

# Trabalho 5

MO443 - Introdução ao Processamento de Imagem Digital

Diego Alysson Braga Moreira

RA:230640

diegoalyssonbm@gmail.com / d230640@dac.unicamp.br

UNICAMP

## I. INTRODUÇÃO

Transformações geométricas é algo bem comum de serem realizadas em imagens por qualquer usuário. A maioria dos pacotes gráficos disponibilizam múltiplos métodos para realizar operações do tipo escala ou rotação. Estas opções têm formas diferentes de serem obtidas, com melhores resultados ou menores tempo de execução.

Neste trabalho propõe-se implementar duas diferentes tipos de transformações geométricas, escala e rotação, em imagens para quatro diferentes operações. As transformações implementadas são métodos de interpolações, dentre eles, Interpolação do vizinho mais próximo, Interpolação Bilinear, Interpolação Bicúbica e Interpolação por Polinômios de Lagrange.

Cada um desses problemas serão abordados e discutidos nas próximas seções, com instruções de execução, modelos e decisões de implementação, saídas dos dados, resultados e conclusões.

## II. EXECUÇÃO

Para a execução padrão deste projeto, o arquivo de implementação dos códigos, imagens e pastas foram enviadas em conjunto. Abaixo segue a lista dos componentes que podem ser encontrados neste trabalho:

- Pasta Códigos: Arquivo de Execução do Código: trabalho5.py (descrito a seguir).
- Pasta com Imagens: baboon.png, butterfly.png, city.png, house.png e seagull.png.
- Pasta Raiz: Pasta Código, Pasta com Imagens e Relatório em PDF.

Onde, as imagens estão em uma subpasta chamada Imagens, a qual deve ser utilizada como padrão de localização das imagens utilizadas como entrada as transformações geométricas. As pastas e seu conteúdo (arquivos .py e imagens) estão na pasta Raiz do projeto, juntamente com o relatório em pdf. O arquivo .py necessário para a execução e observação dos resultados é o arquivo trabalho5.py.

Nesta seção serão apresentadas todas as instruções de execução para ao código enviado, suas dependências, funcionalidades, parâmetros, decisões de implementação e saídas de dados.

O projeto deve ser executado com uma linha básica de chamada no terminal, como exemplo:

```
python3 trabalho5.py -i <imagem_entrada.png>-a <30  
>-m <1 >
```

### A. Dependências

Para a execução correta dos códigos em anexo, as seguintes dependências são essenciais:

- opencv-python: Utilizado para importação, exportação, exibição das imagens.
- NumPy: Manipulação de vetores e matrizes.
- math: Utilização de métodos matemáticos como seno e cosseno.
- sys: Utilizada para aquisição dos argumentos em linha de comando.
- getopt: Este módulo ajuda os scripts a analisar os argumentos da linha de comando em sys.argv.

### B. Funcionalidades

Nesta seção será descrito as funcionalidades disponíveis no código em anexo, os parâmetros recebidos por este código e alguns exemplos de execução. Detalhes de implementação e técnicas são descritas na seção III.

Para este trabalho, foram implementadas duas diferentes técnicas de transformações geométricas, escala e rotação. Entre as quatro operações disponibilizadas para cada uma das duas operações, estão Interpolação do vizinho mais próximo, Interpolação Bilinear, Interpolação Bicúbica e Interpolação por Polinômios de Lagrange.

### C. Parâmetros

Para a correta execução dos códigos em anexo, deve-se seguir o seguinte modelo de forma geral:

```
python3 trabalho5.py (INPUT) [OPCOES]
```

A partir do padrão acima, os elementos em parênteses são obrigatórios, os elementos entre colchetes podem expandir suas possibilidades de acordo com as necessidades de cada questão, com utilização opcional, que serão explicadas a seguir.

