Quantização Vetorial de Cores em Imagens Digitais

Leonardo C. Araújo, João Pedro H. Sansão e Sandro A. Fasolo

***Resumo*— Este artigo apresenta um comparativo, utilizando diferentes medidas objetivas de dissimilaridade entre imagens (utilizando ou não características psicovisuais), da quantização vetorial das cores em imagens digitais utilizando-se o algoritmo de Lloyds, *dithering* e diferentes métricas para distância entre cores. Os resultados mostram que cada medida de dissimilaridade entre imagens é minimizada por uma abordagem diferente.**

***Palavras-Chave*— quantização vetorial, cores, dithering, dissimilaridade entre imagens**

***Abstract*— The present paper stablish a comparison among different color vector quantization approaches. Different dissimilarities measures between images are used, using psychovisual features or not. The vector quantization is achieved by Lloyds algorithm and dithering. Different color distance metrics are tested. Results show that each image dissimilarity metric is minimized by a different approach.**

***Keywords*— vector quantization, color, dithering, image dissimilarity**

I. Introdução

Compressão de imagens e vídeos é essencial em aplicações de comunicação, tais como serviços web que utilizam fotos digitais, câmeras digitais, TV digital, streaming de vídeos, etc. Mesmo que os recursos computacionais e capacidade de transmissão de dados ou armazenamento tenham crescido enormemente nos últimos anos, o número de pessoas e a quantidade de acesso a serviços que utilizam codificação de imagens e vídeos vêem crescendo a uma taxa ainda maior, criando um tsunami de dados. Estudos mostram que as novas tecnologias de informação e comunicação irão consumir por volta de 20% de toda energia e ser responsáveis por 5,5% da emissão de gás carbônico em 2025 [1]. Para evitar os efeitos desse crescimento desenfreado do consumo de dados e, consequentemente, de energia, é necessário contrapô-lo com a melhoria da eficiência tecnológica.

Existem várias técnicas utilizadas para codificar a informação e reduzir o custo de transmissão, armazenamento e processamento. Este trabalho foca a utilização da quantização vetorial (VQ) aplicada à codificação de cores em imagens digitais, onde verificar-se-á a utilização da técnica de meio-tom concomitante à quantização. A VQ, aliada às técnicas de meio-tom digital, é fundamental quando o objetivo é reproduzir imagens de cores reais com menor distorção visual possível nos aparelhos incapazes de reproduzir todo espectro de cores visíveis. O problema na representação das cores se agrava quando é necessário representar diversas imagens concomitantemente, utilizando uma paleta de cor restrita.

A quantização vetorial (VQ) veem sendo aplicada há muito tempo na codificação da fala e imagens digitais [2], [3], [4], [5], [6], [7], inclusive na codificação de cores [8], [9]. Exemplos recentes também mostram a aplicação da VQ em reconhecimento de fala, mesmo em condições ruidosas e com requisitos de baixa complexidade [10], [11]. Ainda, a VQ pode ser utilizada no projeto de modelos de mistura de Gaussianas, utilizando a abordagem de Lloyd para criar um classificador robusto [12]. Em visão computacional a VQ é usada como uma ferramenta para construir representações de imagem em nível superior, sendo utilizada para se obter 'modelos de mundo' [13], [14].

Aplicar a VQ sobre as cores de uma imagem implica na aceitação de uma certa degradação da informação visual, imposta pela restrição quanto ao número de cores simultaneamente reproduzidas. O artefato mais grave costuma ser o aparecimento de regiões uniformes, causando um efeito desagradável na representação de imagens do mundo real. Para contornar os efeitos deste artefato, utiliza-se técnicas de *dithering* e meio-tom digital no processo de quantização, explorando a característica de filtro passa-baixas conferida pelo sistema visual humano (cores ilusórias podem ser criadas pela mistura espacial de duas ou mais cores, efeito este utilizado na técnica francesa de pintara do *Pontilhismo*, que teve como expoentes os pintores Georges Seurat e Paul Signac [15]).

O propósito deste trabalho é utilizar a VQ nas cores de uma imagem, usando diferentes métricas de cor para estabelecer o *codebook* (paleta de cores). Utilizar-se-á um modelo de difusão de erro e a avaliação dos resultados será feita utilizando diferentes métodos, como o MSE[1] (erro médio quadrático), PSNR[1] (relação sinal-ruído de pico), SSIM[1] (índice de similaridade estrutural, método muito utilizado para prever a qualidade percebida de imagens digitais) e o Buteraugli (modelo psicovisual para estimar a similaridade de imagens). Estes métodos serão apresentados na seção IV.

