

Processamento de Imagem

4 - Imagens de Cor

Análise e Processamento de Imagem (M4031, M4094)

Processamento de Sinal e Imagem em Física Médica (F4012)

[Conteúdo]

click on it

André R. S. Marçal

Departamento de Matemática
Faculdade de Ciências, Univ. Porto (FCUP)
<http://www.fc.up.pt/pessoas/andre.marcal>

versão 1.0 - 9 Fevereiro 2021

Conteúdo

1	Introdução	3
1.1	Padrão CIE	3
1.2	Modelos aditivos e subtrativos	5
2	Modelos de cor	6
2.1	Modelo RGB	6
2.1.1	Composições RGB	7
2.2	Modelos CMY e CMYK	8
2.3	Modelos HSV e HSI	9
2.4	Outros Modelos	10
3	Imagens de cor indexada	11
4	Imagens de pseudo-cor	13
5	Operações Pontuais em imagens de cor	15
5.1	Equalização de histograma	16
5.2	Correção de cor	17
6	Filtros Espaciais em imagens de cor	18
6.1	Deteção de contornos	19
7	Imagen de cor no MATLAB	20
7.1	Conversão entre modelos de cor	20
7.2	Imagens de cor indexada	21
7.3	Processamento de imagens de cor	22
8	Referências Bibliográficas	23
9	Exercícios propostos	24
9.1	APPENDIX - Exercices (in English)	25

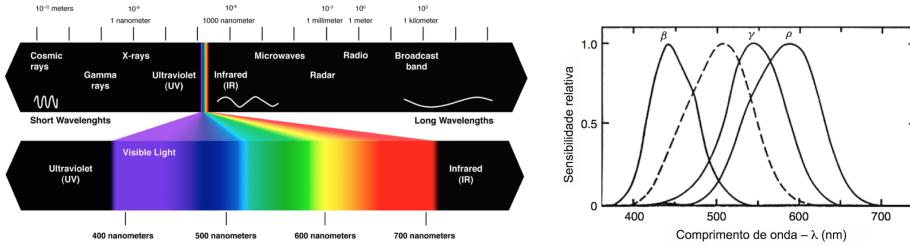


Figura 1: Esq. - Espectro electromagnético, com destaque da região do visível (aprox. $0.4\mu\text{m} < \lambda < 0.7\mu\text{m}$). [Widen] Dir. - Sensibilidade espectral do olho humano para 3 tipos de cones (β , γ , ρ) e bastonetes (tracejado). [Hunt]

1 Introdução

A capacidade de visão o ser humano limita-se a uma pequena faixa do espectro electromagnético, a que se chama região do visível, e que corresponde a comprimentos de onda (λ) entre 0.4 e $0.7\mu\text{m}$. A figura 1 (esq.) mostra uma representação esquemática do espectro electromagnético, com a região do visível destacada, e com identificação das várias zonas. A zona destacada corresponde a radiação ultravioleta ($\lambda < 0.4\mu\text{m}$), visível (lilás, azuis, verdes, amarelos, laranjas e vermelhos, por ordem crescente de λ), e radiação infravermelha ($\lambda > 0.7\mu\text{m}$).

À semelhança do que se observa com intensidades de luz, ou de tons de cinzento numa imagem, a percepção visual de cor é subjetiva, variando de pessoa para pessoa. A figura 1 (dir.) apresenta gráficos da sensibilidade espectral média do olho humano para 3 tipos de cones (β , γ , ρ), responsáveis pela discriminação de cores, e de bastonetes (linha tracejada), associados apenas à percepção de intensidade de radiação ou luminosidade.

No caso de imagens a cores, é habitual pretender-se representar a imagem digital com a melhor aproximação possível da percepção que um ser humano tipicamente teria do local observado no momento de obtenção da imagem. Isso deverá ser feito pelo dispositivo de aquisição da imagem (ex. câmara fotográfica) e também pelo dispositivo de visualização posterior (ex. ecrã). Por outro lado, poderá ser útil realçar a imagem, por exemplo aumentando o contraste, mas tentando-se preservar a componente cromática (de cor). Ou seja, apresentar uma versão melhorada, mas realista, do que o olho humano conseguiria observar no local. Uma outra abordagem totalmente diferente é criar imagens coloridas que não pretendem ser uma representação fiel da percepção humana de cor, mas sim usar a maior capacidade que o sistema de visão humano tem em distinguir cores do que intensidades de luz (ou níveis de cinzento) para apresentar o conteúdo de uma imagem de forma mais eficiente.

1.1 Padrão CIE

A Comissão Internacional de Iluminação (CIE - Commission internationale de l'éclairage) desenvolveu o padrão CIE 1931, que foi o primeiro modelo a caracterizar a percepção de cor de um ser humano, relacionando-a com a radiação electromagnética. Uma vez que o olho humano tem 3 tipos de cones com dife-

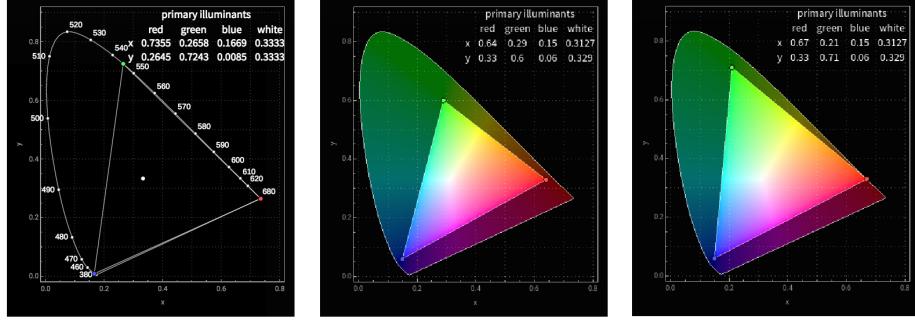


Figura 2: Diagramas de cromaticidade CIE: espaço de cor com cores espectrais, ponto acromático e componentes primárias (esq); sub-espacos de cores representadas pelos sistemas PAL/SECAM (centro) e HDTV (Direita). [Chang]

rentes sensibilidades espectrais (figura 1, dir.), a representação de todas as cores visíveis envolve o uso de 3 variáveis independentes (modelo tri-dimensional). O modelo CIE XYZ usa 3 variáveis (X , Y e Z) para representar as intensidades observadas associadas às respostas espectrais dos cones β (\bar{x}), γ (\bar{y}), e ρ (\bar{z}), de acordo com (1), onde $L(\lambda)$ é a radiação incidente (observada).

$$\begin{aligned} X &= \int L(\lambda) \bar{x}(\lambda) d\lambda \\ Y &= \int L(\lambda) \bar{y}(\lambda) d\lambda \\ Z &= \int L(\lambda) \bar{z}(\lambda) d\lambda \end{aligned} \quad (1)$$

Uma outra forma do modelo CIE, potencialmente mais interessante, é versão CIE xyz, onde as variáveis x , y e z estão normalizadas, relacionando-se com X , Y e Z através de (2). Nesta forma, há apenas 2 variáveis independentes, por exemplo x e y , sendo $z = 1 - x - y$. Ou seja, bastam 2 variáveis (x e y) para caracterizar a componente cromática, sendo a luminosidade representada por uma variável não normalizada (e.g. Y).

$$x = \frac{X}{X + Y + Z} \quad y = \frac{Y}{X + Y + Z} \quad z = \frac{Z}{X + Y + Z} \quad (2)$$

A figura 2 mostra diagramas de cromaticidade CIE, onde as cores que podem ser vistas pelo olho humano aparecem dentro da região destacada no plano das variáveis x e y (espaço de cor). As cores espectrais aparecem no contorno do espaço de cor, estando os comprimento de onda (λ) correspondentes indicados no diagrama do lado esquerdo. Neste diagrama é igualmente assinalado o ponto acromático ($x = y = z = 1/3$), com uma marca branca, e 3 cores primárias, correspondendo aproximadamente às cores espectrais com $\lambda = 0.4\mu m$ (azul), $\lambda = 0.55\mu m$ (verde) e $\lambda = 0.7\mu m$ (vermelho).

