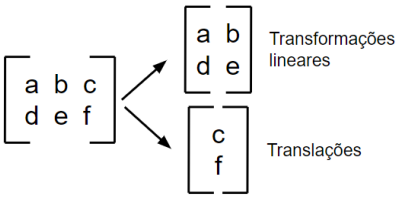
As transformações geométricas disponíveis no OpenCV são: redimensionamento, translação, rotação, transformação afim e transformação de perspectiva. Cada uma delas tem seu objetivo específico para manipular e modificar a geometria de imagens para que se adapte às suas necessidades.

- Redimensionamento: ajusta o tamanho da imagem, aumentando ou diminuindo, pode ser feito manualmente ao dar valores para a altura e largura da imagem, ou automaticamente especificando o fator de escalonamento. Essa função pode ser usada para padronizar o tamanho de um grupo de imagens, por exemplo.
- Translação: desloca uma imagem considerando o eixo x e o eixo y, para isso deve ser especificado o deslocamento nos dois eixos e se é positivo ou negativo. Um exemplo de uso dessa função é alinhar uma imagem em relação a outra em um website.
- Rotação: gira uma imagem e tem a possibilidade de ajustar o centro de rotação. Uma situação que ela pode ser usada é para ajustar a rotação de uma foto que foi tirada no celular mas ficou de cabeça para baixo.
- Transformação afim: é uma transformação na qual todas as linhas paralelas da imagem inicial vão manter seu paralelismo na imagem final. As três transformações iniciais e o cisalhamento são consideradas transformações afins.
- Transformação de perspectiva: ajusta a perspectiva de uma imagem de acordo com a seleção de 4 pontos. Um uso prático é que essa função pode ser usada para criar um quadrilátero em uma foto de um documento e assim conseguir retirar o fundo da imagem e ajustar sua perspectiva para que ele fique retinho.

As funções cv2.warpAffine e cv2.warpPerspective do OpenCV são utilizadas para aplicar as transformações geométricas em imagens, a primeira usa uma matriz de transformação 2x3 enquanto a segunda usa uma 3x3.

A matriz 2x3 abaixo pode ser usada para representar a operação geométrica de redimensionamento, translação e rotação, que são todas transformações afins. Ela pode ser dividida em duas sub-matrizes que representam as transformações lineares e as translações.



Para o escalonamento é preciso multiplicar o valor das coordenadas por um número determinado, maior que 1 para dar aumentar a imagem ou menor que 1 para diminuir a imagem. Por ser uma transformação linear, serão feitas modificações apenas na primeira submatriz, multiplicando os valores das coordenadas do ponto pelos elementos não-nulos da matriz.

$$\begin{bmatrix} e_x & 0 & 0 \\ 0 & e_y & 0 \end{bmatrix}$$

Como para a translação iremos mover a imagem para cima ou para baixo e para a direita ou para a esquerda precisamos adicionar um valor de distância d_x e d_y aos valores de x e y iniciais nos componentes referentes à translação (“c” e “f”) e com isso conseguimos fazer a matriz abaixo. É necessário manter os valores padrões dos outros componentes já que eles não estão relacionados às translações e já que a translação não é considerada uma transformação linear.

$$\begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \end{bmatrix}$$

Para a rotação, a matriz é similar a do escalonamento, já que também é uma transformação linear, mas para calcular os valores já é mais complicado pois é necessário fazer o cálculo da soma de arcos para deduzir a fórmula de rotação e por fim encontrar a matriz de transformação abaixo.

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Por fim, para a matriz de perspectiva também é possível expressá-la em coordenadas homogêneas, como na imagem abaixo e terá uma lógica similar a matriz de transformação afim e funcionará similarmente no código. Assim, é necessário colocar no código os valores das coordenadas nos quais você quer transformar a perspectiva e o tamanho da imagem transformada que você deseja e com isso ele irá manter todas as linhas horizontais paralelas e ajustar as verticais, que irão perder seu paralelismo no final da transformação, e a mudança na matriz dependerá da mudança específica que é desejada para a imagem.

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ w_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$