Além de proporcionar compressão dos dados, a VQ das cores é fator importante em se analisar, visto que diversos aparelhos apresentam uma limitação no número de cores que podem ser geradas simultâneamente. As impressoras jato de tinta, por exemplo, possuem uma gama (*gamut*) de cores consideravelmente reduzida em relação às telas LCD. Ademais, no caso de impressão, a extensão possível de cores dependerá da qualidade do papel utilizado. Bem como, a análise da VQ é importante por ser esta uma ferramenta útil no processo de reconhecimento de padrões em imagens digitais. Neste trabalho será abordado apenas a tarefa de estabelecer uma paleta de cores visualmente ótima para representar uma imagem. A tarefa de mapeamento desta paleta para a reprodução da imagem não será abordada.

Leonardo C. Araújo, João Pedro H. Sansão e Sandro A. Fasolo, CELTA, Universidade Federal de São João del Rei, Ouro Branco-MG, E-mails: leolca@ufsj.edu.br, joao@ufsj.edu.br e sandro@ufsj.edu.br.

[1]MSE (*mean squared error (MSE)*), PSNR (*peak signal-to-noise ratio*) e SSIM (*structural similarity index*.

II. Quantização Vetorial

Considere uma variável aleatória X que assume valores $x \in \mathcal{X} \subseteq \mathbb{R}$, e $\mathcal{X}^n \subseteq \mathbb{R}^n$ o conjunto de vetores $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$. Um quantizador vetorial é uma função de mapeamento $Q : \mathbb{R}^n \rightarrow \mathcal{C}$, onde $\mathcal{C} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_M\}$ é um subconjunto constituído por M elementos, $\hat{x}_i \in \mathcal{X}^n$, para $i = 1, \ldots, M$. O parâmetro n indica a dimensionalidade dos dados e do quantizador. O conjunto de aproximação $\mathcal{C}$ é chamado de *codebook*. Pode-se visualizar este processo como um codificador, que associa um endereço de reprodução no *codebook* a cada vetor de entrada (o endereço onde está armazenado o valor de reprodução dado por $Q(x)$) e um decodificador, que irá utilizar este endereço para reproduzir o vetor $\hat{x}$ armazenado no referido endereço. Se uma medida de distorção for definida, $d(x, \hat{x})$, o melhor mapeamento Q será aquele que minimiza a distorção esperada, $\mathrm{E}d(x, \hat{x})$ [16].

Uma simples medida de distorção é o erro quadrático,

$$d(x, \hat{x}) = ||x - \hat{x}||^2 = \sum_{i=1}^{n} (x_i - \hat{x}_i)^2. \tag{1}$$

Outras formas de medida de distorção podem ser utilizados ou propostas, como por exemplo, a utilização do erro quadrado ponderado (*weighted mean square error*, WMSE).

Um dos principais problemas na VQ é o custo computacional associado à busca, em todo *codebook*, pelo valor de representação que mais se aproximada da entrada atual do codificador. Contudo, a estrutura de árvore apresentada por Buzo [2] pode ser utilizada para reduzir este custo computacional.

A. Projeto do Quantizador Vetorial

O objetivo é encontrar o quantizador vetorial ótimo, constituído por um conjunto de M pontos de representação, que minimiza o valor esperado da distorção. O quantizador atinge um ótimo local quando não pode alcançar, através de um procedimento de minimização da distorção, um outro quantizador que produza um melhor resultado. A VQ pode ser vista como um problema de *clusterização*, sendo passível de ser formulado pelo problema das k-médias. Cada *cluster* é representado por um único ponto, ao qual associam-se todos os pontos contidos neste *cluster*. Os pontos pertencentes a um de terminado *cluster* estão mais próximos (segundo uma determinada métrica) ao ponto de representação associado ao seu *cluster* do que a qualquer outro ponto de representação. O problema de *clusterização* do k-médias resume-se no problema de se encontrar o conjunto $\mathcal{C} = \{\hat{x}_1, \ldots, \hat{x}_k\}$ de k pontos de representação para os k *clusters* que irão se formar. Adota-se, então, $k = M$ para o tamanho do *codebook*.