A percepção de cores é aproximadamente linear, pelo que as cores correspondentes a valores de x e y dentro do triângulo definido pelas 3 cores primárias

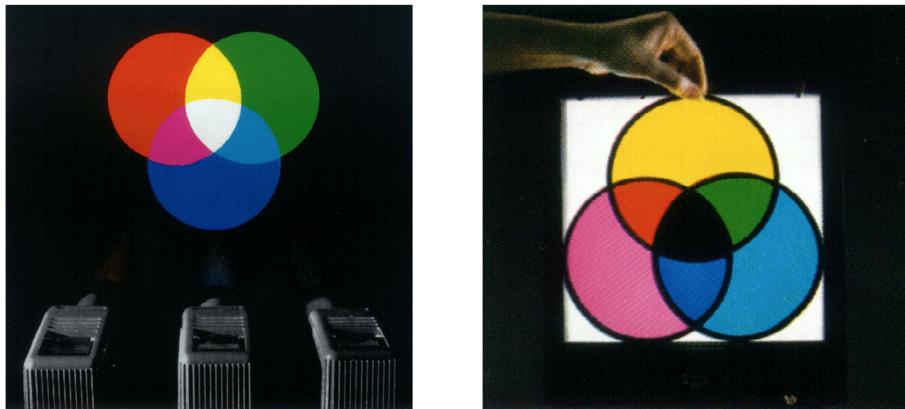


Figura 3: Modelo de cor aditivo com focos de luz (esquerda); modelo de cor subtrativo com filtros sobre fonte de luz branca (direita). [Gonzalez]

podem ser obtidas misturando diferentes intensidades das cores primárias. A figura 2 mostra os sub-espacos de cores representadas pelos sistemas PAL/SECAM (centro) e HDTV (direita). A gama de cores que pode ser apresentada por um dado sistema é em geral bastante menor que o espaço de cor (cores que o olho humano pode captar). Convém referir que as cores que aparecem na figura são meramente ilustrativas, não representando a diversidade de cores apresentadas pelos sistemas PAL/SECAM e HDTV, por limitação dos meios utilizados (digital ou impresso) na visualização da figura.

1.2 Modelos aditivos e subtrativos

De uma forma geral a representação das cores visíveis envolve o uso de um modelo com 3 variáveis independentes, podendo ser considerados dois grupos de modelos: aditivos, normalmente mais relevantes para cor luz; subtrativos, normalmente úteis para cor pigmento, por exemplo em imagens impressas.

O modelo aditivo pode ser ilustrado com 3 focos de luz com as cores primárias vermelho, verde e azul, como é apresentado na figura 3 (esquerda). A zona de interseção entre 2 feixes dá origem às cores secundárias: amarelo (vermelho+verde), azul claro ou ciano (verde+azul) e magenta (azul+vermelho). A zona de sobreposição das 3 cores primárias corresponde ao branco, e a zona sem incidência de luz de qualquer projetor ao preto.

O modelo subtrativo pode ser ilustrado com tintas colocadas sobre papel branco, ou com filtros a bloquear uma fonte de luz branca, como apresentado na figura 3 (direita). As cores primárias são neste caso o amarelo, ciano e magenta, e as cores secundárias o vermelho, verde e azul. A sobreposição dos 3 filtros, que corresponde à soma das 3 cores primárias, dá origem ao preto.



Figura 4: Esquema da estrutura de dados RGB (esquerda); imagem RGB de 512 por 512 pixels (centro); detalhe com os valores da intensidade das componentes R, G e B para alguns pixels (direita).

2 Modelos de cor

Nesta secção são apresentados de forma breve os modelos de cor mais relevantes para processamento digital de imagem.

2.1 Modelo RGB

Uma imagem a cores no modelo RGB consiste numa matriz tridimensional, com N_L linhas e N_C colunas nas duas primeiras dimensões, à semelhança de uma imagem de cinzento, mas com 3 bandas que correspondem às intensidades de cada uma das cores primárias - vermelho (*Red*), verde (*Green*) e azul (*Blue*). A estrutura de uma imagem RGB é ilustrada na figura 4 (esquerda), sendo apresentada também um exemplo de uma imagem RGB de 512 por 512 pixels (centro), de 24 bits (8 bits ou 1 byte para cada banda). No lado direito da figura 4 são apresentados os valores das componentes R, G e B de pixels numa pequena região da imagem onde está uma bola cor de rosa. Como se pode observar na figura, os valores de intensidade destes pixels na banda de vermelho são muito mais altos do que nas bandas de verde e azul, produzindo assim a tonalidade rosa forte observada. Na parte superior esquerda da secção destacada aparecem alguns pixels com tonalidade cinzenta escuro, com intensidades baixas e muito parecidas nas 3 bandas. Na verdade um nível de cinzento terá exactamente a mesma intensidade nas 3 bandas, sendo preto quando $I_r = I_g = I_b = 0$ e branco quando $I_r = I_g = I_b = 255$ (valor máximo para 8bits sem sinal).

A estrutura mais comum para imagens RGB é 24 bits, onde é usado um byte para cada componente, 3 bytes (24bits) no total. Este formato de dados permite representar $2^{24} = 256^3 = 16777216$ cores. Ou seja, cerca de 16 milhões, muito mais do que as poucas centenas de cor que o ser humano consegue distinguir. Há no entanto outros formatos de dados usados para imagens RGB. O mínimo será 3 bits - 1 bit (0/1 ou ON/OFF) para cada componente, o que permite representar apenas $2^3 = 8$ cores (branco, preto as 3 cores primárias e as 3 cores secundárias). Este tipo de cor RGB foi usada em computadores e consolas de jogos, sendo na altura uma melhoria considerável relativamente aos sistemas monocromáticos. Considerando uma escala de 6 níveis para cada componente, é possível representar $6^3 = 216$ cores, podendo neste caso ser reservado apenas 1 byte (256 níveis) por pixel.

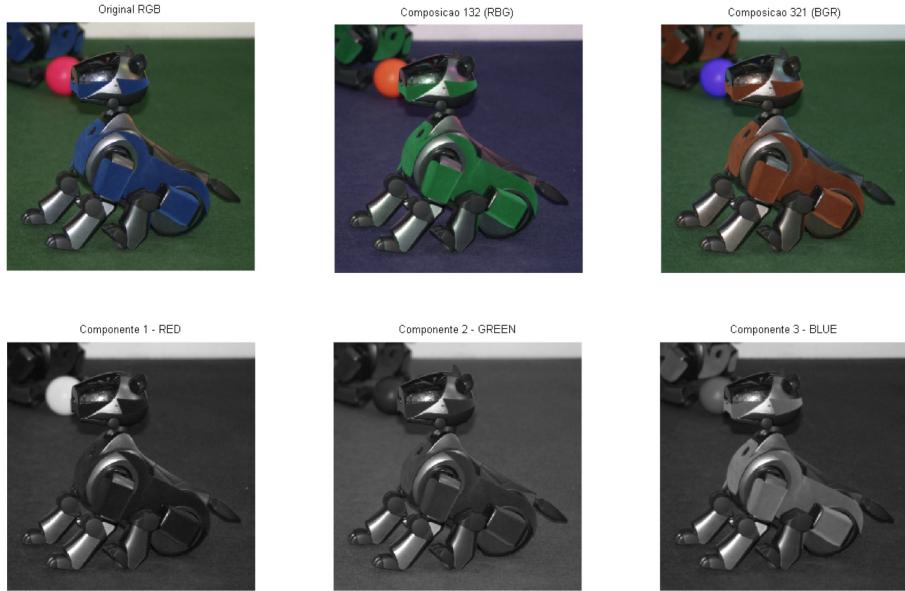


Figura 5: Imagem RGB original (em cima à esquerda); imagens em escala de cinzento com a intensidade de cada componentes R, G e B (em baixo); composições RGB 132 (em cima ao centro) e 321 (em cima à direita).

O modelo RGB relaciona-se com o modelo CIE XYZ ([1.1](#)) através de (3), sendo os valores apresentados exactos. Para converter cores de XYZ para RGB é necessário inverter a matriz de (3) o que dá origem a valores aproximados, que não são apresentados aqui.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{1}{0.17697} \begin{bmatrix} 0.49 & 0.31 & 0.20 \\ 0.17697 & 0.8124 & 0.01063 \\ 0 & 0.01 & 0.99 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3)$$

2.1.1 Composições RGB

Uma imagem RGB a 24 bits, como a imagem do Cão-Robot apresentada nas figuras 4 (centro) e 5 (em cima à esquerda), tem 3 bytes por pixel, ou seja 1 byte para cada componentes R, G e B. Estas componentes são imagens de 1 banda, com valores entre 0 e 255, podendo ser vistas com uma escala de cinzento, como se mostra na figura 5 (em baixo). Por exemplo, a bola aparece muito clara na componente 1-Red (esquerda) e escura na componente 2-Green (centro). Em geral neste tipo de imagens o objectivo é ter uma representação de cor o mais próxima possível da de um observador humano ('cor verdadeira'). No entanto, a atribuição das 3 componentes RGB pode ser feita de uma forma diferente da normal ($R=1, G=2, B=3$), o que dá origem a imagens de 'cor falsa'. A figura 5 mostra 2 composições RGB deste tipo, em que nos canais RGB foram colocadas as bandas 132 (em cima ao centro) e 321 (em cima à direita).

