

Transformaciones Geométricas (Rotación, Traslación, Factor Escala)

Contents

- [Rotación](#)
- [Traslación](#)

Rotación

La rotación de una imagen puede realizarse mediante el comando `imrotate`. Es necesario especificar además el ángulo y el método de remuestreo. Por ejemplo:

```
police=imread('police.jpg');  
subplot(1,2,1)  
imshow(police)  
subplot(1,2,2)  
imshow(imrotate(police,90))
```



como puede verse un ángulo positivo implica un sentido de giro contrario al de las agujas del reloj. Los comandos `flipud` y `fliplr` revuelven imágenes reflejadas de abajo arriba y de izquierda a derecha. Estas funciones no obstante sólo pueden trabajar con matrices 2D. De modo que es necesario realizar la transformación para cada canal.

```
subplot(2,2,1)  
imshow(police)  
title('Original')  
subplot(2,2,2)  
pol1=flipud(police(:,:,1));pol2=flipud(police(:,:,2));pol3=flipud(police(:,:,3));  
pol(:,:,1)=pol1;pol(:,:,2)=pol2;pol(:,:,3)=pol3;  
imshow(pol)  
title('flipud')  
subplot(2,2,3)  
pol1=fliplr(police(:,:,1));pol2=fliplr(police(:,:,2));pol3=fliplr(police(:,:,3));  
pol(:,:,1)=pol1;pol(:,:,2)=pol2;pol(:,:,3)=pol3;  
imshow(pol)  
title('fliplr')  
subplot(2,2,4)  
imshow(imrotate(police,180))  
title('imrotate, 180°')
```

Original



flipud



fliplr



imrotate, 180°



Factor de Escala

En MATLAB las transformaciones geométricas pueden generarse mediante dos funciones `maketform` e `imtransform`. La función `maketform` crea una estructura que contiene una transformación espacial que luego se aplicará a la imagen con la función `imtransform`. Para emplear la primera función es necesario escribir la transformación mediante una matriz en coordenadas homogéneas. Por ejemplo un cambio de escala diferente en la dirección de las x y de las y's será.

```
sx=0.75; sy=1.25;
```

```
T=[sx 0 0;0 sy 0; 0 0 1]
```

```
T =
```

0.75	0	0
0	1.25	0
0	0	1

Esta matriz es una transformación de tipo afín muy sencilla en donde no existe rotación ni ángulo de falta de perpendicularidad entre ejes (shear) ni traslación. Para crear la estructura que contenga esta matriz escribiremos:

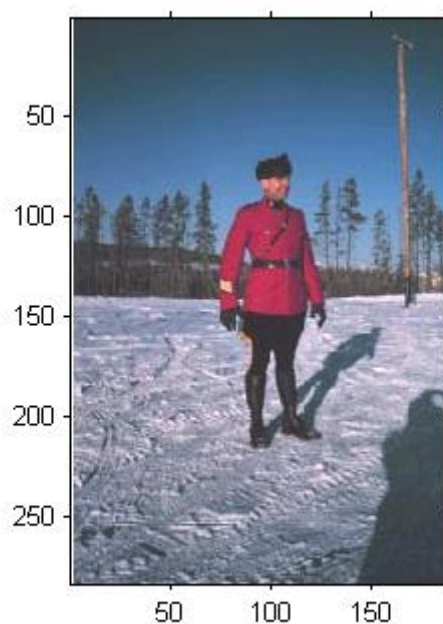
```
t1=maketform('affine',T);
police_affine=imtransform(police,t1);
figure
subplot(1,2,1)
imshow(police)
subplot(1,2,2)
imshow(police_affine)
```



Traslación

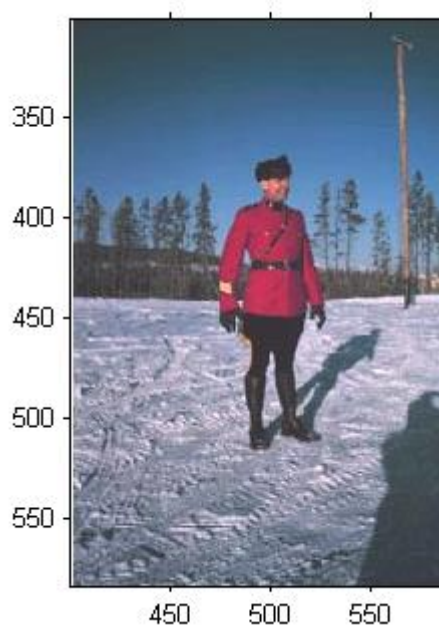
Trabajando de un modo semejante es posible efectuar una traslación, por ejemplo de 100 píxeles en la dirección horizontal y de 200 en la vertical.

```
figure
Ttras=[1 0 0;0 1 0; 400 300 1]
t2=maketform('projective',Ttras);
imshow(imtransform(police,t2))
axis on
Ttras =
     1     0     0
     0     1     0
    400    300     1
```



Sin embargo pese a haber efectuado una traslación en los dos ejes no resulta apreciable porque MATLAB automáticamente ajusta el espacio de la figura a la imagen que se visualiza exclusivamente. Para evitar este comportamiento es posible recuperar dos nuevos parámetros de la función `imtransform` como son `xdata` e `ydata`, y aplicarlos a la función `imshow`.

```
figure
[pol_tras,xdata,ydata]=imtransform(police,t2);
imshow(pol_tras,'XData',xdata,'YData',ydata)
axis on
```



Aun así es poco evidente la traslación puesto que sólo queda indicada por las etiquetas de los ejes. Veamos otro método para poner en la misma figura las dos imágenes:

```
imshow(police)
hold on
imshow(pol_tras,'Xdata',xdata,'YData',ydata)
axis auto % aumenta el espacio visible hasta el máximo de ambas figuras
axis on
```

