

Aproximando Tons de Cinza Utilizando Mínimos Quadrados

Jovani Oliveira Brasil, *PUCRS*

I. INTRODUÇÃO

Uma imagem é basicamente uma matriz de pontos, onde cada ponto possui alguns valores numéricos que definem a sua cor. Aqui no nosso caso intensidade de cinza, pois possuímos uma matriz de tons de cinza. Poderíamos representar uma imagem como uma lista, um grafo, ou também, como veremos aqui, utilizando polinômios.

Neste trabalho, vamos, a partir de uma imagem de entrada, utilizar o método de ajuste dos mínimos quadrados para gerar polinômios capazes de aproximar a própria imagem. Para atingir os resultados esperados foram necessárias duas etapas. A primeira, foi a parte teórica onde buscou-se entender o método e suas aplicações. E a segunda, onde foi realizado o desenvolvimento da aplicação capaz de executar o método e gerar os resultados.

II. UM POUCO DA TEORIA

Considerando uma função $f(x, y)$ que relaciona as coordenadas x e y com as cores (em tons de cinza) de uma imagem, a ideia é utilizar o método de ajuste dos mínimos quadrados para encontrar os coeficientes de uma função $\phi(x, y)$ que aproxima $f(x, y)$. O método dos mínimos quadrados é capaz de gerar uma boa aproximação de $f(x)$, para isso, o método busca que a função aproximadora minimize a soma dos quadrados das diferenças entre os valores referência e os valores gerados pela função aproximada.

Basicamente o que vamos fazer neste trabalho é separar a imagem em quadrantes e encontrar $\phi(x, y)$ que representa as intensidades de cinza em cada um deles. Para mais informações sobre o método de ajuste dos mínimos quadrados pode ser consultado [1].

III. DESENVOLVIMENTO

Como estamos trabalhando em um domínio discreto, significa que vamos utilizar valores tabelados como entrada. Esta tabela é composta pelas colunas de posição (x, y) e intensidade de cinza $f((x, y))$. Estes dados são extraídos da imagem de entrada. Para execução do método precisamos basicamente desta tabela de dados e de uma função para ajuste. A figura 1 demonstra mais claramente a fluxo da solução gerada.

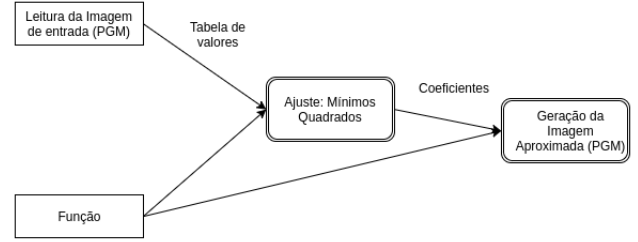


Fig. 1. Diagrama do fluxo da solução implementada para o problema.

Aplicado o método, nós temos os coeficientes da função e então podemos gerar o resultado final. Utilizando a função $\phi(x, y)$ com os coeficientes gerados, geramos os tons de cinza para cada ponto (x, y) e então geramos uma imagem aproximada.

A. Resultados obtidos

A aplicação construída foi testada com algumas imagens de diferentes tamanhos e qualidades. Veja o exemplo do C3PO na figura 1. Observe que, conforme o aumento de quadrantes da imagem, existe uma melhora a qualidade da imagem gerada.

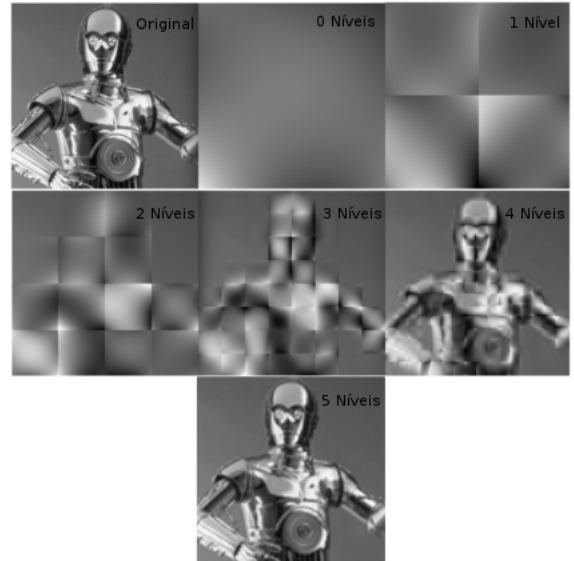


Fig. 2. Retrato do C3PO gerado a partir de funções aproximadas, com diferentes níveis de subdivisão. A quantidade de quadrantes em um nível n é $(2^n)^2$.

Com a imagem sub-dividida em diversos quadrantes atingimos uma melhor aproximação devido ao fator de que o método aplicado a quadrantes menores alcança aproximações melhores de coeficientes, dado que cada quadrante possui coeficientes específicos para o polinômio gerado que generalizam melhor o espaço da imagem.

Também foram gerados aproximações para o retrato de *Monalisa* de *Leonardo da Vinci* e *The Criation of Adam* de *Michelangelo*. Os resultados podem ser conferidos nos diretórios específicos de cada obra no pacote deste trabalho.

B. Função escolhida

Não existe uma regra específica para a escolha da função utilizada para ajuste. De forma geral, para a modelagem de um problema é necessário analisar o domínio do problema que está sendo resolvido e escolher o tipo de função com comportamento mais próximo ao esperado. Segundo [1], a escolha das funções pode ser feita observando o gráfico dos pontos tabelados ou baseando-se em fundamentos teóricos do experimento que forneceu a tabela de valores. Aqui no nosso caso, pela complexidade de se fazer essa análise, optou-se por utilizar a função não linear disponibilizada na especificação.

$$f(x, y) = ax^2y^2 + bxy^2 + cy^2 + dx^2y + exy + fy + gx^2 + hx + i \quad (1)$$

C. Considerações Sobre a Implementação

Foram desenvolvidos três módulos em Python para implementação da solução do problema proposto:

- *ConvertToPGM.py* - Recebe uma imagem qualquer e a converte para PGM.
- *FitLqm.py* - Recebe uma imagem PGM e a partir da função definida como um método em *Python*, aplica método numérico de ajuste. O resultado é um arquivo com coeficientes.
- *Viz.py* - Recebe o arquivo de coeficientes e a partir da função implementada como método em *Python*, gera a imagem com os tons de cinza aproximados. O método mostra as imagens original e aproximada lado a lado.

Algumas observações:

- A funcionalidade de apresentação das imagens utiliza o próprio software disponível no sistema operacional em uso.
- Foram utilizadas as bibliotecas *Scipy* e *PIL* para realização do trabalho.
- A aplicação foi desenvolvida no sistemas operacional Linux, e eu não garanto funcionamento em outros sistemas.

IV. CONCLUSÃO

A subdivisão da imagem em diferentes quadrantes permite a geração de aproximações de melhor qualidade. Isso se dá devido a menor quantidade de dados disponíveis para o processo de ajuste. O segundo fator determinante para a qualidade é a escolha da função inicial. Para uma melhor

aplicação do método, ainda é necessário por parte do aluno um maior entendimento em relação ao processo de escolha da melhor função.

REFERENCES

- [1] M. Ruggiero, *Cálculo Numérico*, 2nd ed. São Paulo, Brasil: PERSON, 1988.