As composições RGB de 'cor falsa'são particularmente úteis para imagens

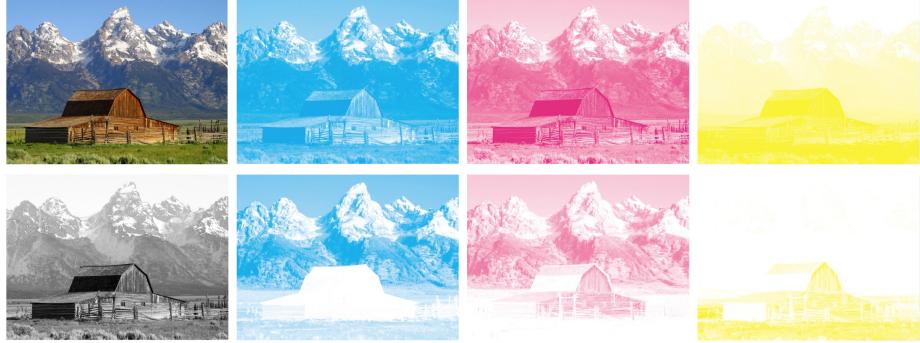


Figura 6: Componentes CMY (em cima) de uma imagem RGB (em cima à esquerda); componentes CMYK da mesma imagem RGB (em baixo).

multi-espectrais, que normalmente tem bandas fora da região do visível (e.g infravermelho). O objectivo é normalmente apresentar o máximo de informação numa única imagem. Considerando as características do olho humano, podem ser apresentadas 3 bandas em simultâneo através de uma composição RGB de 3 das bandas da imagem multi-espectral.

2.2 Modelos CMY e CMYK

O modelo de cor CMY é um modelo subtrativo, ilustrado na figura 3 (direita) para uma fonte de luz branca com filtros sobrepostos, mas que é normalmente usado para meios impressos. O modelo CMY relaciona-se com o modelo RGB (2.1) através de (4). As cores primárias do modelo CMY são as cores secundárias do modelo RGB e vice-versa (ver figura 3).

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (4)$$

Uma variante do modelo CMY é o modelo de cor CMYK - *Cyan, Magenta, Yellow, Key* (preto), que tem 4 componentes associadas às camadas de tinta usada em muitos sistemas de impressão. Uma das razões para se usar a componente adicional de preto (K) é o facto do preto obtido por sobreposição das componentes CMY não ser tão bem definido como o preto da componente independente K. Para além disso, há vantagens em termos de custos, relacionados com o menor uso de tinta colorida, e melhor qualidade da impressão, em particular em zonas a preto (por exemplo texto). A figura 6 mostra uma imagem RGB (em cima à esquerda) decomposta nas componentes CMY (em cima) e nas componentes CMYK (em baixo). Comparando as componentes dos 2 modelos, pode facilmente ser verificado que a quantidade de tinta colorida usada no modelo CMYK é muito menor do que no modelo CMY. Para além disso a imagem a cores impressa deverá ter melhor qualidade usando o modelo CMYK.

A técnica de *halftoning* é frequentemente usada com o modelo CMYK para impressão. São usadas pequenas marcas com um padrão regular, em vez de tinta sobreposta, permitindo a percepção de cor sólida a uma certa distância.

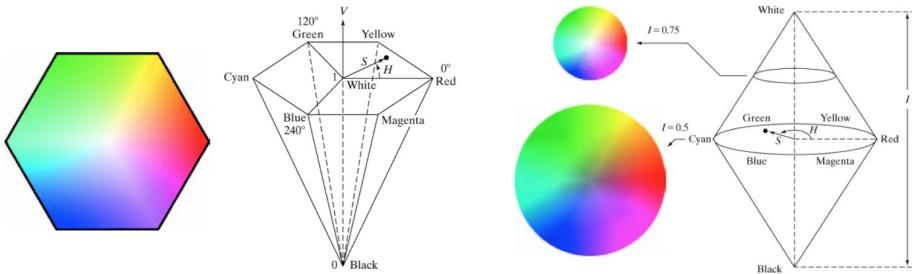


Figura 7: Representação esquemática dos modelos de cor HSV hexagonal (esquerda) e HSI cônico (direita).

2.3 Modelos HSV e HSI

Os modelos de cor referidos anteriormente permitem identificar uma cor sem ambiguidade, mas não são muito adequados à forma habitual de se descrever uma cor. Os modelos de cor HSV (Hue, Saturation, Value) e HSI (Hue, Saturation, Intensity) foram desenvolvidos para estarem mais de acordo com a percepção de cor de um ser humano. Os modelos HSV e HSI, também referidos como HSL (L-Lightness) ou HSB (B-brightness), tem a componente cromática associada a 2 variáveis - H (matiz) e S (saturação), e uma terceira variável, independente das outras duas, para a intensidade ou luminosidade. Ou seja, a informação sobre a cor está desacoplada da informação sobre a intensidade.

Há várias formas de definir estes modelos, considerando-se um plano para as componentes de cor (H e S) e um eixo perpendicular para a intensidade (mais/menos claro/escuro). A figura 7 (esquerda) mostra o esquema do modelo HSV que dá origem a uma pirâmide, neste caso hexagonal. A componente H (Hue) é uma variável angular, que toma valores entre 0° e 360° , com a sequência de cores vermelho (0°), amarelo (60°), verde (120°), ciano (180°), azul (240°) e magenta (300°). A componente S (Saturation) toma valores entre 0 e 1. A cor fica definida pelos valores H e S, que definem um ponto no plano da base. A variação de luminosidade consiste em deslocar este ponto ao longo da linha que o une ao vértice da pirâmide. Desta forma a variação de luminosidade não tem qualquer efeito na componente cromática. Podem ser consideradas outras formas para a base, tais como um triângulo ou um círculo, dando origem a diferentes espaços de cor.

Uma versão eventualmente mais ajustada à percepção de cor é o modelo HSI, ilustrado na figura 7 (direita). Neste caso o espaço de cor consiste em 2 cones adjacentes, sendo as componentes H e S representadas num círculo. O tamanho deste círculo é máximo para $I=0.5$, diminuindo à medida que I se aproxima de 1 (branco) ou 0 (preto). Esta representação é particularmente feliz na medida em que permite representar uma maior diversidade de cores para intensidades intermédias, e menor diversidade para intensidade extremas (baixas ou altas), o que corresponde à percepção de cor pelo olho humano.

A conversão de cor RGB para HSV ou HSI depende da forma do espaço de cor considerada, sendo essencialmente obtida através de uma transformação geométrica a partir do cubo RGB. A figura 8 mostra as componentes HSV de uma imagem colorida, sendo V a versão tons de cinzento da imagem original.

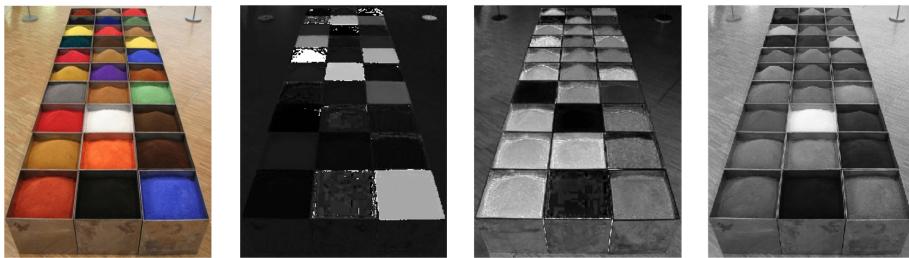


Figura 8: Imagem de cor RGB 24 bits (esquerda) e componentes da imagem no modelo HSV (esq. para dir): Hue (H), Saturation (S), Value (V).

2.4 Outros Modelos

Há vários outros modelos de cor que são regularmente usados em processamento digital de imagem. Alguns desses modelos relacionam-se com o modelo RGB (2.1) de uma forma simples, através de um sistema de 3 equações lineares. Por exemplo, o modelo YIQ (Y-Luminance, I-Hue, Q-Saturation) através de (5) e o modelo YCbCr através de (6), para imagens de cor a 24 bits. Nestes modelos a informação sobre a luminosidade é representada apenas na componente Y. No caso do modelo YCbCr a informação sobre a cor está associada às componentes Cb e Cr - diferenças cromáticas em azul (Cb) e vermelho (Cr).

