Ampliação de Imagens Utilizando Interpolação de Newton

Universidade do Estado do Rio de Janeiro

Disciplina: Álgebra Linear: Aspectos Teóricos e Computacionais

Autor: Matheus Henrique Branco Zeitune

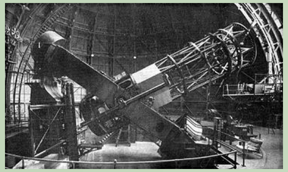
Orientador: Americo Barbosa da Cunha Junior

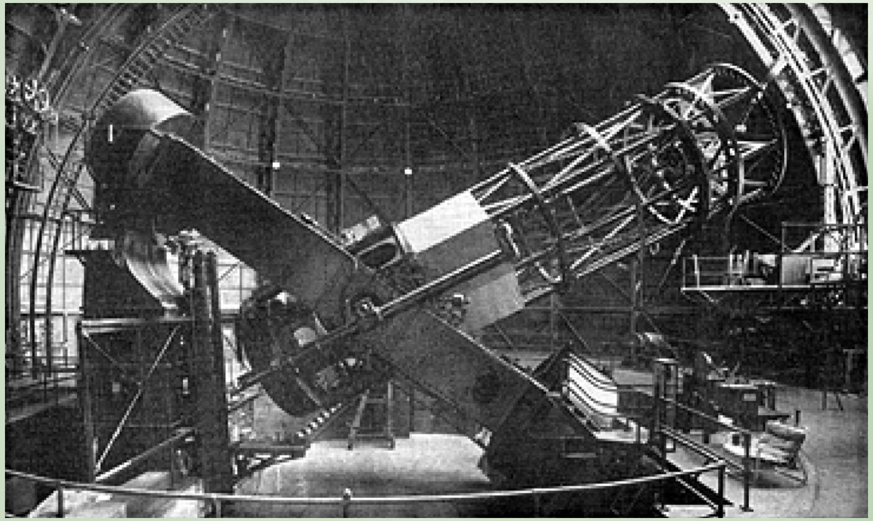
Ampliação de Imagens Utilizando Interpolação de Newton

Lei de Moore e impacto na resolução de imagens digitais

Digitalização de imagens antigas e os desafios de redimensionamento

Utilizar a Interpolação de Newton para ampliar imagens mantendo proporções e qualidade





Ampliação de Imagens Utilizando Interpolação de Newton

Representação de imagens como matrizes bidimensionais

Valores de 0 a 255 representando intensidade luminosa

Modelo RGB com três canais (R, G, B)

$$I = \begin{bmatrix} I_{1,1} & \cdots & I_{1,n} \\ \vdots & \ddots & \vdots \\ I_{m,1} & \cdots & I_{m,n} \end{bmatrix}$$

$$I_{RGB} = \begin{cases} R = \begin{bmatrix} R_{1,1} & \cdots & R_{1,n} \\ \vdots & \ddots & \vdots \\ R_{m,1} & \cdots & R_{m,n} \end{bmatrix} \\ G = \begin{bmatrix} G_{1,1} & \cdots & G_{1,n} \\ \vdots & \ddots & \vdots \\ G_{m,1} & \cdots & G_{m,n} \end{bmatrix} \\ B = \begin{bmatrix} B_{1,1} & \cdots & B_{1,n} \\ \vdots & \ddots & \vdots \\ B_{m,1} & \cdots & B_{m,n} \end{bmatrix} \end{cases}$$

$$I_{RGB} = [R, G, B]$$

Ampliação de Imagens Utilizando Interpolação de Newton

Ampliação: Estima novos pixels para aumentar o tamanho da imagem.

Redução: Remove pixels adjacentes para diminuir o tamanho da imagem.

