

Mc920 - Trabalho 5

Felipe Santana Dias - 215775

1. ESPECIFICAÇÃO DOS PROBLEMAS

O objetivo desse trabalho é implementar quatro métodos de interpolação (*Vizinho Mais Próximo*, *Bilinear*, *Bicúbica* e *Polinômios de Lagrange*) para realizar as transformações geométricas de escala e rotação em uma imagem.

2. ENTRADA DE DADOS

O código desenvolvido para realizar as transformações das imagens foi desenvolvido em python através das bibliotecas OpenCV e NumPy em um arquivo chamados '*transformar.py*'. O programa foi projetado para imagens monocromáticas no formato PNG.

Para executar o código é necessário fornecer algumas informações no momento da requisição, sendo elas:

```
python transformar.py angulo escala dimensão método imagem_entrada.png imagem_saida.png
```

As análises feitas neste relatório estão baseadas nas observações realizadas sobre a imagem '*seagull.png*' (512px x 512px), disponibilizada pelo professor e a imagem '*triangulo.png*' (500px x 500px) criada para observar os resultados dos diferentes métodos. Ambas as imagens estão presentes na pasta raiz juntamente com o código e as imagens resultantes.

Após o processamento, o arquivo gerado será salvo na pasta raiz respeitando a nomenclatura passada na linha de comando.

3. CÓDIGO E DECISÕES TOMADAS

3.1 Algoritmo - *transformar.py*

Inicialmente, a imagem foi carregada através da biblioteca OpenCV e convertida em uma NumPy Array. Também foi criada uma nova NumPy Array de zeros de mesma dimensão que a passada na linha de comando para comportar a nova imagem. Vale ressaltar que, como o objetivo é observar o comportamento dos métodos em algumas transformações, a imagem gerada possui o valor fixo do parâmetro passado, apresentando-se cortada dependendo da transformação realizada. Dessa maneira é possível observar claramente qual foi a modificação ocasionada na imagem.

Em seguida, é chamada a função correspondente ao parâmetro *método*, sendo os valores *vizinho*, *bilinear*, *bicúbica* e *lagrange* os únicos que permitem que as

funções sejam executadas. Caso o termo passado seja diferente desses, será retornado uma mensagem "*Método não encontrado.*".

O código foi implementado de forma a realizar uma única transformação por execução. Logo, é necessário que um dos dois parâmetros *escala* e *ângulo* seja zero quando não utilizado. Além disso esses valores não podem ser menores que zero, podendo gerar erros na execução.

Após recebido todos os parâmetros, as funções criadas para cada método possuem estrutura semelhante: Dois laços encaixados percorrem toda a nova imagem a ser preenchida, sendo que em cada iteração é chamada a função *pixels* que calcula os novos valores de x e y após a transformação aplicada. A partir desses valores é encontrado as distâncias dx e dy do ponto original ao novo ponto e então é calculado os valores das intensidades dos pixels de acordo com cada método.

Após feito esse processo, é gerado a imagem transformada, resultando em um arquivo na pasta raiz cujo título é o escolhido no parâmetro *imagem_saída.png*.

4. SAÍDA DE DADOS

Ao final da execução é gerado uma única imagem transformada de acordo com o nome especificado na linha de comando.

5. RESULTADOS

Para discutir os resultados obtidos utilizaremos as imagens '*triangulo.png*' e '*seagull.png*'.

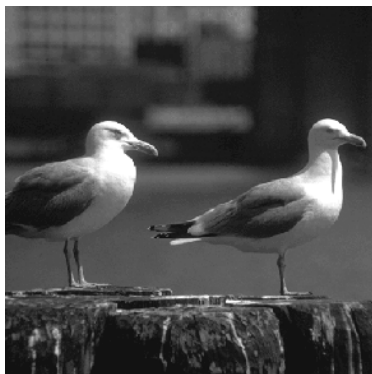


Figura 1 e 2 - '*seagull.png*' e '*triangulo.png*', respectivamente.