Lloyd propôs um método iterativo, chamado de 'método I', para o projeto de um quantizador escalar [17]. Este método foi, posteriormente, estendido por Linde et al. para o caso vetorial [4]. O algoritmo para o caso de dados com uma distribuição de probabilidade desconhecida é dado a seguir:

1) Seja M o número de pontos de representação e $\epsilon \geq 0$, um limiar de distorção. Considere um conjunto de representação inicial $\mathcal{C}^{(0)} = \{\hat{x}_1^{(0)}, \ldots, \hat{x}_M^{(0)}\}$, um conjunto de treinamento $\mathcal{T} = \{x_1, \ldots, x_N\}$ e $m = 0$, o número inicial de iterações.
2) Dado $\mathcal{C}^{(m)}$ encontre a partição mínima $P(\mathcal{C}^{(m)}) = \{S_i^{(m)} : i = 1, \ldots, M\}$ das sequências de treinamento: $x_j \in S_i^{(m)}$ se $d(x_j, \hat{x}_i^{(m)}) \leq d(x_j, \hat{x}_l^{(m)})$, para todo $l \neq i$. Calcule, então, a distorção média definida por:
$$D^{(m)} = D[(\mathcal{C}^{(m)}, P(\mathcal{C}^{(m)}))] \tag{2}$$
$$= \frac{1}{N} \sum_{j=1}^{N} \min_{\hat{x} \in \mathcal{C}^{(m)}} d(x_j, \hat{x}). \tag{3}$$
3) Se $(D^{(m-1)} - D^{(m)})/D^{(m)} \leq \epsilon$, interrompe a iteração e considera-se $\mathcal{C}^{(m)}$ como o alfabeto final de representação (*codebook*); caso contrário, continue.
4) Encontre o alfabeto ótimo de representação para $P(\mathcal{C}^{(m)})$, $\hat{x}(P(\mathcal{C}^{(m)})) = \{\hat{x}(S_i) : i = 1, \ldots, M\}$ para $P(\mathcal{C}^{(m)})$, onde
$$\hat{x}(S_i) = \frac{1}{||S_i||} \sum_{x \in S_i} x. \tag{4}$$
5) Faça $\mathcal{C}^{(m+1)} = \hat{x}(P(\mathcal{C}^{(m)}))$, incremente m e volte ao passo 2.

O algoritmo descrito acima utiliza um determinado $\mathcal{C}^{(0)}$ como *codebook* inicial para o processo de otimização. Existem diversas maneiras de escolher um alfabeto inicial, a mais simples delas consiste em escolher pontos de representação espaçados e coincidentes com os dados de treinamento, ou seja, os dados utilizados para se estabelecer o *codebook*.

O problema das k médias é um problema NP-difícil [18], desta forma, qualquer algoritmo utilizável levará a uma solução ótima local para o problema. Métodos iterativos, como o de Lloyds, garantem melhorias sucessivas na solução a partir de um ponto inicial. Como a função de minimização, em geral, não é convexa, cada mínimo local terá uma região de atração para si. Os parâmetros de inicialização podem levar a diferentes soluções, algumas melhores que as demais [19].

Existem, ainda, várias outras formas de projetar um *codebook* para realizar a VQ, como por exemplo redes neurais, mapas auto organizáveis de Kohonen[20], recozimento simulado (*Simulated Annealing*) [21], dentre outros [22].

B. Difusão do Erro

Em 1852 foi patenteado por William Fox Talbot a técnica de impressão conhecida como meio-tom (*halftoning*). A técnica consiste na utilização de pontos, variando o tamanho e espaçamento, gerando um efeito de gradiente e sendo assim capaz de simular tons contínuos. A mesma técnica também é utilizada na impressão colorida, variando a densidade das quatro cores secundárias, ciano, magenta, amarelo e preto (CMYK).