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.528 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.533 & 24.966 \\ -37.797 & -74.203 & 112 \\ 112 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (6)$$

Um modelo com características particularmente interessantes é o espaço de cor CIELAB ou L*a*b*. O modelo L*a*b* pretende ser independente do dispositivo (écran, impressora, etc.). À semelhança de outros modelos, a intensidade de luz é representada numa componente (L*) e a informação cromática nas outras duas (a* e b*). No entanto, uma característica particular do modelo L*a*b* é que este está focado na representação de luminosidade e cor numa perspectiva de percepção humana. Em princípio, os valores de L* deverão estar relacionados com a percepção (subjetiva) de luminosidade de um observador humano, entre 0 (preto) e 100 (branco).

O modelo L*a*b* foi projectado com o objetivo de assegurar que há uma correspondência entre distâncias no plano de cor (a*b*) e a distância aproximada em termos de percepção. Para isso são usadas as 4 cores consideradas únicas no ponto de vista de percepção: vermelho, verde, azul e amarelo. Os pares de cores opostas considerados são vermelho/verde (a*) e azul/amarelo (b*), podendo tomar valores positivos ou negativos (ou 0).

A conversão de cor RGB para L*a*b* deverá em princípio envolver a linearização da imagem em relação à fonte de iluminação, que poderá não ser conhecida.

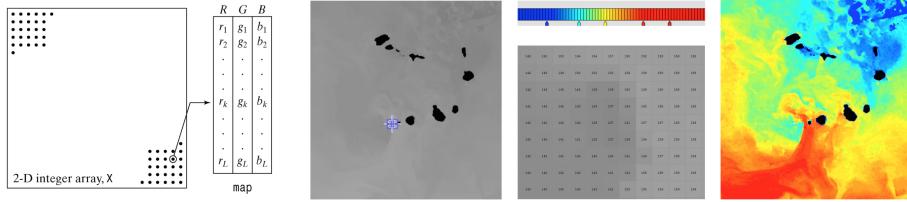


Figura 9: Esquema de imagem de cor indexada (esquerda) e exemplo para uma imagem de temperatura da superfície do mar apresentada com escala de cinzento e com uma tabela de cor favorável à interpretação (direita).

3 Imagens de cor indexada

Os modelos de cor apresentados anteriormente baseiam-se normalmente em 3 componentes (bandas) para representar 3 variáveis de independentes. Por exemplo uma imagem RGB a 24 bits é uma estrutura com 3 bandas (3x8 bits), permitindo representar um total de $2^{24} = 16777216$ cores. É no entanto habitual que uma imagem colorida tenha apenas um pequeno número de cores diferentes (e.g. algumas dezenas), pelo que o uso de 24 bits é excessivo para representar a informação de cor. Uma alternativa nestes casos é a utilização de imagens de cor indexada. Uma imagem de cor indexada tem apenas 1 banda, representando valores numéricos num dado intervalo (ex. de 0 a 255, para 8 bits), tendo associada uma tabela de cor onde são indicados os valores RGB a atribuir a cada nível. A figura 9 (esq.) apresenta de forma esquemática a estrutura de uma imagem de cor indexada. O pixel destacado, com intensidade k , irá ser apresentado não com o nível de cinzento k mas sim com a cor RGB obtida da tabela associada à imagem (r_k, g_k, b_k) . Se o número de cores for até 256, é possível armazenar o conteúdo de uma imagem RGB em apenas 1 banda sem qualquer perda de qualidade, o que poderá ser benéfico não apenas em termos de utilização de espaço de armazenamento mas também em velocidade de processamento. Há no entanto outras razões para se usarem imagens de cor indexada.

Na figura 9 (direita) é apresentado um exemplo com uma imagem de satélite do Arquipélago de Cabo Verde. Esta imagem de infravermelho térmico foi georeferenciada e calibrada, tendo sido colocado o valor 0 nas zonas de terra (as ilhas). Os restantes pixels tem como valor 10 vezes a temperatura da água em °C. Na pequena região destacada podem-se observar alguns valores (por ex. 148, correspondendo a 14,8 °C) e as respectivas tonalidades de cinzento. A imagem em tons de cinzento é difícil de interpretar porque há pouca diversidade de valores nos pixels de água, logo os tons de cinzento são muito parecidos. Uma alternativa que melhora consideravelmente a percepção visual é usar uma tabela de cor passando a imagens de níveis de cinzento a ser uma imagem de cor indexada. A figura 9 (direita) mostra a tabela de cor escolhida e a versão colorida resultante. Neste caso escolheu-se uma sequência de cores particularmente favorável para representar temperaturas - a começar em azul escuro, azul claro, esverdeado para águas mais frias, e amarelos, laranjas vermelhos para águas mais quentes. Como se pode observar na figura, esta versão colorida é muito mais fácil de interpretar do que a versão original, a níveis de cinzento.



Figura 10: Imagem RGB 24 bits usando 147866 cores diferentes (esq.); versões de cor indexada com 32 e 8 cores (centro); versão de 8 cores com *dither* (direita).

As imagens de cor indexada podem ser úteis para melhorar a percepção visual do contúdo da imagem ou para reduzir o volume de dados. Considerando dados a 8 bits sem sinal (valores inteiros entre 0 e 255), é possível representar 256 cores diferentes na imagem de cor indexada. Uma forma simples de converter uma imagem RGB 24 bits para 8 bits indexada é reduzir a resolução radiométrica de cada componente (R,G,B) de 256 para apenas 6 níveis. A tabela de cor resultante permite representar 216 cores (2.1), e diz-se ser ‘segura’(*safe colour table*) uma vez que é adequada para qualquer imagem, cobrindo de forma razoável uma grande diversidade de cores. Há no entanto outras formas de selecionar a tabela de cor a usar na versão indexada de uma imagem RGB, optimizando a escolha para a imagem RGB a converter. A figura 10 mostra um exemplo para a imagem RGB 24 bits ‘Fringe’. Na imagem original (figura 10, esquerda) são usadas 147866 cores diferentes, tendo sido obtidas tabelas de cor com apenas 32 e 8 cores optimizadas para esta imagem. As versões de cor indexada são apresentadas na figura 10 (centro), com a escala de cores na parte de baixo. Como se pode observar, a representação é bastante boa, mesmo com apenas 8 cores. No entanto as zonas de variação gradual, por exemplo nas superfícies douradas dos bonecos e no cinzento do pavimento, não são bem representadas.

Uma técnica interessante na preparação de imagens para impressão, é fazer a conversão para cor indexada com *dither*. A ideia desta técnica, que tem semelhanças com *halftoning* (2.2), é introduzir ruído (ou agitação) de forma intencional, para evitar a existência de zonas de transição entre regiões de cor homogénea, que surgem por limitação do número de cores disponíveis. A figura 10 (direita) mostra a imagem ‘Fringe’ com apenas 8 cores obtida com *dither*. Pode observar-se que as zonas de transição gradual são melhor representadas nesta imagem do que na versão de 8 cores simples.

Convém referir que nem todos os formatos *standard* de imagem permitem gravar imagens de cor indexada. Por exemplo, o formato JPG, que um dos mais usados, não permite gravar imagens de cor indexada. Há formatos, como BMP e PNG que apenas permitem usar imagens de cor indexada a 8 bits, ou a 1, 2 (só PNG) e 4 bits. E no caso do formato TIFF é possível gravar imagens quer em cor RGB 24 bits, quer em cor indexada com 1 a 8 bits.

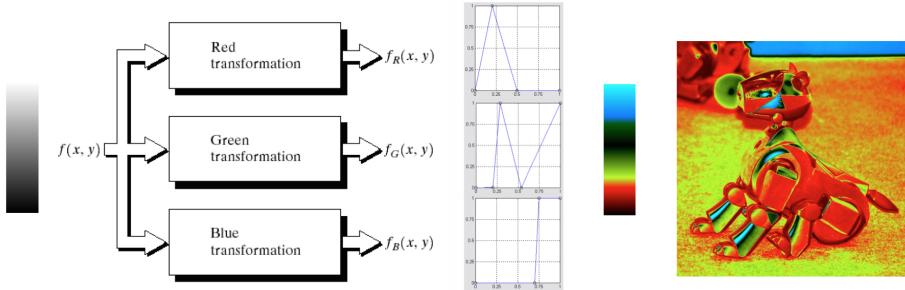


Figura 11: Esquema do processo de criação de imagem de pseudo-cor (esquerda) e exemplo de uma imagem e correspondente tabela de cor (direita).

