

Análise Técnica e Implementação de Equalização de Histograma: Conceitos, Exemplos e Passo a Passo em C e SDL

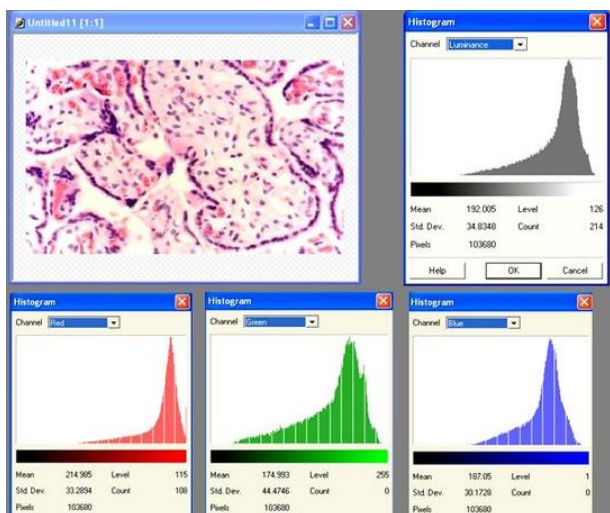
Introdução

Esse artigo tem como finalidade apresentar um tutorial simples e direto da implementação de equalização de histograma, usando C e SDL. Para tal efeito, se destaca como fundamental primeiro discorrer sobre o conceito de histograma, e como ele representa a distribuição de intensidade de pixels em uma imagem. Além disso, é crucial destacar as aplicações práticas dessa técnica.

Essa técnica é relevante em áreas como: Dados em estatísticas, usada em projetos que envolvem aprendizado de máquina, tratando a escala de entrada, o que pode melhorar a performance dos modelos preditivos, pois a irregularidade dos dados pode causar *overfitting*, induzindo o programa a aprender os padrões do ruído; Áudio, para tornar os áudios mais equilibrados e audíveis, eficaz em reconhecimento de fala.

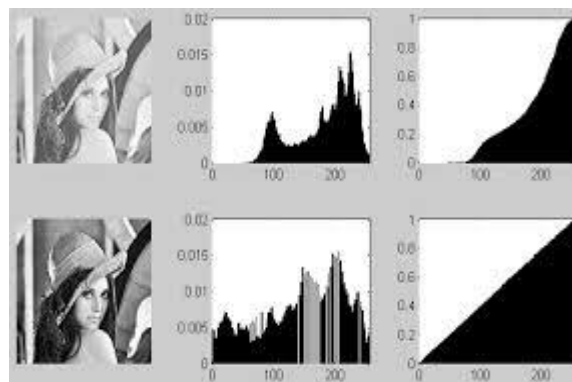
Conceitos Fundamentais

No contexto de imagens, para criar um histograma, é necessário constatar a distribuição de valores nos canais de cores RGB dos pixels da imagem, ou seja, obter a quantidade de cada intensidade de byte, de 0 a 255, em cada canal de cor. Por outro lado, para imagens em *grayscale* (escala de cinza), onde os canais RGB tem valores iguais, apenas um canal é usado.



A partir disso, os valores são mapeados em uma representação gráfica, comumente um gráfico de barras, onde, no eixo X, estão os pontos de 0 a 255 para cada valor de byte, e o eixo Y representa a quantidade de pixels com o valor relativo ao eixo X. Em imagens coloridas, gera-se um histograma para cada canal de cor. Os histogramas resultantes servem de ferramenta importante para interpretar a imagem em um nível técnico, assim como para manipulá-la pela equalização.

A equalização de um histograma é um método de normalização de histograma, ela consiste em redistribuir os valores a fim de diminuir as diferenças. No caso de imagens, trazendo mais contraste e destacando detalhes que antes não eram tão percebidos ao diminuir a predominância em certos tons, realçando os que anteriormente estavam menos perceptíveis, útil também para realçar a visibilidade perdida em zonas de iluminação desiguais. Esse método apresenta relevância notável em diversos campos de estudo além de apenas tratar de imagens.