Ampliação de Imagens Utilizando Interpolação de Newton

Interpolação de Newton

Polinômio interpolador

$$P_n(x) = f[x^0] + (x - x^0) \cdot f[x^0, x^1] + (x - x^0)(x - x^1) \cdot f[x^0, x^1, x^2] + \ldots$$
$$+ (x - x^0)(x - x^1) \ldots (x - x_n{}^{-1}) \cdot f[x^0, x^1, \ldots, x_n]$$

Diferenças Divididas

1. Base inicial:

$$f[x_i] = y_i$$

2. Primeira ordem:

$$f[x_i, x_i{}^{+1}] = \frac{[f[x_i{}^{+1}] - f[x_i]]}{[x_i{}^{+1} - x_i]}$$

3. Segunda ordem:

$$f[x_i, x_i{}^{+1}, x_i{}^{+2}] = \frac{[f[x_i{}^{+1}, x_i{}^{+2}] - f[x_i, x_i{}^{+1}]]}{[x_i{}^{+2} - x_i]}$$

4. Ordem geral:

$$f[x_i, x_i{}^{+1}, \ldots, x_i{}^{+}{}_{k}] = \frac{[f[x_i{}^{+1}, \ldots, x_i{}^{+}{}_{k}] - f[x_i, \ldots, x_i{}^{+}{}_{k}{}^{-1}]]}{[x_i{}^{+}{}_{k} - x_i]}$$

Ampliação de Imagens Utilizando Interpolação de Newton

Aplicação da Interpolação de Newton em duas variáveis

$$P(x,y) = \sum_{i=0}^{n}\sum_{j=0}^{m} f[x_i, y_i] \prod_{k=0}^{i-1}(x - x_k) \prod_{l=0}^{j-1}(y - y_l)$$

Estimativa de intensidade no ponto (0.5, 0.5).

Coordenada	Valor do Pixel
(0, 0)	$I_{0,0}$
(0, 1)	$I_{0,1}$
(1, 0)	$I_{1,0}$
(1, 1)	$I_{1,1}$

1. Diferenças Divididas em x:

$$f[x_0] = I_{0,0}$$

$$f[x_1] = I_{1,0}$$

$$f[x_0, x_1] = \frac{I_{1,0} - I_{0,0}}{x_1 - x_0}$$

2. Diferenças Divididas em y:

$$f[y_0] = I_{0,0}$$

$$f[y_1] = I_{1,0}$$

$$f[y, y_1] = \frac{I_{1,0} - I_{0,0}}{y_1 - y_0}$$

3. Construção do Polinômio Interpolador:

$$P(x,y) = f[x_0, y_0] + (x - x_0) f[x_0, x_1] + (y - y_0) f[y_0, y_1] + (x - x_0)(y - y_0) f[x_0, x_1, y_0, y_1]$$

4. Estimativa do Valor Interpolado:

$$I_{0.5,0.5} = P(0.5, 0.5)$$

Ampliação de Imagens Utilizando Interpolação de Newton

```
function upscale_image_newton_optimized_jpg(input_image, scale_factor)

    original_image = imread(input_image);
    original_image = double(original_image);

    if ndims(original_image) == 3

        red_channel = original_image(:, :, 1);
        green_channel = original_image(:, :, 2);
        blue_channel = original_image(:, :, 3);

        upscaled_red = process_channel(red_channel, scale_factor);
        upscaled_green = process_channel(green_channel, scale_factor);
        upscaled_blue = process_channel(blue_channel, scale_factor);

        upscaled_image = cat(3, upscaled_red, upscaled_green, upscaled_blue);
    else
        upscaled_image = process_channel(original_image, scale_factor);
    end

    imwrite(uint8(upscaled_image), 'upscaled_image_newton_optimized.jpg', 'Quality', 95);
    imshow(uint8(upscaled_image));
end
```