- 1) python3: Palavra chave para utilização da linguagem python, 3 edição, no sistema linux.
- 2) trabalho5.py: Parâmetro obrigatório, define o arquivo python a ser executado com os códigos para a solução dos problemas, neste caso, trabalho5.py é o arquivo em questão.
- 3) INPUT: Especifica o arquivo imagem que será tratado através do algoritmo de esteganografia. Também parâmetro obrigatório. Deve-se explicitar o comando Input através da palavra -i ou --ifile, o nome da imagem também deve especificar a extensão do arquivo.

Neste trabalho, arquivos png (.png) foram extensamente testados. Como exemplo:

- `python3 trabalho5.py -i baboon.png`

4) OPCODES: Este parâmetro representa as múltiplas opções disponibilizadas entre as diferentes técnicas disponíveis no trabalho.

- Parâmetro: `-o` ou `--ofile`.
- Descrição: Este parâmetro opcional é utilizado para especificar o nome do arquivo da imagem gerada após o processamento do algoritmos. O nome deve conter também o formato de saída caso deseje-se que a extensão seja definida. Caso não informado, a imagem resultante terá o nome no formato, `Secret-Image<Inputfile>.png` caso o encode esteja selecionado como 1 ou o arquivo de saída terá o nome 'arquivo' caso selecionado o encode modo como 0, que descreve que é uma imagem com conteúdo secreto seguido pelo nome da imagem de entrada. Para definir o valor deste parâmetro, deve-se utilizar a palavra `-o` ou `--ofile`. Como possível caso de utilização, tem-se:
  - `python3 trabalho5.py -i baboon.pgm -e 1 -o saida.png`
- Parâmetro: `-a` ou `--angulo`.
- Descrição: Este parâmetro é utilizados para operações de rotação e define em qual angulo a rotação deve ser realizada. Este parâmetro deve ser utilizado com unidades em graus, aceitando-se valores reais. Para utilizar esta opção, utilize a palavra `-a` ou `--angulo` seguido pelo angulo desejado. Operações de escala têm prioridade se definidas, desta forma não deve-se utilizar a variável angulo ao utilizar opções de escala. Como exemplo:
  - `python3 trabalho5.py -i baboon.png -a 30.578 -m 2`
- Parâmetro: `-e` ou `--escala`.
- Descrição: Para realizar a escala para imagens, pode-se selecionar através da variável escala, em quantas vezes tal imagem deve ser redimensionada. Esta variável aceita valores reais como entrada. A variável escala tem prioridade sobre outras variáveis opcionais de escala ou angulo. Para utiliza-la basta usar a palavra `-e` ou `--escala` seguido pela escala desejada. Como exemplo:
  - `python3 trabalho5.py -i baboon.png -e 2.746 -m 4`
- Parâmetros: `-dh` e `-dw` ou `--dimensaoH` e `--dimensaoW`.
- Descrição: Estas duas variáveis são utilizadas para escala de imagens, dando a opção para o usuário escolher o tamanho final desejado da imagem, sem a necessidade de utilizar a variável escala. Deve-se lembrar que a variável escala tem prioridade sobre outras variáveis de escala, desta forma não deve-se utilizas em conjunto com tais variáveis. Tanto

Parâmetro (INPUT)	Palavra	Possibilidades	Opcionalidade
	<code>-i</code> <code>--ifile</code>	Nome da imagen de entrada	Obrigatória
Opções	<code>-o</code> <code>--ofile</code>	Nome da imagem de saída	Opcional
	<code>-a</code> <code>--angulo</code>	Angulo de rotação para imagem	Opcional] (Valor Padrão: 0)
	<code>-e</code> <code>--escala</code>	Grau de escalamento da imagem	Opcional] (Valor Padrão: 0)
	<code>-dh</code> e <code>-dw</code> <code>--dimensaoH</code> e <code>--dimensaoW</code>	Dimensões definidas para escalamento	Opcional (Valor Padrão: 0, 0)
	<code>-m</code> <code>--method</code>	Método de interpolação utilizado	Opcional (Valor Padrão: 1)