Implementação

É necessário possuir uma IDE com suporte à compilação, e possuir acesso ao código do SDL e SDL_image, seus binários e *headers*. Isso pode ser feito por uma instalação simples por comandos como no terminal do sistema operacional Linux, ou manualmente com os lançamentos pelo github. Abaixo está o código da implementação da equalização em uma imagem em grayscale:

```

static void histogramEqualization(void) {
    if (!surface)
    {
        SDL_Log("Erro em histogramEqualization(): Imagem inválida!\n");
        return;
    }

    const SDL_PixelFormatDetails* format = SDL_GetPixelFormatDetails(surface->format);
    const size_t pixelCount = surface->w * surface->h;

    Uint32* pixels = (Uint32*)surface->pixels;
    Uinta r = 0;
    Uinta g = 0;
    Uinta b = 0;
    Uinta a = 0;

    if (!checkGrayscale()) { printf("A imagem deve ser cinza.\n"); return; }

    imageHistogram(); //cria os valores de currentHistogram
    double histogram[256] = { 0 };

    k = format->bits_per_pixel / format->bytes_per_pixel;
    L = pow(2, k); // cálculo dos níveis discretos de intensidade L = 2 ^ k

    int M = surface->h;
    int N = surface->w;

    int MXN = M * N;

    double maxValue = -1;

    for (int k = 0; k < 255; k++) { //0 a 255 intensidades
        int soma = 0;

        for (int j = 0; j <= k; j++) //soma de todas as quantidades de intensidades
        {
            //de 0 até o valor K atual
            soma += currentHistogram[j];
        }

        histogram[k] = (L - 1) * soma / (M * N); // T(rk) = (L - 1) * somatória de nj / (M * N);
        if (histogram[k] > maxValue) { maxValue = histogram[k]; }
    }

    //Normalização
    //achar o maior valor no histograma para normalizar os outros com base nele
    for (int i = 0; i < 256; ++i) {
        if (histogram[i] > maxValue) {
            maxValue = histogram[i];
        }
    }

    // Normalizar os valores do histograma para 0-255
    //valorExemplo/valorMax será uma porcentagem do valorMax
    //multiplicado por 255 entra na escala proporcional, com valorMax sendo 255
    for (int i = 0; i < 256; ++i) {
        histogram[i] = (histogram[i] * 255) / maxValue;
    }

    int newHistogram[256] = { 0 }; // Novo histograma já equalizado

    for (size_t i = 0; i < pixelCount; ++i) {
        // obtém o valor RGBA de cada pixel
        SDL_GetRGBA(pixels[i], format, NULL, &r, &g, &b, &a);

        // r == g == b para uma imagem em grayscale, então apenas 'r' é usado
        // Altera o valor 'r' com base no histograma
        // Mapeando r para o novo valor no histograma
        r = g = b = (int)histogram[r]; // R ganha intensidade nova calculada no histograma
        // Como histograma foi usado para o cálculo das intensidades
        // usa-se o novo histograma para obter a distribuição
        // E atualiza o valor do pixel com a nova cor
        pixels[i] = SDL_MapRGBA(format, NULL, r, g, b, a);
        newHistogram[r]++;
    }

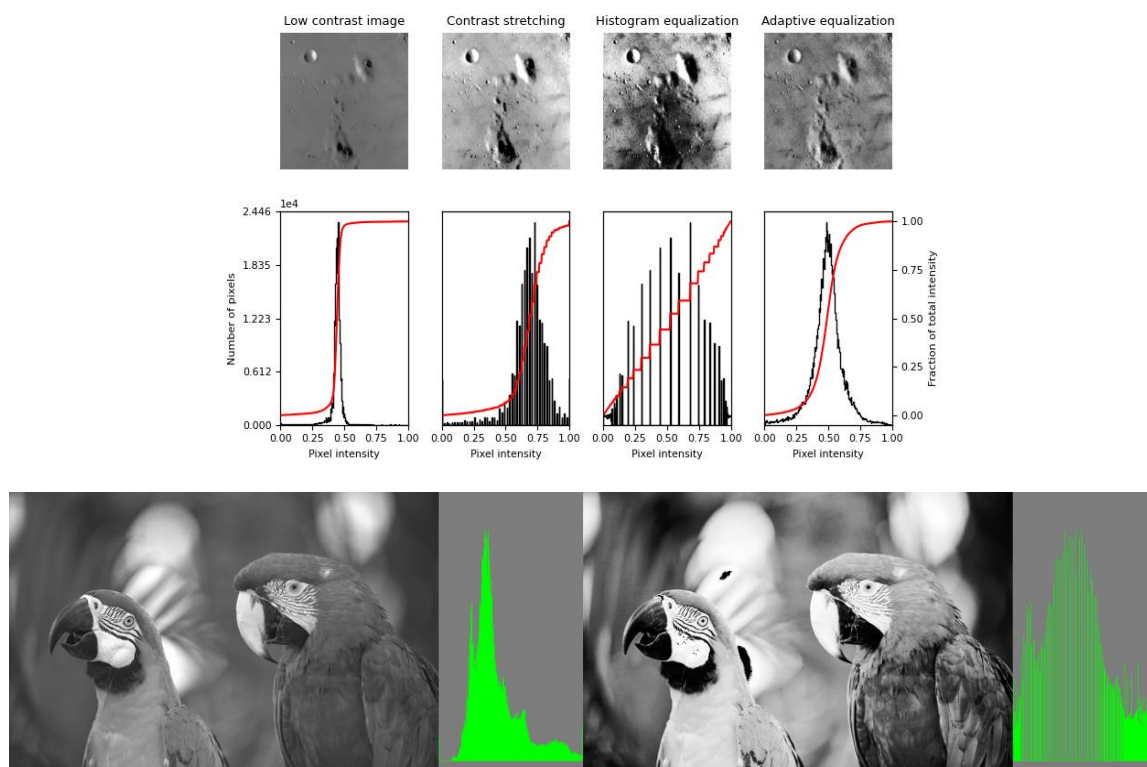
    for (int i = 0; i < 256; ++i) {
        currentHistogram[i] = newHistogram[i];
    }

    SDL_UnlockSurface(surface);
    SDL_DestroyTexture(texture);
    texture = SDL_CreateTextureFromSurface(renderer, surface);
}