4 Imagens de pseudo-cor

É possível criar uma imagem a cores, com 3 componentes independentes, a partir de uma imagem em tons de cinzento, a que se chama imagem de pseudo-cor. O objectivo é normalmente melhorar a percepção visual da imagem, beneficiando do facto do ser humano tem mais facilidade em distinguir cores do que níveis de cinzento. A criação de uma imagem de pseudo-cor é feita recorrendo a operações pontuais, aplicando 3 funções de transformação diferentes à imagem original. As 3 imagens resultantes são usadas para criar uma composição colorida RGB ([2.1.1](#)), podendo igualmente ser usado outro modelo de cor.

A figura 11 apresenta um esquema do procedimento (esquerda) e um exemplo (direita). A partir de uma imagem com 1 banda - $f(x,y)$, obtém-se 3 novas versões (f_R , f_G , f_B) através da aplicação de operações pontuais com funções de transformação diferentes, que são usadas como componentes RGB da imagem de pseudo-cor. Apesar desta imagem ser de 24 bits, ela tem apenas 256 cores diferentes, pelo que o formato cor indexada a 8 bits pode igualmente ser usado. A figura 11 mostra um exemplo, sendo apresentadas as funções de transformação, a escala de tons de cinzento da imagem original, e a escala de cores da imagem de pseudo-cor. Ao contrário do exemplo da figura 9, onde há uma relação bem definida entre cores e temperaturas para facilitar a interpretação, nas imagens de pseudo-cor normalmente não há uma ligação clara entre cores e elementos da imagem, como se pode verificar no exemplo da figura 11. O objectivo é simplesmente maximizar a percepção visual do conteúdo da imagem.

Um exemplo mais elaborado é apresentado na figura 12 para uma imagem de raio-X de inspeção de bagagens num aeroporto [**Gonzalez**]. A figura mostra a imagem original em escala de cinzento (em cima) e duas imagens de pseudo-cor (esquerda). Estas imagens foram criadas usando funções de transformação periódicas, do tipo sinusoidal (em módulo), com diferentes valores de fase para cada componente (figura 12, direita). A diferença entre as duas imagens de pseudo-cor resulta de se usar funções de transformação com períodos diferentes. Ou seja, pequenas variações de apenas 1 parâmetro produzem imagens de pseudo-cor bastante diferentes. Esta abordagem permite implementar um sistema onde o operador poderá ter um botão que ajusta um parâmetro (o período das funções) e observa de forma imediata a imagem de pseudo-cor.

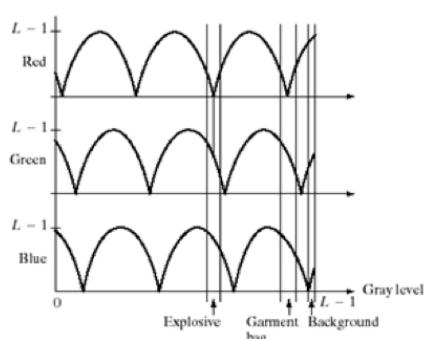
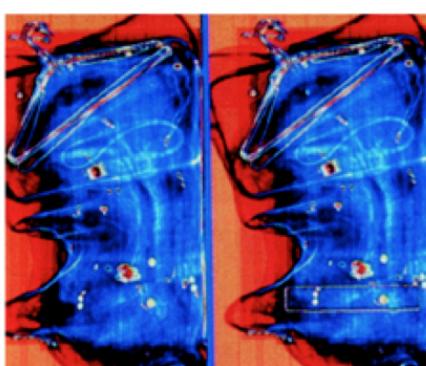
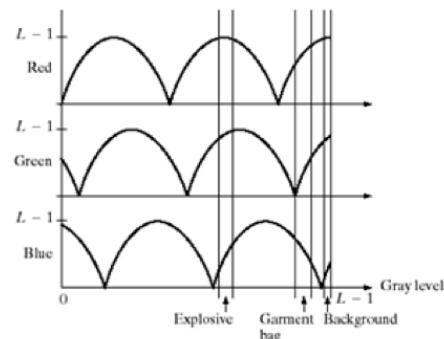
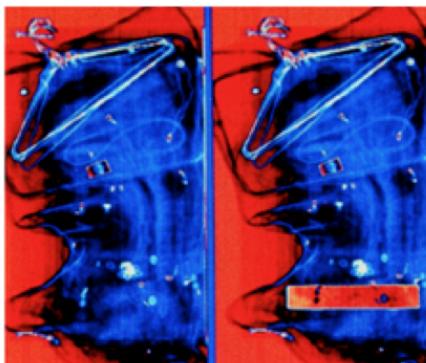
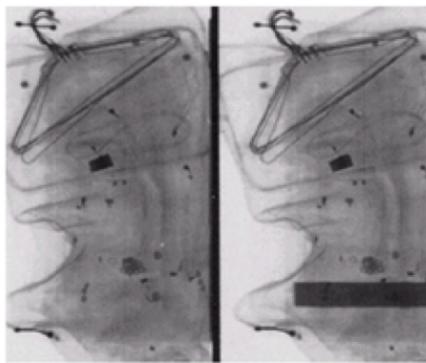


Figura 12: Uso de pseudo-cor para melhorar a interpretação de imagem raio-X de inspeção de bagagens em aeroporto. Imagem original em escala de cinzento (em cima); imagens de pseudo-cor (esquerda) e as respectivas funções de transformação usadas em cada caso (direita). [Gonzalez]

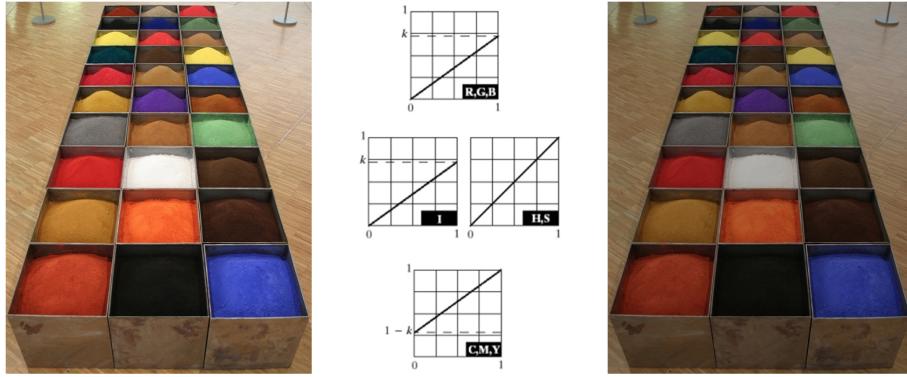


Figura 13: Exemplo de operação pontual aplicada a imagem de cor: original (esquerda); funções de transformação para redução de contraste com fator k , nos modelos RGB, HSI e CMY (centro); imagem final para $k = 0.7$ (direita).

5 Operações Pontuais em imagens de cor

As operações pontuais, apresentadas inicialmente para imagens de níveis de cinzento (1 banda), podem ser aplicadas em imagens de cor (com 3 bandas) ou em imagens multi-espectrais (com N bandas). No caso de imagens de cor, há modelos mais convenientes para certas transformações do que outros. Em operações pontuais simples, tais como alteração de contraste ou pequenas variação de luminosidade, não há grande diferença entre os vários modelos em termos de resultado final. No entanto, a forma de implementação depende do tipo de operação pontual e do modelo de cor usado. Considerando uma função de transformação f , para imagens RGB, os novos valores R_n, G_n, B_n são obtidos dos originais (R_o, G_o, B_o) através de (7). Ou seja, a função é aplicada a cada uma das componentes RGB. No caso de modelos onde a informação sobre a cor está desacoplada da informação sobre a intensidade, como os modelos HSV e HSI (2.3), a função é aplicada apenas à componente de intensidade. Por exemplo, para o modelo HSI, os novos valores H_n, S_n, I_n são obtidos dos originais (H_o, S_o, I_o) através de (8), sendo alterada apenas a componente I .

$$R_n = f(R_o) \quad G_n = f(G_o) \quad B_n = f(B_o) \quad (7)$$

$$H_n = H_o \quad S_n = S_o \quad I_n = f(I_o) \quad (8)$$

A figura 13 apresenta um exemplo para a variação de contraste por um fator k , com $k < 1$, ou seja redução de contraste. A figura mostra a imagem original (esquerda), a imagem final para $k = 0.7$ (direita), assim como as funções de transformação usadas para implementar esta transformação nos modelos RGB, HSI e CMY (centro). No caso do modelo CMY (2.2), uma vez que se trata de um modelo subtrativo, os novos valores numéricos das componentes C, M e Y passam a estar contidos no intervalo $[1-k, 1]$, em vez de $[0, k]$ como para R, G, B e I. Neste caso a imagem resultante (figura 13, direita) é praticamente igual actuando em qualquer um dos modelos (as diferenças são residuais).