A difusão de erro é um tipo de *halftoning* em que o erro gerado no processo de quantização de uma imagem é distribuído entre os *pixels* vizinhos que ainda não foram quantizados. Floyd e Steinberg [23] descreveram o um método para difusão de erro baseado no seguinte *kernel* (núcleo)

$$\frac{1}{16} \begin{pmatrix} & * & 7 \\ 3 & 5 & 1 \end{pmatrix}, \tag{5}$$

onde $*$ denota o *pixel* que está sendo quantizado na etapa corrente do processo de quantização e os demais valores

denotam o percentual do erro que será distribuído aos *pixels* vizinhos. Para aplicar o esquema de difusão de erro de Floyd e Steinberg, o processo de quantização transcorre de cima para baixo e da esquerda para a direita. Em cada passo um *pixel* é quantizado e o erro de quantização é difundido nos *pixels* vizinhos, segundo o *kernel* dado na eq. (5). O erro vai sendo acumulado nos *pixels* que ainda serão quantizados, aumentando assim a probabilidade deles serem representados por um ponto de representação vizinho àquele que seria representado sem a presença do erro difundido de seus vizinhos.

Em algumas implementações, a direção de varredura da imagem durante o processo de quantização se alterna entre as linhas. Este tipo de varredura é chamado de varredura em serpentina ou transformação Bustrofédon[2].

No caso de imagens coloridas, o mesmo algoritmo pode ser aplicado a cada uma das componentes de cor (RGB ou CMYK) separadamente. Entretanto, argumenta-se que melhores resultados podem ser obtidos quando é realizada previamente uma conversão dos canais de cores utilizando um modelo perceptivo de cores, separando os canais de luminosidade, matiz e saturação. Desta forma, pondera-se com um peso maior a difusão do erro no canal de luminosidade, uma vez que a visão humana percebe mais facilmente variações em luminosidade do que variações nas componentes de cor.

III. Diferenças entre cores

O presente trabalho analisou a VQ de cores em uma imagem digital. Como visto na secção anterior, para definir um *codebook* otimizado utilizando a abordagem das k-médias e o algoritmo de Lloyds, é necessário definir uma medida de dissimilaridade entre cores. Define-se a dissimilaridade entre cores como a distância entre elas em um determinado espaço de cores. A maneira usual de se definir distância é através da métrica euclidiana. Se uma cor é representada pela enupla das 3 componentes de cor no sistema RGB, define-se a distância euclidiana entre duas cores como

$$d_{\text{RGB}} = \sqrt{(R_2 - R_1)^2 + (G_2 - G_1)^2 + (B_2 - B_1)^2}. \quad (6)$$

Entretanto, esta distância não leva em consideração as características visuais perceptivas do sistema visual humano. No intuito de melhor ajustá-la à percepção humana, poderia-se, simplesmente, ponderar cada uma das componentes pela sensitividade do cone correspondente (30% vermelho, 59% verde e 11% azul), entretanto, tal ponderação é, em suma, a contribuição de cada componente ao brilho, mostrando-se assim pior na determinação de cores.

A Comissão Internacional de Iluminação (CIE, *International Commission on Illumination*) definiu diferentes métricas para medir a dissimilaridade de cores que proporcionam um JND (*just-noticeable difference*)[3]. Neste trabalho utilizou-se as formulações para diferença de cores dada pela CIE [24], [25]. Estas formulações utilizam o espaço de cores CIE L*a*b* (CIELAB), também especificado pela CIE.

O CIELAB foi criado para descrever todas as cores visíveis pelo olho humano. As três componentes do CIELAB representam a luminosidade da cor ($L^* = 0$ representa preto e $L^* = 100$ indica o branco difuso), sua posição entre vermelho/magenta e verde (a* negativo representa verde e positivo magenta), e sua posição entre amarelo e azul (b* negativo representa azul e positivo amarelo). As diferenças entre duas cores, dadas as enuplas (L_1^*, b_1^*, b_1^*) e (L_2^*, b_2^*, b_2^*) são

- CIE76 definida em 1976, utiliza as coordenadas no espaço de cores CIELAB:

$$\Delta_{ab}^* = \sqrt{(L_2^* - L_1^*)^2 + (a_2^* - a_1^*)^2 + (b_2^* - b_1^*)^2}, \quad (7)$$