`dimensaoH` quanto `dimensaoW` devem ser utilizadas em conjunto. Para utiliza-las basta citar a palavra `-dh` e `-dw` ou `--dimensaoH` e `--dimensaoH`. Como exemplo:

- `python3 trabalho5.py -i butterfly.png -dh 720 -dw 710 -m 2`

- Parâmetro: `-m` ou `--metodo`.
- Descrição: Para selecionar o método utilizado para escalonar ou rotacionar a imagem, utiliza-se esta variável. São aceitos valores de 1 à 4, correspondendo aos quatro métodos implementados neste trabalho, (1) Interpolação do vizinho mais próximo, (2) Interpolação Bilinear, (3) Interpolação Bicúbica e (4) Interpolação por Polinômios de Lagrange. Exemplos:

- `python3 trabalho5.py -i seagull.png -e 3.745 -m 1`
- `python3 trabalho5.py -i city.png -e 7.745 -m 2`

#### D. Exemplos de Execução

Para facilitar a execução dos códigos, segue alguns exemplos de linhas de código que pode ser utilizadas no terminal e algumas utilizações de parâmetros opcionais.

- `python3 trabalho5.py -i baboon.png -e 1.888 -m 1`
- `python3 trabalho5.py -ifile house.png -escala 9.452 -m 2`
- `python3 trabalho5.py -ifile city.png -e 9.452 -metodo 3`
- `python3 trabalho5.py -ifile baboon.png -dh 720 -dw 1080 -m 3`
- `python3 trabalho5.py -i house.png -a 45.789 -m 2`
- `python3 trabalho5.py -i city.png -angulo 80.785 -m 4`

### III. IMPLEMENTAÇÃO E DECISÕES TOMADAS

Nesta seção será descrito o funcionamento dos algoritmos do trabalho 5 e alguns aspectos mais específicos de implementação e decisões importantes.

De forma geral algumas decisões foram tomadas:

- Todas as entradas aceitam imagens tons de cinza.
- Imagens de diferentes tamanhos podem ser utilizadas. Não necessariamente matrizes quadradas ou com valores pares em linhas e colunas

- Os valores dos parâmetros opcionais, quando não declarados, exibem a imagem com valores padrões.
- Sempre que possível buscou-se utilizar métodos e equações em formato vetorizado.
- É possível utilizar números reais para as transformações propostas em todos os quatro métodos.
- Aplicou-se as interpolações inversas para obter a imagem final

#### A. Transformações Geométricas

Deve-se notar que, como uma imagem digital é por uma matriz de valores, de forma que a posição  $(x, y)$  indica a posição nesta matriz, pode ocorrer de  $(x', y')$  resultar em um valor não inteiro e desta forma, não represente diretamente uma posição na matriz.

Para solucionar este problema utiliza-se interpolações, de forma que uma dada posição da imagem possa ser representada por uma outra. Neste trabalho serão utilizadas quatro diferentes formas de interpolação.

1) *Interpolação pelo vizinho mais próximo*: Para este método, o valor do pixel  $(x', y')$  de uma imagem resultante é dado pelo valor mais próximo  $(x, y)$  na matriz de imagem original. Como ilustrado na imagem descrita no trabalho e reproduzida aqui, Imagem 1

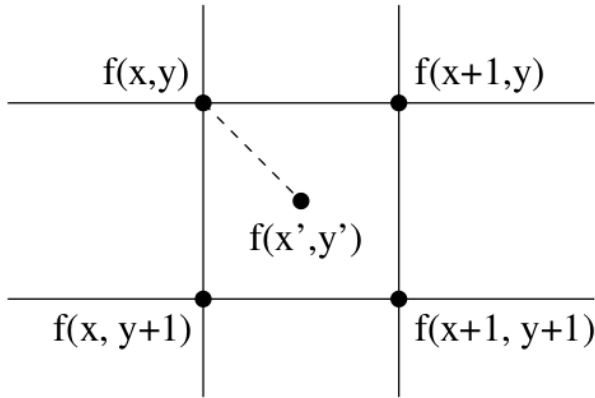


Fig. 1. Interpolação pelo vizinho mais próximo.

