Transformações Geométricas Matlab e qualquer outro software que faça uso de operações matriciais

Israel Jesus Santos Filho

Operações Geométricas

- Operação de Translação/Rotação/Escalonamento:
  - Todas essas operações dependem das coordenadas do pixel na imagem em questão;
  - A imagem transformada receberá as informações associadas ao valor do pixel na imagem fonte em uma coordenada diferente. Com o auxílio da álgebra linear estas operações podem ser simplificadas de tal modo que podemos executar as transformações por simples operações matriciais.
  - Eu percebi que se tivesse uma forma de gerar um mapa com as coordenadas da imagem fonte de tal modo que eu as organizasse em um arranjo matricial, todo esse processo de iteração entre as imagens seria reduzido. É necessário iterar ao longo do canal de cor, da dimensão da linha e na dimensão da coluna.

Operações Geométricas

- Exemplo de código:
- Basicamente o que falta é dizer quem é a matriz_T que vai operar as coordenadas.
- Algumas decisões também precisam ser tomadas a respeito do que fazer com pontos que eventualmente sejam mapeados fora do range da imagem fonte. Mas isso é bem particular e depende da utilidade e das necessidades do projeto em que o algoritmo será utilizado.

```
[linhas, colunas, canais] = size(imagem_fonte);


for canal = 1:canais:
  for linha = 1:linhas:
    for coluna = 1:colunas:
      nova_coordenada = matriz_T * [linha; coluna; 1];
      imagem_transformada(linha, coluna, canal) = imagem_fonte(nova_coordenada(1), ...
                                                               nova_coordenada(2), canal);
```

Operações Geométricas

- Exemplo de código:
- Minha ideia foi criar o grid antes usando a função meshgrid disponível, geralmente, em todo ambiente que prototipa operações matemáticas para visualização e tratamento de dados. Numpy,-Scipy-No python, Scilab, Matlab, R e afins...
- Nada impede de se criar manualmente esse grid também, mas como estou usando o matlab e python (a depender da lib usada), as funções internas do bicho são compiladas usando código C.

```
[linhas, colunas, canais] = size(imagem_fonte);


for canal = 1:canais:
  for linha = 1:linhas:
    for coluna = 1:colunas:
      nova_coordenada = matriz_T * [linha; coluna; 1];
      imagem_transformada(linha, coluna, canal) = imagem_fonte(nova_coordenada(1), ...
                                                  nova_coordenada(2), canal);
```

```
>> linhas = 3;
>> colunas = 5;
>> [xx, yy] = meshgrid(1:linhas, 1:colunas);
>> xx
xx =
     1     2     3
     1     2     3
     1     2     3
     1     2     3
     1     2     3
>> yy
yy =
     1     1     1
     2     2     2
     3     3     3
     4     4     4
     5     5     5
```

Operações Geométricas

- Exemplo de código:
- Porque eu estava preocupado com a velocidade? Porque estamos trabalhando com 3 loops aninhados. Esse bicho demorará muito para ser executado caso queiramos processar imagens FullHd, por exemplo.

```
[linhas, colunas, canais] = size(imagem_fonte);


for canal = 1:canais:
  for linha = 1:linhas:
    for coluna = 1:colunas:
      nova_coordenada = matriz_T * [linha; coluna; 1];
      imagem_transformada(linha, coluna, canal) = imagem_fonte(nova_coordenada(1), ...
                                                               nova_coordenada(2), canal);
```

```
>> linhas = 3;
>> colunas = 5;
>> [xx, yy] = meshgrid(1:linhas, 1:colunas);
>> xx
xx =
     1     2     3
     1     2     3
     1     2     3
     1     2     3
     1     2     3
>> yy
yy =
     1     1     1
     2     2     2
     3     3     3
     4     4     4
     5     5     5
```

Operações Geométricas

- Exemplo de código:
- Com esses passos ao lado, criamos exatamente a matriz que eu queria desde o início e utilizando o conceito de coordenadas homogêneas adicionei os 1's necessário para que a combinação de operações possa ser feita sem mais problemas.
- As considerações a respeito da performance serão discutidas em aúdio mesmo hehehehehh!

```
>> coord_im_fonte = [xx(:)  yy(:) ones(length(xx(:)), 1)];
>> size(coord_im_fonte)
ans =
    15     3
>> coord_im_fonte(5,:)
ans =
     1     5     1
```

```
>> cx =.5, cy = .5;
cx =
   0.500000000000000
>> matriz_T = [cx 0 0 ; 0 cy 0; 0 0 1];
>> nova_coordenada = round(coord_im_fonte*matriz_T');
>> size(nova_coordenada)
ans =
    15     3
>> size(coord_im_fonte)
ans =
    15     3
```