

Transferência de cores por *color matching*

Aluno: Mateus Yamada Müller¹

Professor: Gustavo B. Borba ²

¹Departamento Acadêmico de Eletrotécnica (DAELT)

²Departamento Acadêmico de Eletrônica (DAELN)

Universidade Tecnológica Federal do Paraná

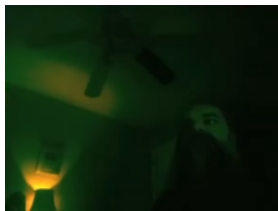
Apresentação do projeto final de disciplina, Fevereiro 2025

Tópicos da apresentação

- 1 Intuição
- 2 Método
- 3 Remapeamento de luminância
- 4 Correspondência entre pixels
- 5 Resultados
- 6 Referências

Intuição

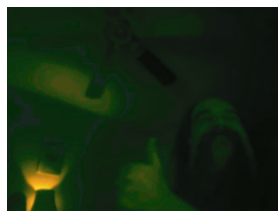
- Baseado no método proposto em T. Welsh, M. Ashikhmin, and K. Mueller, “Transferring color to greyscale images”, 2002 [1];
- Transferência de cores por correspondência entre uma imagem de referência (colorida) e uma objetivo escala de cinza.



Referência



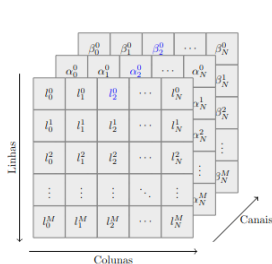
Objetivo



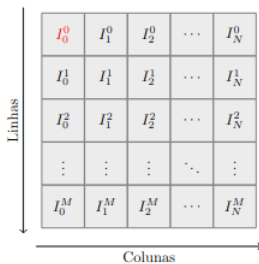
Color matching

Intuição

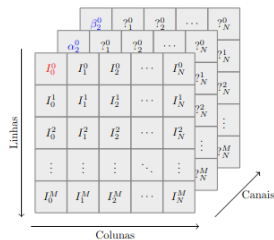
- Operação pixel a pixel de busca por correspondência entre a luminância das imagens;
- Expansão da imagem objetivo pela adição da matiz e da saturação.



Referência



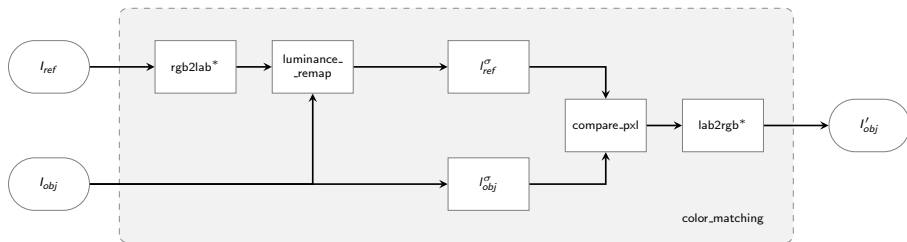
Objetivo



Color matching

Método

- Duas imagens de entrada e a imagem colorida de saída;
- Implementação em MATLAB como uma biblioteca de funções.



Remapeamento de luminância

- Melhor correspondência entre os histogramas [2];

$$I_{ref}[i,j] = \frac{\sigma_{obj}}{\sigma_{ref}}(I_{ref}[i,j] - \mu_{ref}) + \mu_{obj}$$

```
function ref_remap = luminance_remapping(img_ref_lum, img_tar_lum)
    mu_ref = mean(img_ref_lum(:)); %media da luminancia ref
    sigma_ref = std(img_ref_lum(:)); %desvio padrao ref
    mu_tar = mean(img_tar_lum(:)); %media da luminancia gs
    sigma_tar = std(img_tar_lum(:)); %desvio padrao ref
    ref_remap = (sigma_tar/sigma_ref)*(img_ref_lum-mu_ref) + mu_tar;
end
```

Remapeamento de luminância

Objetivo



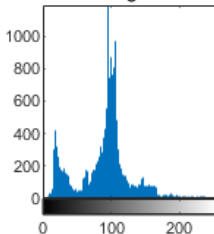
Sem remapeamento



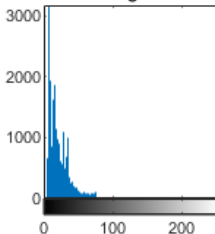
Com remapeamento



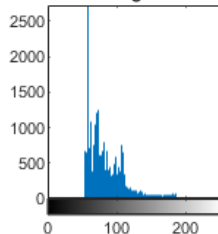
Histograma



Histograma

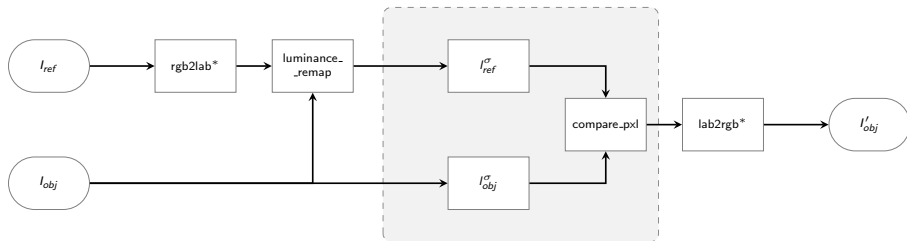


Histograma



Correspondência entre pixels

- Comparação entre o pixel atual da varredura e um conjunto de pixels de luminância da imagem de referência;
- Luminância e estatísticas de vizinhança de pixels.