Figura 14: Equalização de imagem de cor RGB. Imagem original (esquerda); imagens equalizadas no modelo RGB (centro) e no modelo HSV (direita).

Em geral o objectivo é aplicar uma operação pontual para melhorar a percepção visual da imagem, mas preservando dentro do possível as cores originais. Ao contrário do exemplo apresentado na figura 13, em alguns modelos de cor certas operações pontuais podem introduzir uma variação considerável nas cores observadas na imagem final, por comparação com a original. Por exemplo um aumento da luminosidade no modelo RGB (2.1) corresponde a somar um valor constante às componentes R,G,B podendo provocar saturação apenas numa das componentes, o que terá como efeito alterar a percepção de cor dos pixels afetados. Caso se aplique a mesma operação num modelo com as componentes de cor separadas da intensidade, como os modelos HSV e HSI (2.3), as cores originais são preservadas desde que não haja saturação da intensidade.

5.1 Equalização de histograma

A equalização de histograma é um caso onde há vantagens em usar modelos de cor onde a intensidade está separada das componentes de cor. Caso se faça a equalização de histograma de uma imagem de cor directamente no modelo RGB, as funções de transformação são obtidas de forma separada para cada componente, com base no histograma cumulativo normalizado. Ou seja, aplicam-se funções de transformação diferentes a cada componente de cor, o que resulta numa imagem que pode ter cores bastante diferentes da imagem original. A figura 14 mostra um exemplo de uma imagem de teste ('Serralves') com baixo contraste (esquerda) e o resultado da equalização aplicada no modelo RGB (centro). Como se pode observar, a imagem equalizada tem mais diversidade de níveis do que a imagem original, facilitando a visualização, mas as cores aparecem alteradas. Por exemplo na zona da fonte de água do lado esquerdo, que é esverdeada na imagem original, e no céu.

A equalização de histograma de imagens de cor nos modelos HSV e HSI (2.3), ou noutras em que a intensidade está desacoplada das componentes de cor, é feita aplicando a equalização apenas à componente associada à intensidade. Por exemplo, para o modelo HSI determina-se uma função de transformação f com base no histograma cumulativo normalizado de I , e aplica-se essa função a I mantendo-se as componentes H e S inalteradas (8). O resultado é uma imagem equalizada onde a componente cromática não é alterada. A figura 14 (direita) mostra a imagem 'Serralves' equalizada no modelo HSV. Pode observar-se que a imagem mantém as cores originais mas tem melhor utilização da gama de intensidades disponíveis. Esta imagem foi obtida convertendo a original RGB para HSV, actuando em V (equalização) e convertendo de HSV para RGB.

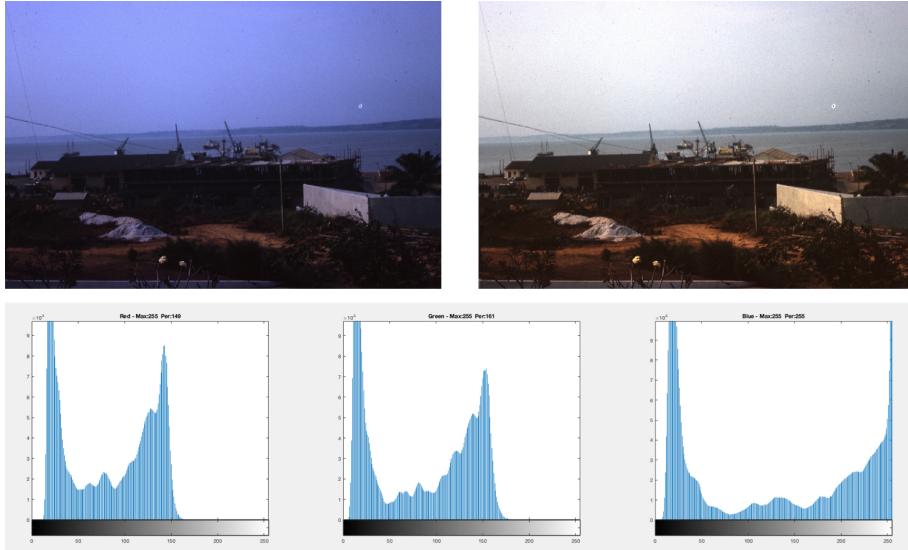


Figura 15: Exemplo de correção de cor de uma fotografia analógica (*slide*): imagem original (esquerda) e os correspondentes histogramas das componentes R, G e B (em baixo, da esquerda para a direita); imagem corrigida (direita).

5.2 Correção de cor

Uma outra tarefa de processamento de imagens de cor em que se recorre a operações pontuais é a correção de cor (*Color Balancing*). Neste caso não se pretende manter as cores originais mas sim modifica-las, normalmente com o objectivo de se ter uma imagem de cor mais realista. As correções podem ser feitas manualmente, aplicando transformações diferentes a cada componente de cor, ou de forma automática com base em alguns pressupostos. No modelo RGB a correção de cor pode ser feita através de variações de contraste diferentes para cada componente (9), sendo os parametros k_r , k_g e k_b obtidos de uma amostra da imagem com iluminação conhecida. Na prática é difícil ter essa informação, pelo que é habitual usar parâmetros globais. O método *scale-by-max* estima os valores de k como a razão entre o valor máximo possível (255 em imagens RGB 24 bits) e os máximos observados em cada componente. Uma alternativa mais segura é considerar percentis próximos de 100 em vez dos valores máximos.

$$\begin{bmatrix} R_n \\ G_n \\ B_n \end{bmatrix} = \begin{bmatrix} k_r & 0 & 0 \\ 0 & k_g & 0 \\ 0 & 0 & k_b \end{bmatrix} \begin{bmatrix} R_o \\ G_o \\ B_o \end{bmatrix} \quad (9)$$

A figura 15 (esquerda) mostra um exemplo de uma imagem obtida na década de 1960, por uma câmara analógica com pelicula positiva (*slide*), que necessita de correção de cor. Como se pode observar nos histogramas desta imagem (figura 15 em baixo), a quase totalidade dos pixels ocupam um intervalo de valores relativamente pequeno em R e G, mas não em B. No entanto, o valor máximo é 255 para as 3 componentes, pelo que se usou o percentil 98 para calcular os coeficientes k : $k_r = 255/149$, $k_g = 255/161$ e $k_b = 1$. O resultado do processo de correção de cor usando (9) é apresentado na figura 15 (direita).



Figura 16: Filtro de média aplicado a imagem de cor (a contar da esquerda):
1 - no modelo RGB; 2 - no modelo HSV (apenas a V); 3 - diferença absoluta entre as imagens 1 e 2; 4 - aplicação errada no modelo HSV (às 3 componentes).

6 Filtros Espaciais em imagens de cor

À semelhança das operações pontuais (5), os filtros espaciais também podem obviamente ser aplicados a imagens de cor. A forma de implementação mais eficaz depende do tipo de filtro e também do modelo de cor onde se actua. Considerando que se tem uma imagem RGB a 24 bits e se pretende aplicar um filtro espacial para suavização, por exemplo um filtro de média ou de mediana, há várias alternativas a considerar. Uma possibilidade é aplicar o filtro diretamente à imagem RGB, actuando-se em cada uma das 3 componentes de forma separada. A aplicação do filtro em modelos onde a cor está desacoplada da intensidade terá de ser feita com mais prudência. Por exemplo nos modelos HSV e HSI (2.3) o filtro deverá ser aplicado apenas à componente de intensidade (V, I). Em particular convém realçar que não é adequado calcular médias com medidas angulares, como é o caso da componente H (Hue).

A figura 16 mostra um exemplo de aplicação de um filtro espacial de média de 5×5 a uma imagem de teste com 580×478 pixels (original na figura 13). As duas primeiras imagens (a contar da esquerda) apresentam o resultado da aplicação do filtro nos modelos RGB (F_{RGB}) e HSV (F_{HSV}). Podem observar-se algumas diferenças entre as duas imagens, sendo notório que a suavização em F_{RGB} é ligeiramente mais intensa do que em F_{HSV} . As zonas de maior diversidade entre os dois resultados podem ser identificadas na imagem da diferença absoluta entre F_{RGB} e F_{HSV} (3^a imagem a contar da esquerda).