  $\Delta_{ab}^* \approx 2{,}3$ corresponde a um JND.
- CIE94 é uma extensão da definição anterior para considerar as não uniformidades perceptuais,

$$\Delta_{94}^* = \sqrt{\left(\frac{\Delta L^*}{k_L S_L}\right)^2 + \left(\frac{\Delta C_{ab}^*}{k_C S_C}\right)^2 + \left(\frac{\Delta H_{ab}^*}{k_H S_H}\right)^2} \quad (8)$$

  onde $\Delta L^* = L_1^* - L_2^*$, $C_1^* = \sqrt{a_1^{*2} + b_1^{*2}}$, $C_2^* = \sqrt{a_2^{*2} + b_2^{*2}}$, $\Delta C_{ab}^* = C_1^* - C_2^*$, $\Delta H_{ab}^* = \sqrt{\Delta a^{*2} + \Delta b^{*2} - \Delta C_{ab}^{*2}}$, $a^{*2} = a_1^* - a_2^*$, $b^{*2} = b_1^* - b_2^*$, $S_L = 1$, $S_C = 1 + K_1 C_1^*$, $S_H = 1 + K_2 C_1^*$, e onde k_C e k_H são usualmente iguais a um e os pesos k_L, k_1 e k_2 dependem da aplicação, sendo $k_L = 1$, $K_1 = 0{,}045$ e $K_2 = 0{,}015$ para artes gráficas e $k_L = 2$, $K_1 = 0{,}048$ e $K_2 = 0{,}014$ para texturas.

IV. Comparando duas Imagens

O efeito do erro de quantização em uma imagem, tradicionalmente, é avaliado por algum método como, por exemplo, o erro médio quadrático (MSE) entre a imagem quantizada e a imagem original:

$$\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left[I(i,j) - K(i,j) \right]^2. \quad (9)$$

Também, pode-se calcular a relação sinal ruído de pico (PSNR), dada por

$$\text{PSNR} = 10 \log_{10} \left(\frac{I_{\text{max}}^2}{\text{MSE}} \right), \quad (10)$$

onde I_{max} é o valor de pico da extensão do tipo da imagem (por exemplo, 255 se a imagem for armazenada utilizando inteiro de 8 bits sem sinal). MSE e PSNR são maneiras objetivas e simples de aferir a dissimilhança entre duas imagens, entretanto, não levam em consideração nenhuma característica perceptiva da visão humano. Desta forma, um valor grande de

[2] O nome Bustrofédon remete a uma escrita arcaica, usando alfabeto grego ou também etrusco, em que o sentido de leitura alternava entre as linhas, ao invés de manter uma direção fixa como esquerda para direita, como no caso das línguas europeias, ou direita para esquerda, como no caso do árabe e hebraico. Este tipo de escrita era a forma usual de se escrever nas pedras na Grécia Antiga.

[3] Os estímulos sensoriais costumam variar em grande extensão de magnitudes. O JND é uma proporção fixa, em relação a uma referência sensorial, de forma que a razão entre a amplitude do estímulo que proporciona uma mínima variação perceptível e a referência é aproximadamente constante, ou seja, $\Delta I / I = k$, onde I é a intensidade de um estímulo em particular (tomado como referência), ΔI é a variação em I necessária para que a mudança seja percebida e k uma constante.

MSE, ou baixo PSNR, não significa necessariamente que as imagens são visualmente muito diferentes.

Existem outras maneiras de calcular o grau de dissimilaridade entre imagens como, por exemplo, o índice de similaridade estrutural (SSIM) proposto por Wang et al. [28]. Este método busca modelar a mudança percebida na informação estrutural de imagens, levando em consideração a relação espacial entre *pixels*. O método utiliza uma imagem sem distorção como referência. O cálculo do índice leva em conta como seria percebida a degradação no conteúdo estrutural de uma imagem, incorporando fenômenos de mascaramento de luminância e contraste [28]. Por ser um método em que os resultados se aproximam mais àqueles apreendidos através da percepção humana, o SSIM se tornou um método amplamente utilizado para aferir a qualidade de imagens na indústria televisiva.

Recentemente foi desenvolvido pelo Google o projeto Butteraugli [26] que busca estimar a similaridade psicovisual de duas imagens. O Butteraugli calcula um mapa de diferenças e um índice de similaridade no domínio da mínimas diferenças perceptíveis. Este projeto foi utilizado pelo projeto subsequente, denominado Guetzli [27], para determinar a melhor escolha da matriz de quantização na compressão JPEG. Uma das motivações do Butteraugli é usar as diferenças estatísticas na localização e densidade dos receptores de cor, os cones. A baixa densidade de cones azuis na região da fóvea é um exemplo de característica explorada pelo método. Outra motivação é a utilização do fenômeno de inibição sensorial, através de uma modelagem mais precisa das células ganglionares, responsáveis pela inibição de frequência espacial. O Butteraugli fornece bons resultados apenas para imagens de alta qualidade, pois foi otimizado para imagens que possuem diferenças sutis quando comparadas com a imagem original, uma vez que seu propósito é ser utilizado pelo método Guetzli para fornecer uma compressão JPEG de alta qualidade.