Correspondência entre pixels

Estatística de vizinhança:

- Desvio padrão com *kernel* 5x5 [3]

$$I_{img}^{\sigma}[i,j] = \sigma_{\{(u,v)|u,v \in R\}} \{I[i+u, j+v]\};$$

```
function std_img = std_filter2D(img, mask_size)
    img = im2double(img);
    [img_zp, pad] = framework.zero_padding(img, mask_size);

    %filtro desvio padrao
    std_img = zeros(size(img)); %a matriz final deve ter o tamanho da original
    %pixels (linha) da imagem original que ser o tratados
    pxl_i = size(img,1); %linha
    pxl_j = size(img,2); %coluna
    for i = 1:pxl_i
        for j = 1:pxl_j
            %pixels na matriz com zero padding que contemplam a operacao atual
            sub_idx_i = i:i+2*pad; %linha
            sub_idx_j = j:j+2*pad; %coluna
            sub_mtx = img_zp(sub_idx_i, sub_idx_j);
            std_img(i, j) = std(sub_mtx(:));
        end
    end
end
```

Correspondência entre pixels

Busca pela melhor correspondência:

- Problema de minimização da soma ponderada;
- Assumindo um pixel p da imagem objetivo

$$(i_{match}, j_{match}) = \underset{(i,j)}{\operatorname{argmin}} \{0.5(I_{ref}[i,j] - I_{obj}[i_p, j_p])^2 + 0.5(I_{ref}^\sigma[i,j] - I_{obj}^\sigma[i_p, j_p])^2\};$$

- **Global color matching:** $\{(i,j) | 1 \leq i \leq M, 1 \leq j \leq N\}$, onde M é o número de linhas e N de colunas de I_{ref} ;
- **Jittered grid color matching:** aproximadamente 200 amostras aleatórias da imagem de referência.

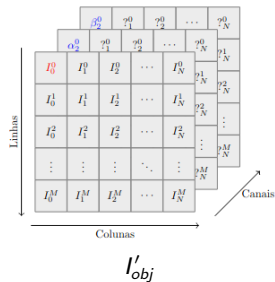
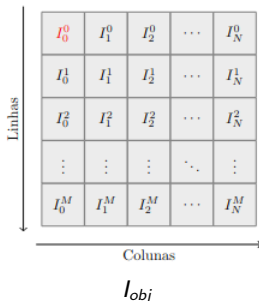
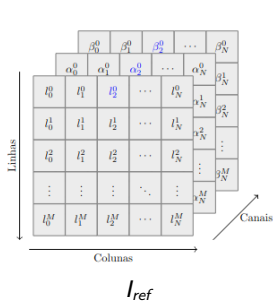
```
function pxl_match = compare_pxl(target_mat, ref_mat)
    [ref_i, ~] = size(ref_mat); %linhas e colunas da imagem ref
    diff_mat = repmat(target_mat, ref_i, 1); %vetor do pixel i*j vezes
    l2_norm = (diff_mat - ref_mat).^2; %norma L2
    weight_sum = 0.5*l2_norm(:,1) + 0.5*l2_norm(:,2); %soma ponderada
    [~, pxl_match] = min(weight_sum); %busca o índice de menor valor
end
```

Correspondência entre pixels

Expansão da matriz de I_{obj} :

$$I'_{obj}[i_p, j_p, 2 : 3] = I_{ref}[i_{match}, j_{match}, 2 : 3],$$

$$I'_{obj}[i_p, j_p, 1] = I_{obj}[i_p, j_p, 1].$$



Resultados



Referência



Objetivo



Global



Jittered grid

Resultados



Referência



Objetivo



Global



Jittered grid

Resultados

Desempenho dos métodos:

Figura	Método	Tempo (s)
Folhagens	<i>global</i>	4,222
	<i>jittered grid</i>	0,435
Landsat 7	<i>global</i>	27,849
	<i>jittered grid</i>	0,657

Eficiência do *jittered grid*:

- 89.27% no caso 1;
- 97.64% no caso 2;

Resultados



Referência



Objetivo

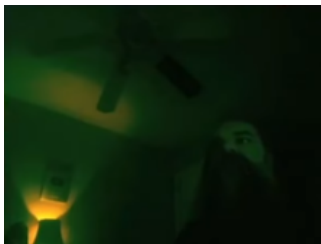


Global



Jittered grid

Resultados



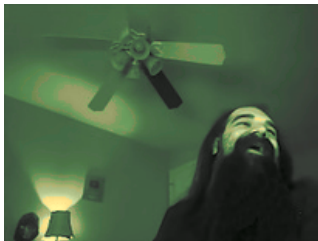
Referência



Objetivo



Sem remapeamento

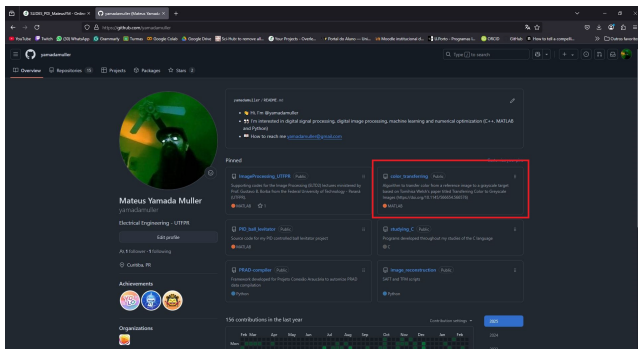


Com remapeamento

Repositório

Implementação disponível em :

https://github.com/yamadamuller/color_transferring



Referências I



T. Welsh, M. Ashikhmin, and K. Mueller, “Transferring color to greyscale images,” *ACM Transactions on Graphics*, pp. 277–280, July 2002.



A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curlezz, and D. H. Salesin, “Image analogies,” in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 327–340, August 2001.



W. Burger and M. J. Burger, *Digital Image Processing: An Algorithmic Introduction Using Java*, vol. 2 of *Texts in Computer Science*.

London: Springer, 2016.