A interpolação pode ser expressa pela equação:

$$f(x', y') = f(\text{round}(x), \text{round}(y)) \quad (1)$$

2) *Interpolação Bilinear*: Esta interpolação é a média ponderada da distância dos quatro pixels vizinhos mais próximos à  $(x', y')$ , como demonstrado na imagem do relatório, Imagem 2

data pela seguinte formula:

$$f(x', y') = (1 - dx)(1 - dy)f(x, y) + dx(1 - dy)f(x + 1, y) + (1 - dx)dyf(x, y + 1) + dxdyf(x + 1, y + 1) \quad (2)$$

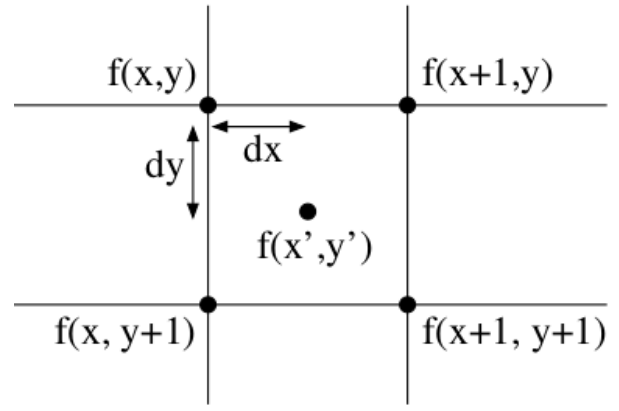


Fig. 2. Interpolação Bilinear.

3) *Interpolação Bicúbica*: Para este método de interpolação utiliza-se a vizinhança 4x4 pontos ao redor de  $(x', y')$  para calcular seu valor de intensidade. Demonstrado na imagem do relatório, Imagem 3

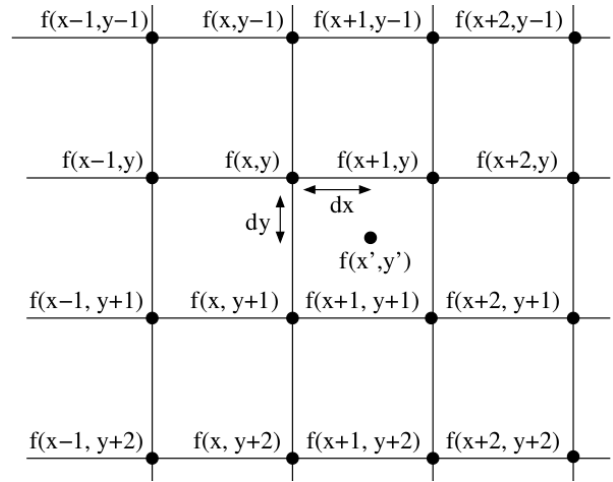


Fig. 3. Interpolação Bilinear.

Esta interpolação pode ser calculada pela função B-spline cúbica, dada por:

$$f(x', y') = \sum_{m=-1}^2 \sum_{n=-1}^2 f(x + m, y + n) R(m - dx) R(dy - n) \quad (3)$$

onde,

$$R(s) = 1/6[P(s+2)^3 - 4P(s+1)^3 + 6P(s)^3 - 4P(s-1)^3] \quad (4)$$

e

$$P(t) = \begin{cases} t, & t > 0 \\ 0, & t \leq 0 \end{cases} \quad (5)$$

4) *Interpolação por Polinômios de Lagrange*: Este método também considera os 4x4 vizinhos de (x',y) para calcular sua intensidade e é definido pela seguinte equação:

$$f(x', y') = \frac{-dy(dy-1)(dy-2)L(1)}{6} + \frac{(dy+1)(dy-1)(dy-2)L(2)}{2} + \frac{-dy(dy+1)(dy-2)L(3)}{2} + \frac{dy(dy+1)(dy-1)L(4)}{6} \quad (6)$$

onde,

$$L(n) = \frac{-dx(dx-1)(dx-2)f(x-1, y+n-2)}{6} + \frac{(dx+1)(dx-1)(dx-2)f(x, y+n-2)}{2} + \frac{-dx(dx+1)(dx-2)f(x+1, y+n-2)}{2} + \frac{dx(dx+1)(dx-1)f(x+2, y+n-2)}{6} \quad (7)$$

#### IV. RESULTADOS

Os resultados obtidos neste trabalho podem ser divididos em dois principais tópicos, resultados para mudança de escala das imagens ou para as rotações. Dentro destes dois tipos de transformações, cada um dos quatro métodos de interpolação foram calculados.

##### A. Escala

Primeiramente, podemos observar a validação para a aplicação da técnica em diferença de escalas, na Imagem 4 tem-se duas imagens resultantes do processo de escala, uma com escala de 0.2 e a segunda com escala 2.7846.

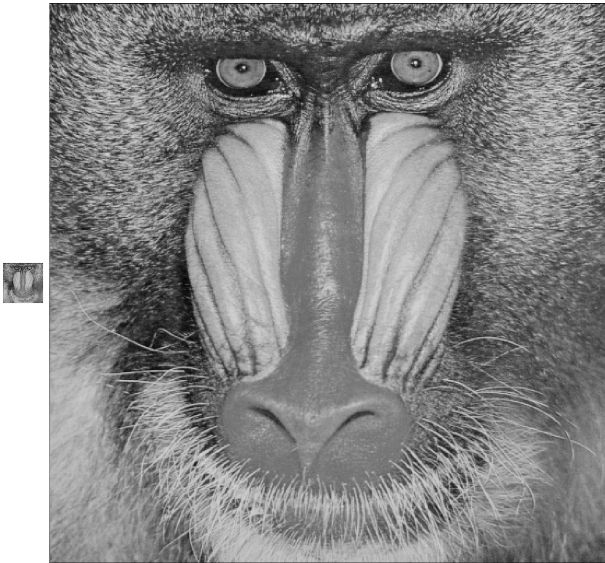


Fig. 4. Imagens resultantes de diferentes escalas.

Para demonstrar as diferenças existentes entre os métodos de interpolação aplicados neste trabalho, serão aplicadas transformações, através destes diferentes métodos, e depois comparar a imagem resultante através de um pedaço aproximado da mesma, para ter uma maior noção das diferenças. Seguem as quatro imagens aproximadas:

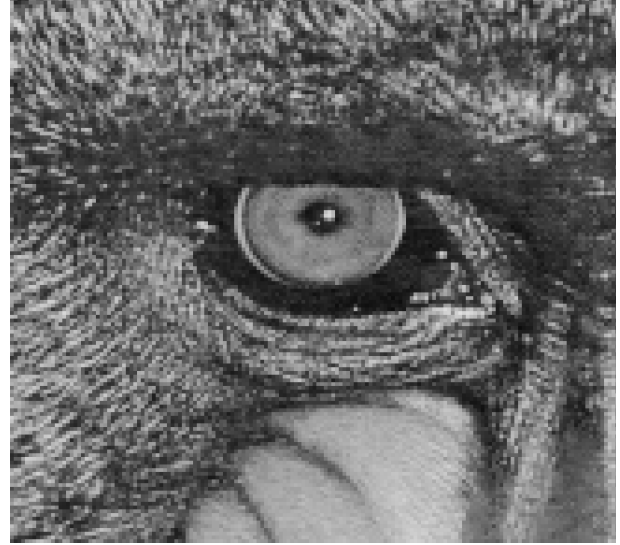


Fig. 5. Imagem aumentada através da técnica de Interpolação pelo vizinho mais próximo.