```

Resultados

Observamos que a equalização de histograma, pode melhorar significativamente a visibilidade de detalhes em imagens. Todavia, a técnica pode resultar em artefatos indesejados, como o aumento do ruído em regiões antes aparentemente homogêneas, devido ao aumento de contraste. Para aprimorar os resultados, métodos como a equalização adaptativa, que ajustaria o contraste por regiões da imagem, ou o uso de técnicas de suavização antes da equalização para minimizar o ruído. Isso é notável na imagem cinza criada com algoritmo de média aritmética (imagem das araras). Algumas áreas claras ficaram escuras, outras áreas aparentam estar muito saturadas.



Conclusão

Este artigo abordou a técnica de equalização de histograma, explicando seu conceito, utilidade, implementação e resultados de modo técnico e direto. Algumas aplicações práticas úteis possíveis seriam: distribuir frequências em músicas para ter mais definição de sons menos percebidos

em dispositivos de áudio mais baratos, ou em formatos de músicas comprimidos, assim como em imagens de dados biométricos a fim de auxiliar na percepção de detalhes menos visíveis. Aplicar uma transformação logarítmica na distribuição do histograma pode ser uma variante a ser estudada, pois teria uma distribuição menos uniforme que poderia ser útil em certos casos. Notavelmente, a aplicação dos conceitos e o conhecimento sobre SDL são os maiores desafios.

Referências:

Wikipedia. Histograma. Disponível em: <https://pt.wikipedia.org/wiki/Histograma>.

Wikipedia. Equalização de histograma. Disponível em:
https://pt.wikipedia.org/wiki/Equalização_de_histograma.

Estatisticafacil. O que é histogram normalization guia completo. Disponível em:
<https://estatisticafacil.org/glossario/o-que-e-histogram-normalization-guia-completo/>.

Embarcados. Limpeza dos Dados. Disponível em: <https://embarcados.com.br/limpeza-dos-dados/>.

Researchgate. Histogram Equalization of the Speech Representation for Robust Speech Recognition. Disponível em:
https://www.researchgate.net/publication/220655263_Histogram_Equalization_of_the_Speech_Representation_for_Robust_Speech_Recognition.

Researchgate. Figura 2 Histograma da imagem Imagem digital colorida multibanda pós captura fig2. Disponível em: https://www.researchgate.net/figure/Figura-2-Histograma-da-imagem-Imagem-digital-colorida-multibanda-pos-captura_fig2_265510189.

Facom. Aula04 ProcessamentoHistogramas. Disponível em:
<https://www.facom.ufu.br/~backes/gsi058/Aula04-ProcessamentoHistogramas.pdf>.

SCIKIT-IMAGE. Equalize Histogram. Disponível em: https://scikit-image.org/docs/stable/auto_examples/color_exposure/plot_equalize.html.

O código fonte do exemplo pode ser acessado em: <https://github.com/Ro-m-l/Equalizacao-de-Histograma-em-Grayscale>