Ampliação de Imagens Utilizando Interpolação de Newton

```
function upscaled_channel = process_channel(channel, scale_factor)

    [rows, cols] = size(channel);

    new_rows = round(rows * scale_factor);
    new_cols = round(cols * scale_factor);

    x = 1:rows;
    y = 1:cols;

    xi = linspace(1, rows, new_rows);
    yi = linspace(1, cols, new_cols);

    upscaled_channel = zeros(new_rows, new_cols);

    for i = 1:new_rows
        for j = 1:new_cols
            row_idx = max(1, floor(xi(i)) - 1):min(rows, ceil(xi(i)) + 1);
            col_idx = max(1, floor(yi(j)) - 1):min(cols, ceil(yi(j)) + 1);

            local_values = channel(row_idx, col_idx);
            local_x = x(row_idx);
            local_y = y(col_idx);

            if length(local_x) < 2 || length(local_y) < 2
                upscaled_channel(i, j) = channel(min(rows, round(xi(i))), min(cols, round(yi(j))));
            else
                temp_values = zeros(1, length(local_y));
                for k = 1:length(local_y)
                    coef_x = local_newton_coefficients(local_x, local_values(:, k)');
                    temp_values(k) = newton_interpolation(local_x, coef_x, xi(i));
                end

                coef_y = local_newton_coefficients(local_y, temp_values);
                upscaled_channel(i, j) = newton_interpolation(local_y, coef_y, yi(j));
            end
        end
    end
end
```

Ampliação de Imagens Utilizando Interpolação de Newton

```
function coef = local_newton_coefficients(xd, yd)
    n = length(xd);
    if n < 2
        coef = yd(1);
        return;
    end
    coef = yd;
    for j = 2:n
        coef(j:n) = (coef(j:n) - coef(j-1:n-1)) ./ (xd(j:n) - xd(1:n-j+1));
    end
end
```

Ampliação de Imagens Utilizando Interpolação de Newton

```
function result = newton_interpolation(xd, coef, x)
    n = length(coef);
    result = coef(n);
    for j = n-1:-1:1
        result = result .* (x - xd(j)) + coef(j);
    end
end
```

Ampliação de Imagens Utilizando Interpolação de Newton

Algoritmo de Peak Signal to Noise Ratio (PSNR)

$$PSNR = 10 \times \log_{10}\left(\frac{MAX^2}{MSE}\right)$$

$$MSE = \frac{1}{m \times n}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(I_{original}(i,j) - I_{alterada}(i,j)\right)^2$$

Acima de 30 db: alta qualidade
Entre 20 db a 30 db: qualidade aceitável
Abaixo de 20 db: baixa qualidade

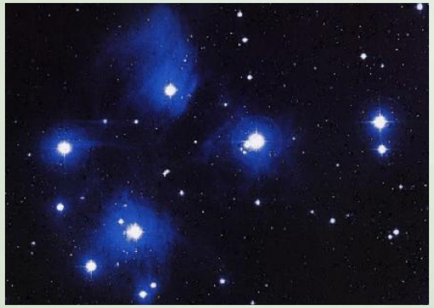
Ampliação de Imagens Utilizando Interpolação de Newton

```
function psnr = calculate_psnr(image_original_path, image_altered_path)

    I_original = imread(image_original_path);
    I_altered = imread(image_altered_path);

    if size(I_original, 3) ~= size(I_altered, 3)
        error("As imagens devem ter o mesmo número de canais (grayscale ou RGB).");
    end

    if size(I_original, 1) ~= size(I_altered, 1) || size(I_original, 2) ~= size(I_altered, 2)
        I_altered = imresize(I_altered, [size(I_original, 1), size(I_original, 2)]);
    end

    I_original = double(I_original);
    I_altered = double(I_altered);

    mse = mean((I_original(:) - I_altered(:)).^2);

    if mse == 0
        psnr = Inf;
        return;
    end

    max_pixel = 255.0;

    psnr = 10 * log10((max_pixel^2) / mse);
end
```

Ampliação de Imagens Utilizando Interpolação de Newton

"The Pleiades Star Cluster" (valores de PSNR para ampliações em 2x, 4x e 10x)

0.095 MP	0.381 MP	1.5 MP	9.5 MP
365x261	730x522	1460x1044	3650x2610
Original	PSNR: 32,94 dB	PSNR: 31,05 dB	PSNR: 33,73 dB



