguin més propers en
% les dues imatges
paired1 = [];
paired2 = [];
distances = zeros(size(centers1, 1), size(centers2, 1));

% Calculem la matriu de distàncies entre tots els centres detectats.
for i = 1:size(centers1, 1)
    for j = 1:size(centers2, 1)
        distances(i, j) = norm(centers1(i, :) - centers2(j, :)); % Distància
euclidiana
    end
end

% Ordenem els centres per trobar la millor correspondència.
for i = 1:size(centers1, 1)
    [~, closestIdx] = min(distances(i, :)); % Trobar el centre més proper
    paired1 = [paired1; centers1(i, :)];
    paired2 = [paired2; centers2(closestIdx, :)];
    distances(:, closestIdx) = Inf; % Evitar repetir el mateix centre
end

% Mostrem els centres ordenats per verificar que l'aparellament sigui
correcte.
disp('Cercles en ordre per img1');
disp(paired1);
disp('Cercles en ordre per img2');
disp(paired2);

% Calculem la transformació afí entre les dues imatges utilitzant
`fitgeotrans`.
tform = fitgeotrans(paired2, paired1, 'affine');
disp('Matriu de transformació afí');
disp(tform.T);

% Mostrem els cercles detectats a les imatges.
figure;
subplot(1, 2, 1), imshow(img1), viscircles(centers1, radii1, 'EdgeColor',
'b'); title('Cercles detectats a img1');
subplot(1, 2, 2), imshow(img2), viscircles(centers2, radii2, 'EdgeColor',
'r'); title('Cercles detectats a img2');

Cercles en ordre per img1
    326.1547    406.8743
```

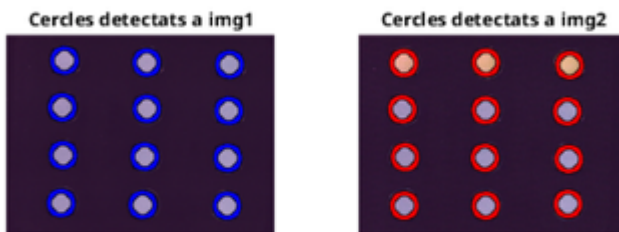
533.5382	71.5023
527.4338	408.9935
528.8950	295.1289
136.1535	288.1338
331.1770	291.8519
531.5125	181.9745
139.5125	61.5007
336.2537	177.0517
131.6993	402.9297
335.8478	66.8467
134.4205	173.6341

Cercles en ordre per img2

303.9627	408.9777
504.1790	72.8628
500.1494	404.8718
501.9615	297.4460
111.3428	292.8975
302.0011	294.4533
498.4278	183.5854
108.5880	67.9783
303.6981	180.6936
109.7563	408.8195
304.2920	66.5345
104.5938	178.5229

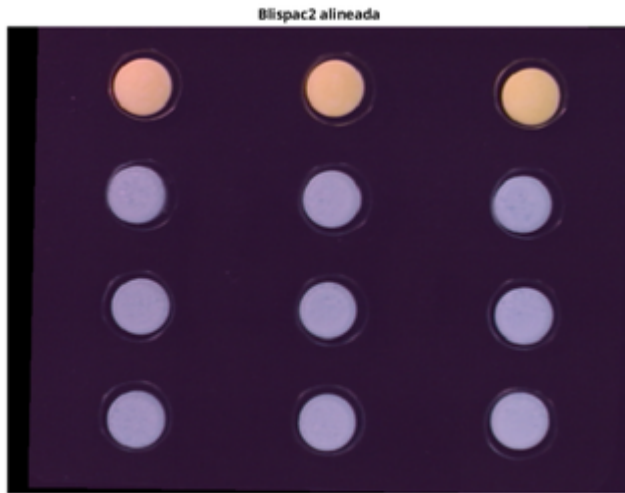
Matriu de transformació afí

1.0060	0.0133	0
-0.0225	1.0034	0
31.8295	-7.4446	1.0000



Pas 3: Calcular la transformació afí

```
% Apliquem la transformació afí a `img2` per alinear-la amb `img1`.  
img2_aligned = imwarp(img2, tform, 'OutputView', imref2d(size(img1)));  
figure, imshow(img2_aligned), title('Blispac2 alineada');
```



Pas 4: Obtenir la imatge diferència després de l'alineació

Calculem la diferència entre `img1` i `img2_aligned`.

```
imgDiff = imabsdiff(img1, img2_aligned);  
  
% Per obtenir una imatge on poguem apreciar la diferència al màxim, convertim  
% la imatge de diferència a escala de grisos i la binaritzem.  
imgDiffGray = rgb2gray(imgDiff);  
imgDiffBinaria = imbinarize(imgDiffGray, 0.1);  
figure, imshow(imgDiffBinaria), title("Diferència alineada i binaritzada");  
  
% Posteriorment amb la intenció d'eliminar les zones on es detecta una petita  
% diferència entre les imatges,  
% que es produeix degut al propi plàstic en el que van encapsulades, vam  
% trobar una funció `bwareaopen` per eliminar regions petites que ens  
% permet mitjançant un parametre especificar el mínim tamany de píxels  
% concatenats que volem que es mostrin  
imgDiffBinariaNeta = bwareaopen(imgDiffBinaria, 500);  
figure, imshow(imgDiffBinariaNeta), title("Diferència neta");
```