Na figura 16 (direita) é igualmente apresentado o resultado de aplicar o filtro de média às 3 componentes HSV, o que não faz sentido. Em algumas zonas desta imagem pode observar-se a presença de cores aberrantes, sem qualquer semelhança com as cores originais (por exemplo no caixote no canto superior direito). O que ocorre nessas zonas é que há pixels próximos (numa janela de 5×5) com valores em torno de $H=0^\circ$, positivos e negativos. Devido ao facto de H ser uma medida angular, com domínio $[0^\circ, 360^\circ]$ escalado para $[0,1]$, a média de dois valores correspondentes a vermelhos muito parecidos (e.g. 0.01 e 0.99) pode resultar numa cor totalmente diferente (e.g. 0.5, ciano).

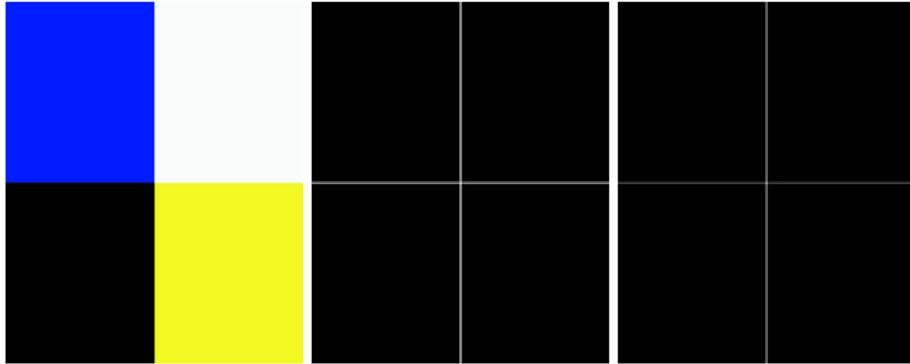


Figura 17: Detecção de contornos em imagem de cor: original RGB (esquerda); a partir dos gradientes no espaço 3D de cor (centro); somando os resultados do filtro espacial gradiente de cada componente RGB (direita). [GonzalezM]

6.1 Detecção de contornos

Os filtros espaciais de detecção de contornos podem ser usados em imagens de cor. Uma forma de o fazer é aplicando o filtro a cada componente de forma separada e combinar as 3 imagens obtidas. Caso se pretenda uma imagem com 1 banda (níveis de cinzento), pode somar-se as 3 imagens ou considerar o maior valor absoluto de cada resultado. Uma outra forma é criar uma imagem de contornos com 3 bandas (e.g. RGB), onde é possível distinguir contornos que estão relacionados com todas as cores (aparecem a cinzento) ou que são específicos de 1+ cores (aparecem com cor).

Uma abordagem alternativa é considerar o espaço de cor 3D, procurando o gradiente (direção e magnitude da variação) para cada pixel. Convém referir que o termo gradiente é usado de forma “livre” em Processamento de Imagem / Visão Computacional, uma vez que as imagens digitais tem variável discreta (espacial e radiométrica). Ou seja, na prática os gradientes são calculados com diferenças de variáveis discretas. No caso de uma imagem RGB, o espaço 3D é definido pelos versores \vec{r} , \vec{g} e \vec{b} . Para um dado pixel são calculados 2 vetores (\vec{u} e \vec{v}) associados às variações de cor RGB em linha e coluna (10).

$$\vec{u} = \frac{\partial R}{\partial x} \vec{r} + \frac{\partial G}{\partial x} \vec{g} + \frac{\partial B}{\partial x} \vec{b} \quad \vec{v} = \frac{\partial R}{\partial y} \vec{r} + \frac{\partial G}{\partial y} \vec{g} + \frac{\partial B}{\partial y} \vec{b} \quad (10)$$

A informação extraída de (10) pode ser usada para criar uma imagem tensorial. No entanto é também possível reduzir a informação a uma imagem simples, com 1 banda, que pode ser apresentada em níveis de cinzento (mais detalhes em **GonzalezM**). Na figura 17 é apresentado um exemplo de detecção de contornos numa imagem RGB (esquerda). A imagem resultante do processo descrito anteriormente (10) é apresentada na forma simplificada (centro). É igualmente apresentada uma imagem com contornos obtida com filtros espaciais (direita), usando o operador gradiente em cada componente e somando o resultado.

Podem igualmente ser aplicados filtros espaciais a imagens de cor para outras tarefas, tais como realce de contornos ou redução de ruído, sendo conveniente escolher para cada caso o modelo de cor e implementação mais adequada.

7 Imagem de cor no MATLAB

Nesta secção são apresentadas algumas funções e códigos relevantes para processamento de imagem de cor no MATLAB. A maior parte dos formatos *standard* (e.g. *jpg*, *tif*, *bmp*) usam cor RGB (2.1) a 24 bits, que é uma das estruturas mais comuns para imagens de cor. A importação da imagem para memória e separação das componentes R, G, B pode ser feita da forma indicada em seguida.

```
>> IM = imread('ArTe.jpg');
>> whos IM
  Name      Size      Bytes  Class      Attributes
  IM        580x478x3    831720  uint8
>> R=IM(:,:,1);
>> G=IM(:,:,2);
>> B=IM(:,:,3);
```

conteúdo do ficheiro na variável IM
para ver estrutura dados
todas as linhas e colunas
Nº banda

7.1 Conversão entre modelos de cor

Há várias funções MATLAB para conversão entre modelos de cor, tais como `rgb2HSV` e `HSV2RGB` entre RGB e HSV (2.3). A utilização destas funções é muito simples, como se ilustra em baixo para a conversão da imagem `IM` de RGB para HSV e subsequente extração das componentes H, S, V (figura 8).

```
>> IM2 = rgb2HSV(IM);
>> whos IM2
  Name      Size      Bytes  Class      Attributes
  IM2       580x478x3   6653760  double
>> H=IM2(:,:,1);
>> S=IM2(:,:,2);
>> V=IM2(:,:,3);
```

RGB para HSV
dados passaram a ser double
versão de IM a tons de cinzento

Há outras funções semelhantes para conversão entre RGB e outros modelos de cor, tais como: `rgb2xyz` (e `xyz2rgb`) para o modelo CIE xyz (2), `rgb2ycbcr` e `rgb2lab` para os modelos YCbCr e L*a*b* (2.4), etc. No caso do modelo CMY (2.2), pode ser usada a função `imcomplement`, com a seguinte sintaxe.

```
>> CMY = imcomplement(IM);
```

Há também funções de conversão de cor sem incluir o modelo RGB, tais como `lab2xyz` e `xyz2lab` para conversão entre os modelos CIE xyz e L*a*b*. A conversão de uma imagem RGB para imagem de tons de cinzento, com apenas 1 banda, pode ser feita extraíndo a componente V da versão HSV da imagem, ou de forma mais simples através da função `rgb2gray`, como se ilustra em baixo.

```
>> IM1 = rgb2gray(IM);
>> whos IM1
  Name      Size      Bytes  Class      Attributes
  IM        580x478     277240  uint8
(x1) apenas 1 banda
```

Colormap Name	Color Scale
parula	
jet	
hsv	
hot	
cool	
spring	
summer	
autumn	
winter	
gray	
bone	
copper	
pink	
lines	
colorcube	
prism	
flag	
white	

Figura 18: Tabelas de cor pré-definidas no MATLAB. [MathWorks]

7.2 Imagens de cor indexada

A apresentação de uma imagem de 1 banda é normalmente feita usando uma escala de níveis de cinzento, com preto para o valor mais baixo (e.g. 0) e branco para o valor mais alto (e.g. 1 ou 255). A função `imshow` atribui por omissão (*default*) esta tabela de cor (`gray`) a imagens de 1 banda. No entanto, pode ser escolhida uma outra tabela de cor, entre as pré-definidas no MATLAB (figura 18) ou uma criada pelo utilizador. O código em baixo mostra a imagem (`IM1`), primeiro com tons de cinzento e depois com a tabela de cor `cool`.

```
>> imshow(IM1);
>> figure
>> imshow(IM1,cool);
```

por omissão,
níveis de
cinzento

escala de
cores pré-
definida

Em certos casos pode haver alguma perturbação em usar mais do que 1 tabela de cor em imagens na mesma janela, pelo que poderá ser preferível apresentar as imagens em janelas separadas, como se fez no exemplo acima, recorrendo à função `figure`. As tabelas de cor pré-definidas tem na verdade apenas 64 cores, pelo que poderão não ser as mais adequadas para representar uma imagem de 1 banda. Podem no entanto ser criadas tabelas de cor com o número de níveis diferente do *default* (64), como se exemplifica em seguida.

```
>> TabCorX = cool(256);
>> imshow(IM1,TabCorX);
```

Tabela de
cor

Nº de cores

A variável `TabCorX` é uma matriz de 256x3 elementos, com as linhas a corresponder às intensidades da imagem de 1 banda (0 a 255) e as colunas às componentes R,G,B das cores associadas a cada nível.

