

SCC 5830 - Processamento de Imagens (2018)

Projeto Final

Relatório Final

Alysson Alexander Naves Silva

anaves@gmail.com nº USP: 6109930

Aluno especial para o doutorado em Ciência da Computação e Matemática Computacional
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo
ICMC-USP, São Carlos - SP

Projeto Final da disciplina SCC 5830 - Processamento de Imagens ministrada pelo Prof. Dr. Moacir Antonelli Ponti no 1º semestre de 2018.

1. GADraw - Redesenhando Imagens através de um Algoritmo Genético

Este projeto foi desenvolvido na linguagem de programação Java com o uso da biblioteca OpenCV.

O código fonte pode ser encontrado no repositório Github disponível no link:
<https://github.com/anaves/GADraw>

2. Área do projeto

O projeto final se enquadra na área de *recuperação de imagens baseada em conteúdo*, especificamente na reconstrução de imagens através de um algoritmo genético que será guiado pela imagem de referência.

3. Objetivo do projeto

Dada uma imagem de referência, o algoritmo genético irá “redesenhar” imagens que serão avaliadas e comparadas com a referência. O objetivo deste projeto é utilizar técnicas de processamento das imagens para aproximar o resultado obtido com a imagem de referência através de um processo evolutivo.

4. Descrição do problema

Cada pixel da imagem possui 3 camadas (R-G-B) que podem assumir 256 valores diferentes:

Camada	R	G	B
Valores	0-255	0-255	0-255
Possíveis valores	256	× 256	× 256 = 256 ³

Isto daria $256^3 = 16.777.216$ combinações de valores RGB por pixel. Considerando ainda as dimensões da imagem o espaço de busca seria ainda maior.

5. Solução

Nesta seção é descrita as etapas que compõe a solução do problema.

1. Receber como parâmetro uma imagem RGB 24 bits como referência (R);
2. Gerar um conjunto (população) com n possíveis soluções (indivíduos). Cada indivíduo representa uma imagens compostas por m círculos, onde n e m são números naturais;

- (a) Cada círculo é centrado numa coordenada (x, y) , transparência e raio definidos de forma aleatória e cor escura;
3. Passo evolutivo:
- (a) Avaliar a população comparando cada indivíduo com a imagem de referência. O cálculo da distância é feito para cada camada de cores RGB seguindo a equação:
- $$distancia = \sum_i \sum_j |g(i, j) - R(i, j)| \quad (1)$$
- (b) Selecionar os indivíduos (imagens) que passarão suas características para a próxima geração (iteração) através do operador genético de cruzamento;
- (c) Combinar (através do cruzamento) os indivíduos selecionados e gerar novos indivíduos (filhos) mesclando as características dos pais (indivíduos) selecionados na etapa (b);
- (d) Atualizar a população, eliminando os indivíduos mais distantes da imagem de referência e substituindo-os pelos filhos gerados na etapa (c) e repetir os passos a partir do item 3 até que o critério de parada seja satisfeito.
4. Pós-processamento: aplicar métodos disponíveis no OpenCV
- (a) Aplicar o algoritmo de remoção de ruídos NLMeansDenoising
- (b) Aplicar filtros Gaussian e Median
5. Saída: Retornar a imagem que mais se aproximou da referência R .

A Figura 1 apresenta o fluxograma que representa as etapas descritas anteriormente.