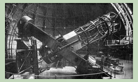
Ampliação de Imagens Utilizando Interpolação de Newton

"The Hooker Telescope on Mt. Wilson" (escala de cinza). Fotocópia da foto original de 1920

0.080 MP

367x219

Original



PSNR: 24,31 dB

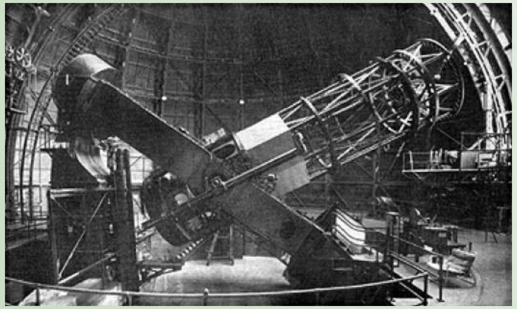
734x438

0.321 MP

1.3 MP

1468x876

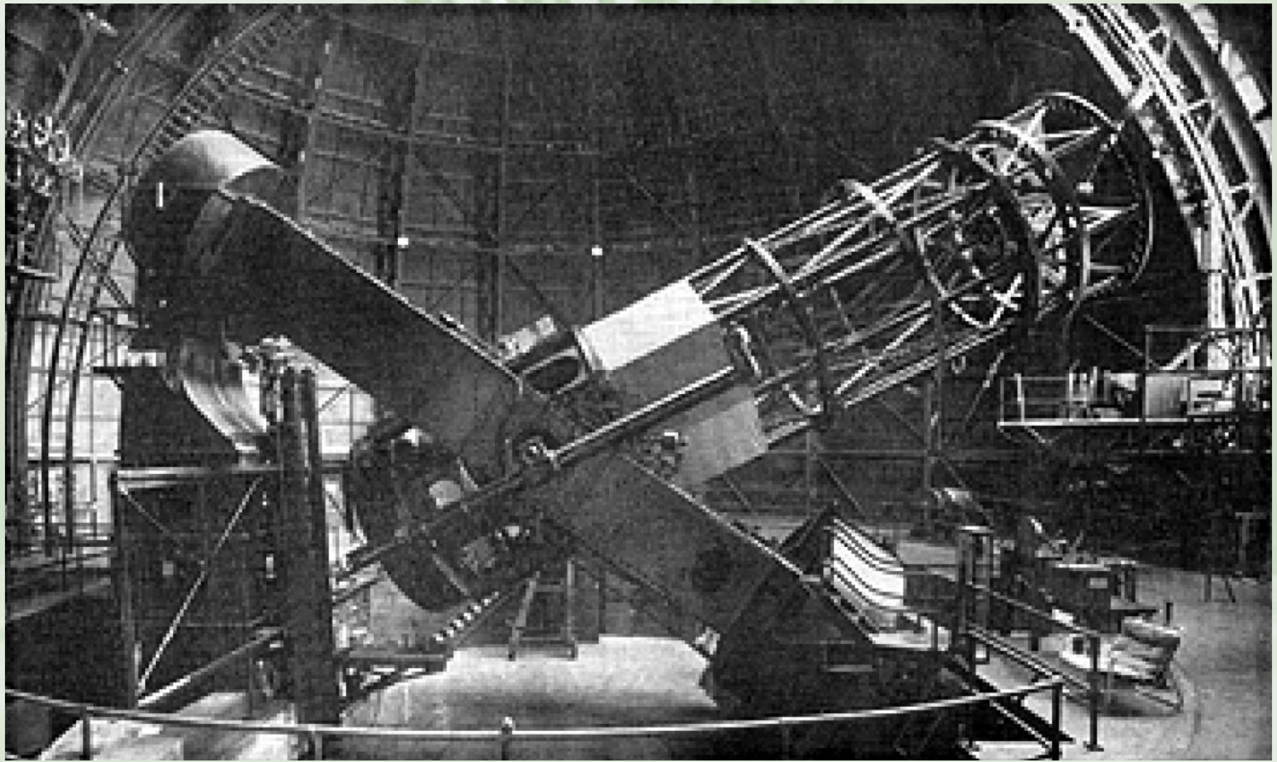
PSNR: 21,67 dB



8.0 MP

3670x2190

PSNR: 24,62 dB



Ampliação de Imagens Utilizando Interpolação de Newton

"Winter and Summer on a Little Planet" (imagem moderna)