O presente trabalho utilizou os 4 métodos descritos anteriormente para aferir a dissimilaridade entre imagens. Como os valores (ou índices) fornecidos pelos diferentes métodos possuem extensões díspares, cada um deles foi normalizado no intervalo $[0,1]$.

V. Base de Dados

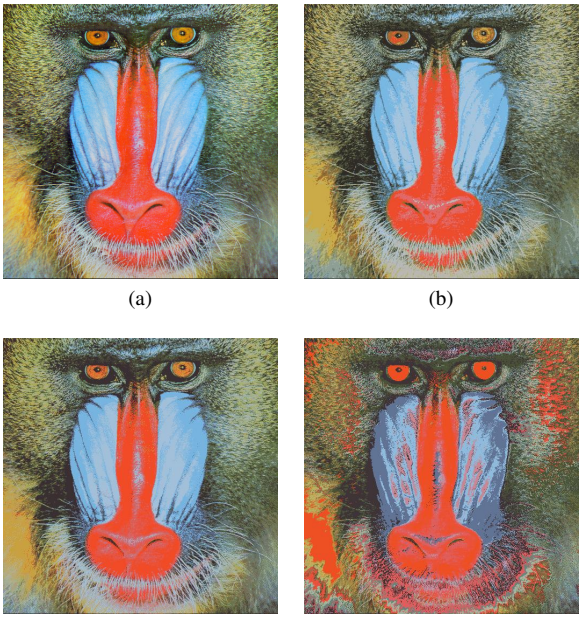
Para a realização dos teste foram utilizadas duas bases de dados: *a*) *The USC-SIPI Image Database*[4], uma base de dados composta por 29 imagens coloridas, todas com 24 bits/pixel, 13 com tamanho de 512x512 pixes e as demais com 256x256 pixels. *b*) A segunda base de dados foi criada a partir de um apanhado de fotos do Instagram realizado com a ferramenta *InstaLooter*[5]. Foram utilizadas 538 imagens do perfil *lisasophielaurent*[6] de variados tamanhos (143 imagens com 1080x1350 pixels, 94 imagens com 640x640 pixels, 85 imagens com 1080x1080 e as demais com diferentes tamanhos).

[4] http://sipi.usc.edu/database/
[5] https://github.com/althonos/instaLooter
[6] https://www.instagram.com/lisasophielaurent/

VI. Resultados

Os resultados foram gerados utilizando uma combinação de *scripts Shell*, *GNU Octave* e *Python*. Todos scripts utilizados estão disponíveis no *GitHub* do autor[7].

Alguns resultados são ilustrados nas imagens da fig. 1. Na fig. 1a apresentamos a imagem original. Esta teve suas cores quantizadas usando uma paleta de 16 cores, criada através o algoritmo de Lloyds. Em cada um dos resultados foram utilizadas diferentes métricas para cálculo da distância entre cores, podendo ter sido utilizado o *dithering* de Floyd e Steinberg ou não. Na fig. 1b foi utilizada distância Euclidiana nas componentes RGB e não foi aplicado *dithering*. Na fig. 1c foi utilizado o CIE76 e aplicado o algoritmo de Floyd e Steinberg. Na fig. 1d foi aplicada a métrica CIE94 e também a técnica de Floyd e Steinberg.



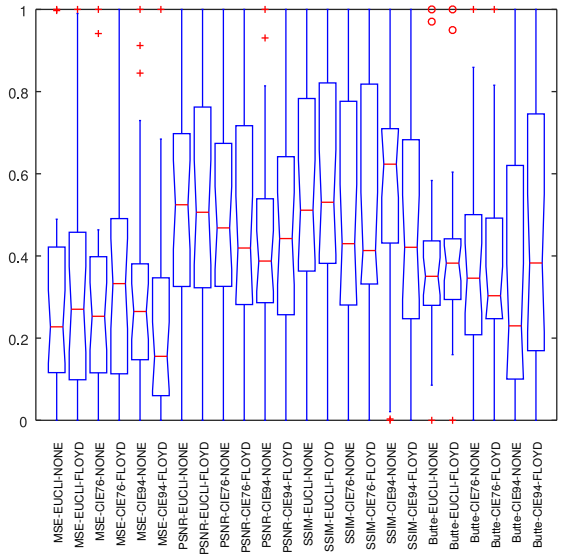
(a) (b) (c) (d)

Fig. 1
Resultados da quantização vetorial. (a) imagem original. (b) distância Euclidiana e sem *dithering*. (c) distância CIE76 e Floyd-Steinberg. (d) distância CIE94 e Floyd-Steinberg.