A conversão de uma imagem RGB 24 bits para cor indexada é feita através da função `rgb2ind`. A função pode ser usada de várias formas, sendo a mais simples indicada em seguida, onde `IRGB` e `Iind` são as imagens RGB e indexada, `n` é o número de cores e `TabCor` a tabela de cor (matriz de $nx3$ elementos).

```
>> [Iind,TabCor] = rgb2ind(IRGB,n);
```

Um parâmetro de *input* adicional que pode ser relevante é a indicação se de uso ou não de *dither* (3). O código em seguida permite produzir as imagens apresentadas na figura 9.

```
>> IRGB = imread('Frindge.jpg');
>> [II32,TabCor32] = rgb2ind(IRGB,32,'nodither'); sem dither
>> [II8,TabCor8] = rgb2ind(IRGB,8,'nodither'); 8 cores
>> [II8d,TabCor8d] = rgb2ind(IRGB,8,'dither'); com dither
>> subplot(1,4,1), imshow(IRGB);
>> subplot(1,4,2), imshow(II32,TabCor32);
>> subplot(1,4,3), imshow(II8,TabCor8);
>> subplot(1,4,4), imshow(II8d,TabCor8d); tem de se
incluir a ta-
bela de cor
```

7.3 Processamento de imagens de cor

Muitas funções MATLAB da *Image Processing Toolbox* funcionam em imagens de cor RGB (3 bandas) da mesma forma que em imagens de cinzento (1 banda). No entanto, isso não acontece para todas as funções MATLAB. Em seguida apresentam-se alguns exemplos de operações pontuais em imagens de cor.

```
>> IMg1 = IM*1.2; aumento de
>> IMg2 = imadjust(IM,[],[],2); correção
>> IMg3 = imadjust(IM,[0.1 0.3 0.2 ;0.8 0.9 0.7],[],2); gama = 2
>> IMg4 = histeq(IM); limites inf.,
sup. input
```

No caso de equalização no modelo HSV (8), o código poderá ser o seguinte

```
>> IHSV = rgb2HSV(IM);
>> Inova = IHSV;
>> Inova(:,:,3) = histeq(IHSV(:,:,3));
>> IRGB = HSV2RGB(Inova);
>> Ifinal = uint8(255*IRGB); equalização
para manter
formato ori-
ginal uint8
```

A função `imfilter` que permite aplicar filtros espaciais lineares funciona igualmente para imagens de cor RGB. Por exemplo, o código em baixo aplica um filtro espacial de detecção de contornos e a imagem obtida (`IC`) é RGB de 24 bits, à semelhança da imagem original (`IM`)

```
>> ff=fspecial('laplacian');
>> IC=imfilter(IM,ff,'replicate');
```

No caso do filtro de mediana, a função usada para imagens de 1 banda (`medfilt2`) não permite operar em matrizes com 3 dimensões. Há no entanto uma versão da função (`medfilt3`) que opera em matrizes tridimensionais, como é o caso de imagens RGB.

8 Referências Bibliográficas

- Chang** Yu-Sung Chang, CIE Chromaticity Diagram, Wolfram Demonstrations Project, <http://demonstrations.wolfram.com/CIEChromaticityDiagram/>
- Gonzalez** Digital image processing, R.C. Gonzalez, R.E. Woods, 3rd Ed., Prentice Hall (2008).
- GonzalezM** Digital image processing using MATLAB, R.C. Gonzalez, R.E. Woods, S.L. Eddins, 2nd Ed., Gatesmark Publishing (2009).
- Hunt** Measuring Colour, R.W.G. Hunt, M.R. Pointer, 4th ed., John Wiley & Sons (2011).
- MathWorks** The MathWorks, MATLAB and Image Processing Toolbox, Release 2019a, Natick, Massachusetts, United States, MathWorks (2019) .
- Widen** ? Figura ? - Espectro Eletromagnético (?) .

9 Exercícios propostos

Nesta secção são propostos alguns exercícios para MATLAB que poderão ser usados para consolidar os conceitos de Processamento de Imagem apresentados.

I - Visualização de imagens de cor

Crie uma função MATLAB que receba como *input* o nome de um ficheiro imagem (formato standard) e mostre em 6 janelas distintas as seguintes imagens: (1) original RGB; (2,3) versões de cor indexada a 8 cores com/sem ‘dither’; (4) versão escala de cinzento (IG); (5,6) IG com tabelas de cor ‘jet’ e ‘spring’.

II - Componentes RGB e HSV

Crie uma função MATLAB que receba como *input* o nome de um ficheiro imagem (formato standard) e apresente 3 janelas com: (1) a imagem original; (2) as componentes RGB e os seus histogramas (`subplot` de 2×3); (3) as componentes HSV e os seus histogramas (`subplot` de 2×3).

III - Equalização nos modelos RGB e HSV

Crie uma função MATLAB que receba como *input* o nome de um ficheiro imagem (formato standard) e mostre numa única janela a imagem original (RGB) e as versões equalizadas através de modelos RGB e HSV (`subplot` de 1×3).

IV - Extração de assinatura de cor

Crie uma função MATLAB que receba como *input* o nome de um ficheiro imagem (formato standard) e extraia variáveis de uma zona retangular no centro da imagem com apenas 1% dos pixels (1/10 em linhas e colunas). Deverá ser apresentada uma janela com a imagem original, a sub-imagem utilizada, e as componentes R,G,B,H,S,V da sub-imagem (`subplot` de 2×4). Deverão igualmente ser apresentados os seguintes elementos para a sub-imagem: (1) média em R,G,B,H,S,V; (2) desvio padrão em R,G,B,H,S,V; (3) min. e max. de R,G,B,H,S,V.

V - Deteção de contornos em imagem RGB

Crie uma função MATLAB que receba como *input* o nome de um ficheiro imagem (formato standard) e apresente duas janelas. A primeira com 4 imagens: original (IO), contornos horizontais (CH), contornos verticais (CV), todos os contornos (`subplot` de 2×2). A segunda com 6 gráficos de intensidades: cortes horizontais e verticais a meio da imagem (linha e coluna central) para IO, CH e CV (`subplot` de 2×3).

9.1 APPENDIX - Exercices (in English)

This Appendix proposes some exercises for MATLAB that can be used to consolidate the concepts of Color Image Processing.

I - Color image visualization

Create a MATLAB function that receives as input an image file name (standard format) and shows the following images in 6 different windows: (1) original RGB; (2,3) indexed color versions with 8 colors with / without ‘dither’; (4) gray scale (IG) version; (5,6) IG with jet and spring color tables.

II - RGB and HSV components

Create a MATLAB function that receives as input an image file name (standard format) and presents 3 windows with: (1) the original image; (2) the RGB components and their histograms (2×3 subplot); (3) the HSV components and their histograms (2×3 subplot).

III - Equalization in the RGB and HSV models

Create a MATLAB function that receives as input an image file name (standard format) and shows in a single window the original image (RGB) and the equalized versions using RGB and HSV models (1×3 subplot).

IV - Color signature extraction

Create a MATLAB function that receives as input an image file name (standard format) and extracts variables from a rectangular area in the centre of the image with only 1% of the pixels (1/10 in rows and columns). A window should be displayed with the original image, the sub-image used, and the components R, G, B, H, S, V of the sub-image (2×4 subplot). The following elements should also be presented for the sub-image: (1) average in R,G,B,H,S,V; (2) standard deviation in R,G,B,H,S,V; (3) min. and max. in R,G,B,H,S,V.

V - Edge detection in RGB image

Create a MATLAB function that receives as input an image file name (standard format) and presents two windows. The first with 4 images: original (IO), horizontal edges (HE), vertical edges (VE), all edges (2×2 subplot). The second with 6 intensity graphics: horizontal and vertical profiles along the middle of the image (central line and column) for IO, HE and VE (2×3 subplot).