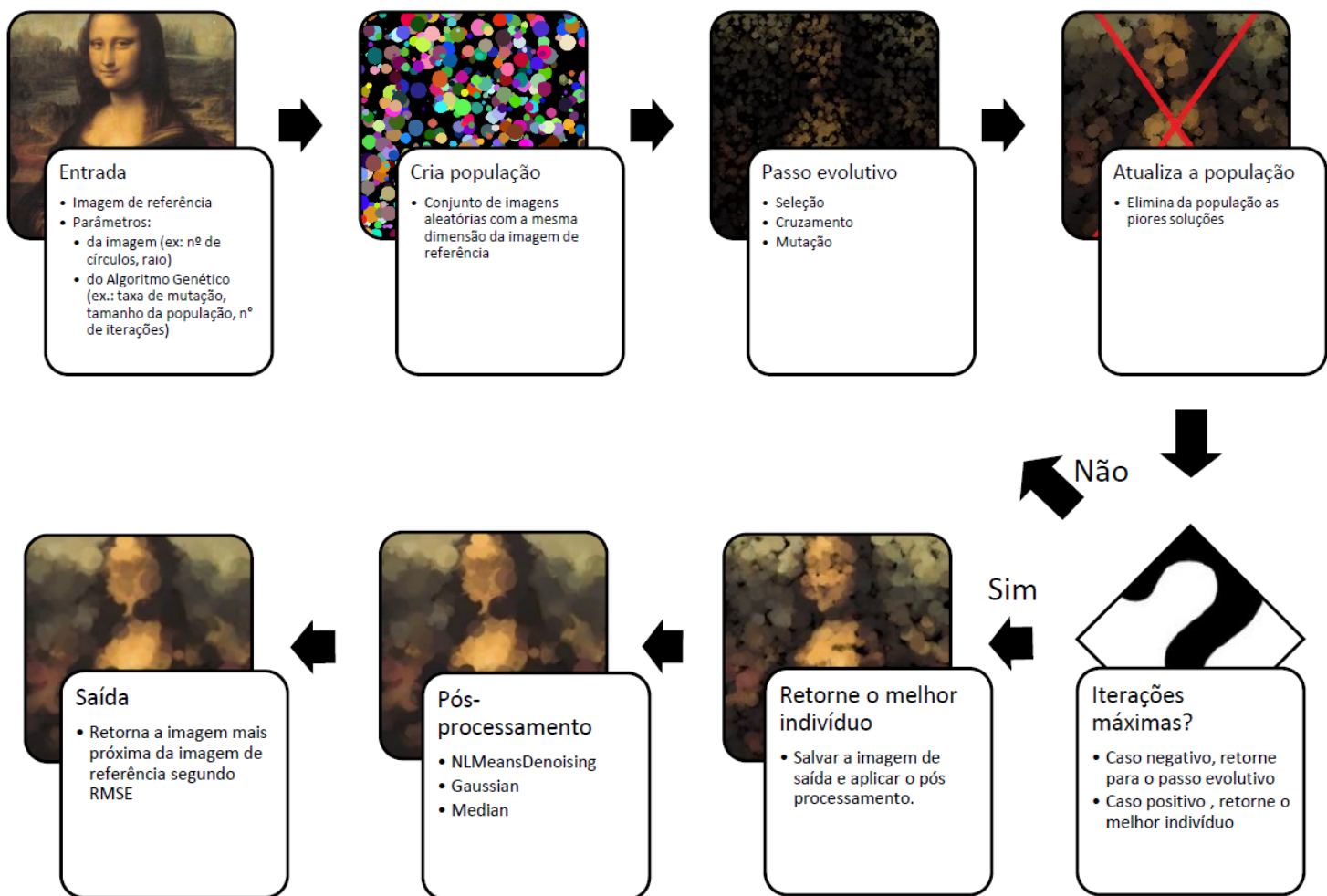


Figura 1. Etapas de execução do algoritmo.

6. Sobre o código

O código foi desenvolvido em Java com o uso da biblioteca OpenCV no ambiente de programação Eclipse. Para execução, basta configurar os parâmetros no arquivo input.txt na pasta input no projeto GADraw.

Os parâmetros para configuração são:

- caminho do arquivo no diretório images
- número de círculos
- raio máximo dos círculos
- taxa de mutação
- tamanho da população
- número de iterações

O arquivo input.txt deve iniciar com a palavra: parametros. Os resultados são demonstrados na seção 7.

7. Resultados

Nesta seção são demonstrados alguns resultados obtidos, os demais podem ser visualizados no GitHub.

Imagem de Mona Lisa Pintura de Leonardo da Vinci

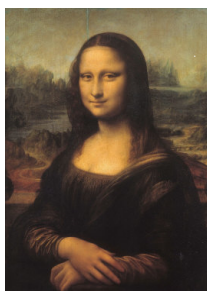


Figura 2. Imagem RGB de referência.

A imagem da Mona Lisa foi obtida na internet no link:

<http://cache2.artprintimages.com/LRG/30/3023/VNZBF00Z.jpg>

Imagem redesenhada.