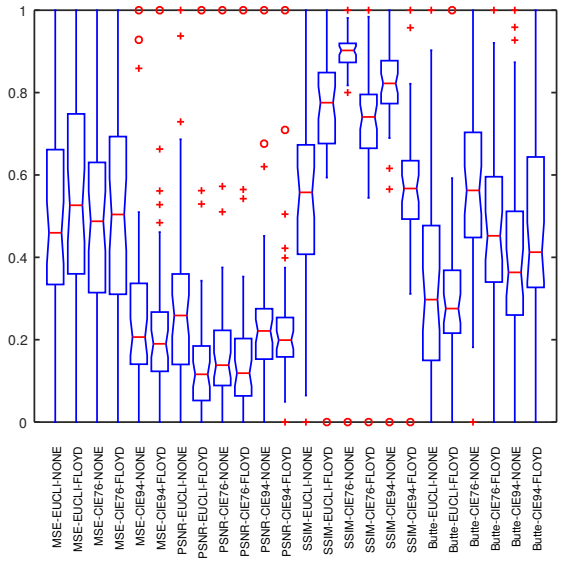
A fig. 2a apresenta um panorama geral dos resultados obtidos na base de imagens USC-SIPI. Podemos analisar a variação dos dados gerados pelos testes variando a métrica de dissimilaridade entre imagens, medida de distância entre cores e aplicação do *dithering*. Utilizando a métrica do MSE para julgar a dissimilaridade entre a imagem original e a imagem quantizada, verificamos que o melhor resultado foi atingido usando CIE94 como medida de distância entre cores e aplicando o algoritmo de *dithering* de Floyd-Steinberg. Ao utilizar o PSNR ou SSIM, o melhor resultado, em média, foi usando CIE76 e com *dithering*. Utilizando o Butteraugli como métrica de dissimilaridade entre imagens, utilizando o CIE94 e sem *dithering* levou, em média, aos melhores resultados.

[7] https://github.com/leolca/color-quantize

Usando uma base de dados maior, os resultados apresentados na fig. 2b mostram desempenho semelhante ao anterior. Para cada umas das métricas: MSE, PSNR, SSIM e Butteraugli, os melhores resultados foram, respectivamente: CIE94 e com *dithering*, Euclidiana com *dithering*, CIE94 e com *dithering*, e Euclidiana com *dithering*.



(a)



(b)

Fig. 2

COMPARAÇÃO NORMALIZADA PELAS MÉTRICAS DE DISSIMILARIDADE ENTRE IMAGENS. VARIAÇÃO MÉTRICA PARA CALCULAR DISTÂNCIA ENTRE CORES E APLICAÇÃO OU NÃO DO *dithering* NO PROCESSO DE QUANTIZAÇÃO. (A) BASE DE DADOS USC-SIPI. (B) BASE DE DADOS INSTAGRAM.

VII. CONCLUSÕES

No presente trabalho foram utilizados métodos objetivos, alguns deles que consideram características psicovisuais, para analisar a qualidade da VQ das cores em imagens digitais. A paleta de cores foi definida utilizando o algoritmo de Lloyds e diferentes métricas de dissimilaridade entre cores foram testadas.

Os resultados apresentados mostram uma grande variabilidade, sendo portanto difícil concluir qual é a melhor abordagem. Entretanto, há de se ressaltar que a abordagem mais simples, utilizando a distância euclidiana e sem *dithering* apresentou resultados com qualidade visual igual ou superior às demais.

REFERÊNCIAS

[1] A. S.G Andrae, "*Total Consumer Power Consumption Forecast*", Nordic Digital Business Summit, 2017.

[2] A. Buzo, A. H. Gray, R. M. Gray, e J. D. Markel, "*Speech coding based upon vector quantization*", IEEE Trans. Acoust. Speech, Signal Processing, vol. ASSP-28, pp. 562–574, Out. 1980.