5.1 Ampliação

Nesse caso, aumentamos a escala em um fator de 2 e as imagens que obtemos para cada um dos métodos foram as seguintes:

5.1.1 Interpolação Vizinhos mais próximos



Figura 3 - Método Vizinhos mais Próximos

A interpolação por vizinho mais próximo é a de mais fácil implementação mas também é a que mais apresentou aparência pixelada quando realizado o aumento, gerando um forte aspecto serrilhado nas bordas.

5.1.2 Interpolação Bilinear



Figura 4 - Método Bilinear

Por utilizar a média ponderada dos 4 vizinhos mais próximos, a interpolação bilinear apresenta aspecto mais suave e contínuo quando comparado ao método anterior porém a imagem final é marcada por um efeito borrado.

5.1.3 Interpolação Bicúbica



Figura 5 - Método Bicúbica

A interpolação bicúbica apresentou os melhores resultados quando comparado com as duas anteriores - com menos serrilhamento do que o *método dos vizinhos mais próximos* e não tão suavizado quanto o *método bilinear*. Esse comportamento pode ser relacionada à média ser realizada com os 16 vizinhos mais próximos, permitindo uma estimativa melhor dos valores. Também é possível observar que houve uma leve alteração nas cores mais claras.

5.1.4 Interpolação Polinômios de Lagrange

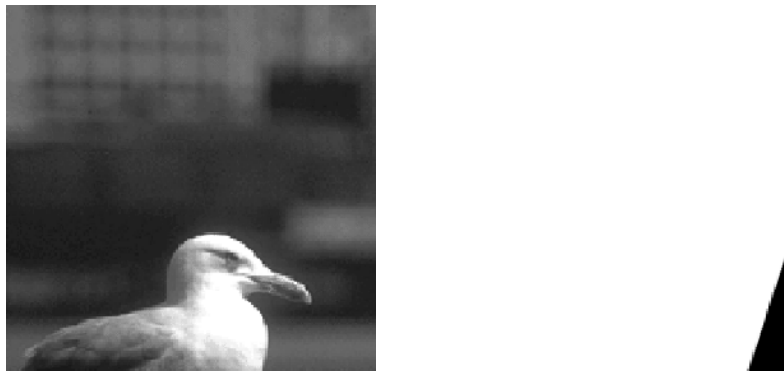


Figura 6 - Método Polinômios de Lagrange

Apesar de também utilizar os 16 vizinhos mais próximos como o *método bicúbico*, a *interpolação por Polinômios de Lagrange* apresentou aspecto mais serrilhado quando comparado ao anterior, não gerando uma significativa melhora no resultado final, apesar de seu algoritmo mais complexo.

5.2 Rotação

Nesse caso, escolhemos o ângulo de 30.3° e as imagens que obtemos para cada um dos métodos foram as seguintes:

5.2.1 Interpolação Vizinhos mais próximos

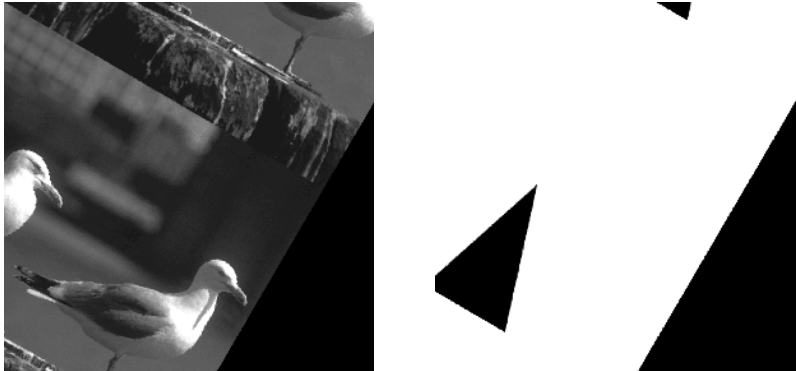


Figura 7 - Método Vizinhos mais Próximos

Semelhante ao observado na mudança de escala, o rotacionamento realizado através desse método gerou um serrilhamento bem marcado principalmente nas extremidades da imagem.