Fig. 6. Imagem aumentada através da técnica de Interpolação Bilinear.



Fig. 7. Imagem aumentada através da técnica de Interpolação Bicúbica.



Fig. 8. Imagem aumentada através da técnica de Interpolação por Polinômios de Lagrange.

Podemos observar diferenças bem notáveis entre as quatro técnicas aplicadas. Primeiramente, através da utilização da interpolação pelo vizinho mais próximo, imagem, 5, a imagem contém pixels muito serrilhados, bem observados na região dos olhos por exemplo, efeito aliasing. Com a aplicação da Interpolação Bilinear, na Imagem 6, podemos perceber uma boa melhora na qualidade quando referido ao serrilhamento. Através desta técnica também podemos perceber um pouco de borrimento, visto que está sendo realizado a média entre uma vizinhança de quatro pixels, assim como filtros Gaussianos.

Com a aplicação da Interpolação Bicúbica, aplicada na imagem 7, tem-se uma intensificação do efeito obtido através da bilinear, uma maior suavização da imagem e das curvas

existentes, porém o efeito de borrimento também é ainda maior, visto que está sendo realizado uma média com uma vizinhança 4x4. Por último, ao utilizar a Interpolação por Polinômios de Lagrange, podemos perceber uma suavização no serrilhado e ao mesmo tempo uma redução no borrimento da imagem. Acredita-se que este método é um dos mais eficientes entre os testados neste trabalho.

### B. Rotação

O mesmo estudo realizado para a escala das imagens pode ser realizado. Para isto, demonstrando primeiramente a validade dos métodos quanto a capacidade de rotação.

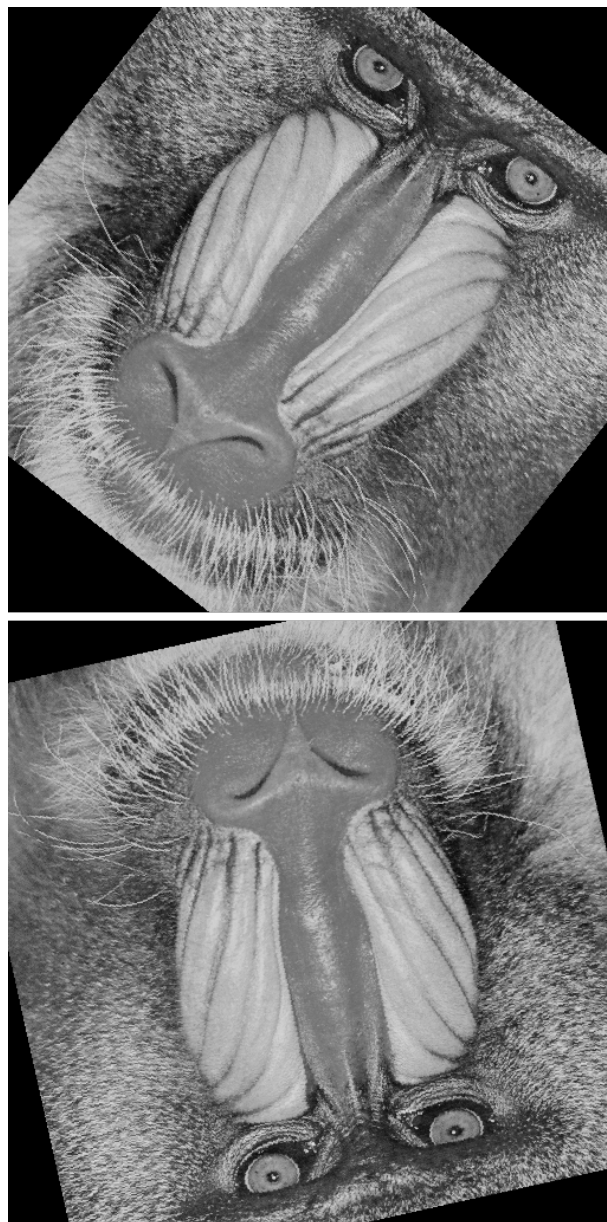


Fig. 9. Imagens resultantes com diferentes rotações.