[3] A. Gersho e V. Cuperman, "*Vector quantization: A pattern-matching technique for speech coding*", IEEE Commun. Mag., pp. 15–21, Dez. 1983.

[4] Y. Linde, A. Buzo e R. M. Gray, "*An algorithm for vetor quantizer design*", IEEE Trans. Commun., vol. COM-28, pp. 84–95, Jan. 1980.

[5] H. Abut, R. M. Gray e G. Rebolledo, "*Vector quantization of speech and speech-like waveforms*", vol. ASSP-30, pp. 423–435, Jun. 1982.

[6] , R. M. Gray, "*Vector quantization*", IEEE ASSP Mag., pp. 4–29, Abr. 1984.

[7] , N. M. Nasrabadi e R. A. King, "*Image Coding using Vector quantization: a review*", IEEE Trans. on Commun. vol. 36, No. 8, Ago. 1988.

[8] M. T. Orchard e C. A. Bouman, "*Color Quantization of Images*", IEEE Trans. on Signal Proc., vol. 39, No. 12, Dez. 1991.

[9] D. Özdemir e L. Akarun, "*A fuzzy algorithm for color quantization of images*", Pattern Recognition, 35, pp. 1785–1791, 2002.

[10] P. Renevey, R. Vetter e J. Kraus, "*Robust Speech Recognition using Missing Feature Theory and Vector Quantization*", Eurospeech, pp. 1107–1110, 2001.

[11] J. Martinez, H. Perez, E. Escamilla e M. M. Suzuki, "*Speaker recognition using Mel frequency Cepstral Coefficients (MFCC) and Vector quantization (VQ) techniques*", 22nd International Conference on Electrical Communications and Computers, 2012.

[12] R. M. Gray, "*Gauss mixture vector quantization*", IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001.

[13] G. Csurka, C. R. Dance, L. Fan, J. Willamowski e C. Bray, "*Visual categorization with bags of keypoints*", Workshop on Statistical Learning in Computer Vision, pp. 1–22, 2004.

[14] A. Coates e A. Y. Ng, "*The importance of encoding versus training with sparse coding and vector quantization*", Proceedings of the 28th International Conference on International Conference on Machine Learning, pp. 921–928, 2011.

[15] H. Widauer e K. A. Schröder, "*Ways of Pointillism: Seurat, Signac, Van Gogh*", Hirmer, 2016.

[16] A. Gersho, "*On the structure of vector quantizers*", IEEE Trans. on Inf. Theo., Vol 29, No. 2, Mar. 1982.

[17] S. P.Lloyd, "*Least-squares quantization in PCM*", IEEE Trans. on Inf. Theo., Vol. 25, pp. 129–137, Mar. 1982.

[18] M. Mahajana, P. Nimbhorkara e K. Varadarajan, "*The planar k-means problem is NP-hard*", Theoretical Computer Science, Vol. 442, pp. 13–21, Jul 2012.

[19] , D. Arthur e S. Vassilvitskii, "*k-means++: the advantages of careful seeding*", Proc. of the 18th annual ACM-SIAM symposium on Discrete algorithms, pp. 1027–1035, 2007.

[20] H. Wei, Y. Chang e J. Wang, "*A Kohonen-based structured codebook design for image compression*", Proceedings of TENCON, 1993.

[21] J. Vaisey e A. Gersho, "*Simulated annealing and codebook design*", International Conference on Acoustics, Speech, and Signal Processing, 1988.

[22] M. Lech e Y. Hua, "*Vector quantization of images using neural networks and simulated annealing*", Proc. of the 1991 IEEE Workshop Neural Networks for Signal Processing, 1991.

[23] R.W. Floyd e L. Steinberg, "*An adaptive algorithm for spatial grey scale*", Proc. of the Society of Information Display, Vol. 17, pp. 75–77, 1976.

[24] M. D. Fairchild, "*Color Appearance Models*", Wiley, 2013.

[25] CIE, "*Colorimetry*", Commission Internationale de l'Eclairage, 2004.

[26] Google, https://github.com/google/butteraugli, 2017

[27] Google, https://github.com/google/guetzli/, 2017

[28] Z. Wang, A. C. Bovik, H. R. Sheikh e E. P. Simoncelli, "*Image quality assessment: from error visibility to structural similarity*", Vol 14, No. 4, pp. 600–612, 2004.