5.2.2 Interpolação Bilinear

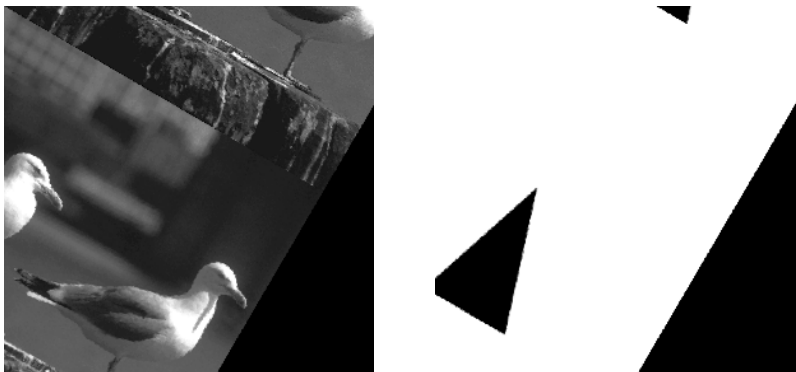


Figura 8 - Método Bilinear

Esperava-se desse método um efeito borrado como já observado mas além dessa alteração é possível notar que esse efeito apresentou-se de maneira mais intensa, podendo ser consequência da angulação na imagem.

5.2.3 Interpolação Bicúbica

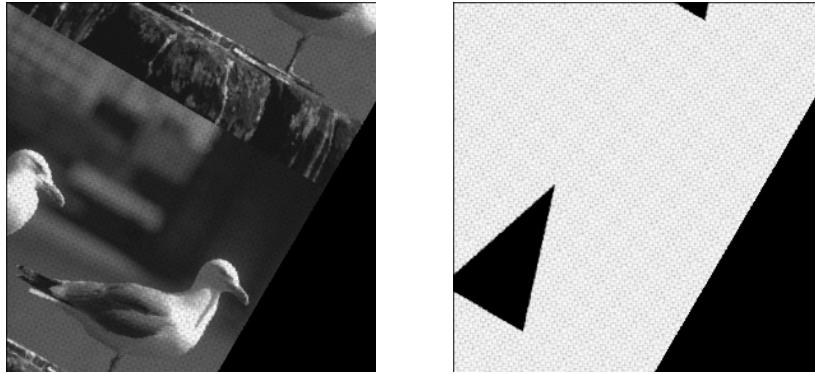


Figura 9 - Método Bicúbica

Apesar de novamente ser o método que apresentou melhores resultados, principalmente no que diz respeito às bordas, com a rotação da imagem surgiu alguns padrões geométricos que encontram-se superpostos por toda a imagem.

5.2.4 Interpolação Polinômios de Lagrange

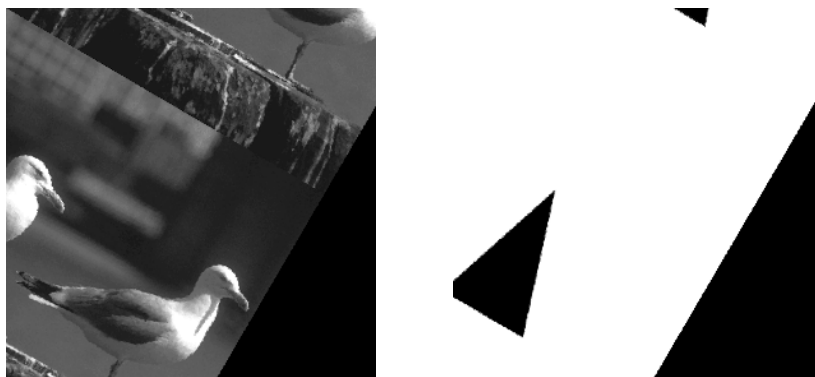


Figura 10 - Método Polinômios de Lagrange

Semelhante ao ocorrido no segundo método, a interpolação por Polinômios de Lagrange gerou um bom resultado mas não completamente satisfatório no que diz respeito ao efeito serrilhado nas bordas.

6. CONCLUSÃO

Observando os resultados obtidos, podemos concluir que algoritmos mais simples e com baixo custo computacional podem apresentar soluções completamente plausíveis para determinadas situações mesmo quando há a possibilidade de utilização de algoritmos mais complexos. Ainda, a utilização de vizinhanças maiores para definir os valores médios melhorou consideravelmente os resultados, sendo que, nas situações observadas, o método da Interpolação Bicúbica foi o que mais se destacou.