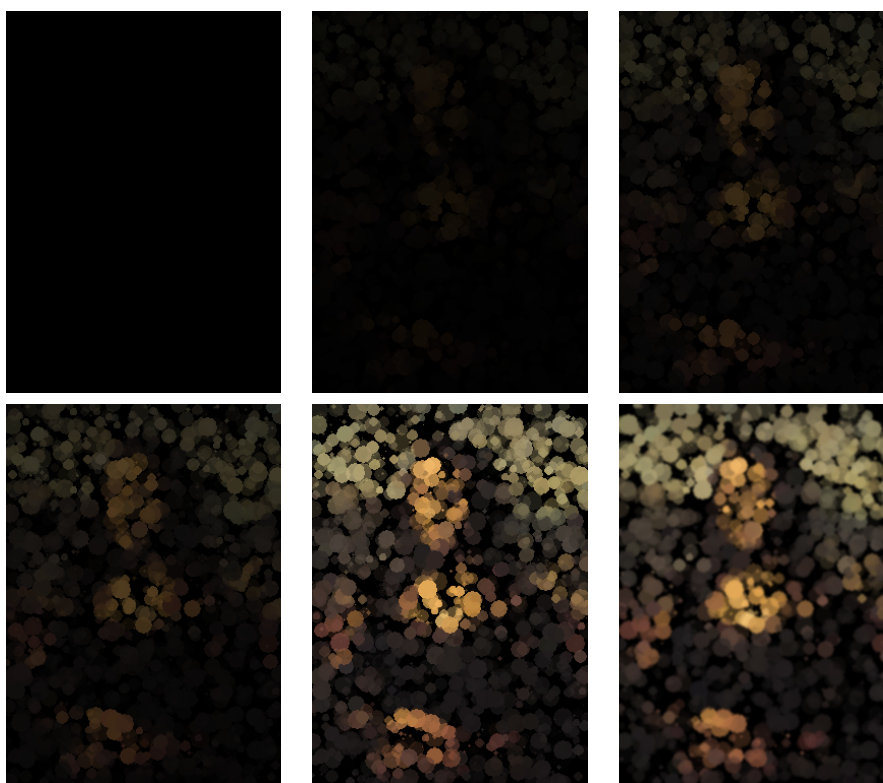


Figura 3. Arara redesenhada com 2000 círculos com raio máximo de 10 pixels.

Imagem de uma arara.

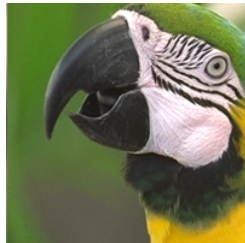


Figura 4. Imagem RGB de referência.

A imagem da arara foi obtida na internet no link:

<http://animais.culturamix.com/blog/wp-content/gallery/companheiro-animal-4/companheiro-animal-11.jpg>

Imagem redesenhada.

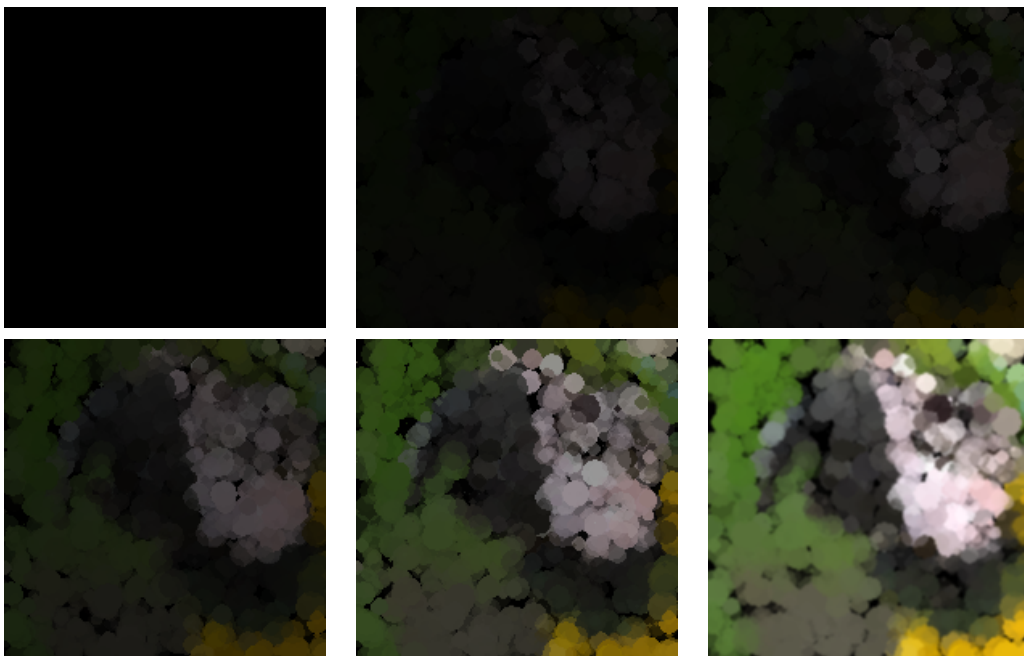


Figura 5. Arara redesenhada com 2000 círculos com raio máximo de 10 pixels.

8. Considerações

Apesar da baixa qualidade das imagens resultantes, a estratégia implementada foi capaz de aproximar da forma da imagem de referência dada a complexidade do problema com um espaço de busca em torno de $n*m*256^3$, onde n e m são as dimensões da imagem.

A convergência do Algoritmo Genético para esta aplicação foi muito lenta, pretende-se melhorar a estratégia de busca no espaço de estados e aplicar outras heurísticas, bem como utilizar filtros espaciais e operadores morfológicos como operadores genéticos de cruzamento e mutação.