A Imagem 9, representa dois ângulos diferentes de rotação para a mesma imagem, a primeira com um ângulo de  $37.897^\circ$  e a segunda com ângulo de  $167.478^\circ$ . As partes em pixels que não representavam nenhuma informação, devido a rotação da imagem foram mantidas como pixels pretos.

Também podemos avaliar o resultados quanto a qualidade da imagem resultante da interpolação:



Fig. 10. Imagem aumentada através da técnica de Interpolação pelo vizinho mais próximo.



Fig. 11. Imagem aumentada através da técnica de Interpolação Bilinear.



Fig. 12. Imagem aumentada através da técnica de Interpolação Bicúbica.



Fig. 13. Imagem aumentada através da técnica de Interpolação por Polinômios de Lagrange.

Quanto a interpolação das imagens, os resultados obtidos são muito parecidos com os resultados demonstrados anteriormente através das operações de escala. Onde, imagens de Interpolação por vizinhos mais próximos, Imagem 10, apresentam um maior serrilhado, imagens por Interpolação bilinear, Imagem 11 uma maior suavização e borramento. Métodos de Interpolação Bicúbica, apresentam uma suavização ainda maior, assim como o borramento apresentado, Imagem 12 e por último, métodos de Interpolação por Polinômios de

TABLE I  
TEMPO DE EXECUÇÃO DAS TRANSFORMAÇÕES E MÉTODOS DE  
INTERPOLAÇÃO

Transformação	Método	Tempo (s)
Escala	Vizinho mais Próximo	9.2
	Bilinear	65
	Bicúbica	490
	Polinômios de Lagrange	270
Rotação	Vizinho mais Próximo	1.1
	Bilinear	2.6
	Bicúbica	16
	Polinômios de Lagrange	8.9

Lagrange apresentam uma imagem sem serrilhados e com menor borramento.

### C. Tempo de Execução

Em relação aos tempos de execução de cada uma das transformações geométricas e cada um dos métodos de interpolação, podemos analisar a seguinte tabela, I.

Observou-se que métodos mais simples, têm um pouco tempo de execução, visto que necessitam de menos operações e métodos mais complexos mais tempo, com execução do método de polinômios de lagrange que têm tempos menores que os métodos de interpolação bicúbicas, devido a grande necessidade de cálculos que são necessários para realizar a média da vizinhança 4x4.

Também é possível observar que tempos de rotações são mais similares visto que têm uma média bem aproximada independente do ângulo escolhido. Este comportamento não pode ser observado para técnicas de escala, visto que quanto maior a imagem, maior a área que deve ser preenchida e calculada, visto que os algoritmos têm uma complexidade dependentes do número de pixels da imagem. No exemplos da tabela acima, I, os tempos mensurados são para uma escala de 5.

## V. CONCLUSÕES

Neste trabalho foram implementados e estudados métodos de transformação geométrica e quatro diferentes modos de interpolação para as imagens geradas. Entre eles, métodos de escala e rotação com as seguintes Interpolações disponíveis, Interpolação do vizinho mais próximo, Interpolação Bilinear, Interpolação Bicúbica e Interpolação por Polinômios de Lagrange. Todos estes métodos aceitando valores reais, o que normalmente não é aceito por bibliotecas convencionais.

Foram observados quais as diferenças entre estes métodos tanto em qualidade da imagem gerada quando em relação ao tempo de execução para cada método. Observou-se que para métodos que consideram menos parâmetros como os do vizinho mais próximos, têm-se menos tempo porém piores qualidades de imagem finais. Também observou-se que alguns métodos tendem a borrar um pouco a imagem, devido a média realizada, quando maior a vizinhança maior o borramento experimentado.