1.0 MP

1024x1024

Original

4.2 MP

2048x2048

PSNR: 30,67 dB

16.8 MP

4096x4096

PSNR: 28,69 dB

104.9 MP

10240x10240

PSNR: 31,85 dB

Ampliação de Imagens Utilizando Interpolação de Newton

Interpretação dos resultados

Todas as imagens geradas com PSNR acima de 20 db

Ampliação de Imagens Utilizando Interpolação de Newton

Perguntas e respostas

Ampliação de Imagens Utilizando Interpolação de Newton

Referências

- **USP - Instituto de Ciências Matemáticas e de Computação**
  USP - Instituto de Ciências Matemáticas e de Computação. *Interpolação de Newton*. Disponível em: https://sites.icmc.usp.br/andretta/ensino/aulas/sme0500-1-12/ipnewton.pdf. Acesso em: 30 nov. 2024.
- **UFPR**
  UFPR. *Interpolação de Newton e Erros* [PDF]. Disponível em: https://docs.ufpr.br/~volmir/MN_12_interpolacao_newton_e_erros_ppt.pdf. Acesso em: 30 nov. 2024.
- **MathWorks**
  MATHWORKS. *PSNR – Peak Signal-to-Noise Ratio*. Disponível em: https://www.mathworks.com/help/vision/ref/psnr.html. Acesso em: 30 nov. 2024.
- **Cadence PCB Design**
  CADENCE PCB DESIGN. *Peak Signal-to-Noise Ratio vs Signal-to-Noise Ratio*. 2023. Disponível em: https://resources.pcb.cadence.com/blog/2023-peak-signal-to-noise-ratio-vs-signal-to-noise-ratio. Acesso em: 30 nov. 2024.
- **Mount Wilson Observatory – The Pleiades Star Cluster**
  MOUNT WILSON OBSERVATORY. *The Pleiades Star Cluster*. Disponível em: https://apod.nasa.gov/apod/ap950701.html. Acesso em: 30 nov. 2024.
- **Mount Wilson Observatory – The Hooker Telescope on Mt. Wilson**
  MOUNT WILSON OBSERVATORY. *The Hooker Telescope on Mt. Wilson*. Disponível em: https://apod.nasa.gov/apod/ap950701.html. Acesso em: 30 nov. 2024.
- **Cunha Jr**
  CUNHA JR. *Interpolação*. Universidade do Estado do Rio de Janeiro – UERJ, 2021.
- **Mount Wilson Observatory – Winter and Summer on a Little Planet**
  MOUNT WILSON OBSERVATORY. *Winter and Summer on a Little Planet*. Disponível em: https://apod.nasa.gov/apod/ap241130.html. Acesso em: 30 nov. 2024.
- **Blog do Enem**
  BLOG DO ENEM. *Matrizes: definição, ordem, notação e tipos de matrizes*. Disponível em: https://blogdoenem.com.br/matrizes-definicao-ordem-notacao-e-tipos-de-matrizes/. Acesso em: 30/11/2024.
- **Blog Aspose**
  BLOG ASPOSE. *Create, Load, Fill, and Draw Bitmap in C#*. Disponível em: https://blog.aspose.com/drawing/create-load-fill-and-draw-bitmap-in-csharp/. Acesso em: 30/11/2024.
- **Exif Tools**
  EXIF TOOLS. *Ferramenta para análise de metadados EXIF*. Disponível em: https://exif.tools/. Acesso em: 30 nov. 2024.
- **MZET97 – Upscale**
  MZET97. *Upscale*. Disponível em: https://github.com/mzet97/Upscale/tree/main. Acesso em: 30 nov. 2024.