



INSTITUTO POLITÉCNICO DE BRAGANÇA Escola Superior de Tecnologia e de Gestão

Dispositivo Móvel para a Percepção Cromática em Deficientes Dicromáticos

Gilberto de Sousa Ferraz

Dissertação apresentada à Escola Superior de Tecnologia e de Gestão de Bragança para obtenção do Grau de Mestre em Sistemas de Informação

Orientado por

Pedro João Soares Rodrigues

**Bragança
2011**

Dispositivos Móveis para a Percepção Cromática em Deficientes Dicromáticos

Dissertação apresentada ao Instituto Politécnico de Bragança para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Sistemas de Informação, sob a supervisão do Professor Doutor Pedro João Soares Rodrigues

Gilberto de Sousa Ferraz

Outubro 2011

Dedicatória

*À memória do meu Pai, João Magalhães Ferraz
e meus Avós Francisco Inácio Ferraz e Judite Lima Ferraz*

Agradecimentos

À minha mãe, Maria Fernanda de Sousa, pelo apoio que me têm dado ao longo de toda a minha vida.

À minha namorada Tânia Pires, pelo apoio, incentivo e paciência que me deu ao longo da elaboração desta tese.

Ao Professor Doutor Pedro João Soares Rodrigues, o meu agradecimento pela possibilidade de realizar esta tese ao lado de alguém que transpira sabedoria; a minha admiração pela sua serenidade, pelo dom no ensino da ciência, capacidade de análise, inibindo sempre a vaidade em prol da simplicidade e eficiência.

Abstract

The inability of perceive all the color spectrum is a deficiency known as color blindness. Generally, this kind of disease is caused by anomalies in the retinal cells or by their absence. These cells are photoreceptors that show sensitivity to three different spectral light bands. In this thesis specific color blindness type, known as Dichromacy, is treated. The Dichromacy is a color vision defect in which, one of the three basic color mechanisms is lacking or not functioning. It is the most significant colorblindness disease, having in account the incapacity caused and knowing that it affects 3% of the world population. Protanopia, Deuteranopia and Tritanopia are Dichromacy subclasses, each one affecting the perception of one of the tree primary colors.

This disease is normally caused by a genetic source and has no treatment, neither based on drugs nor based on surgery. It causes important limitations on the daily tasks execution and can lead to several danger situations. For instance, buying fruits at the supermarket, choosing clothes at the shop or in the traffic identifying the light colors.

There are several image processing algorithms that allow minimizing this problem. The used methods are able to remap the original colors, which are not perceived by the color blindness people, to the colors that are recognized by them. This is done trying not modifying the original visual semantic. The most important methods that are able to perform those transformations will be presented and discussed in this thesis, especially in what concerns to their advantages and disadvantages to the image recognition by the color blind person.

Beyond the algorithms, a practical approach to the use of those techniques in mobile devices is showed. Their ability to capture images (e.g. phone devices) can improve the everyday life of the color blindness people and enrich their perception of reality.

The Human visual system is a very complex functional structure. From the retinal eye cells to the visual cortex layers one can find a large number of physical phenomena contributing to the performance of this mechanism. This fact makes the image processing tasks demand for large computational power.

The usually methods are able to map the pixels of the original image to the corrected image concerning the color blindness spectral range. This process is supported by an optimization task that allows keeping the visual semantic and, at the same time, increases the number of perceived regions. However, these methods do not consider the spatial relation among the color pixels, but only the relation among the color pixels values. This fact, in some images, leads to an incorrect mapping, joining some regions that are separated in the original image. The process presented in this thesis pretends to overcome this issue increasing the number of visual situations that this type of algorithms is able to treat. The method is based on a texture analysis process to demarcate the different colors of an image, thereby ensuring that the mappings are not made to existing colors in the original image, which are belonging to adjoining regions.

Resumo

Estamos tão acostumados ao nosso sistema visual que raramente nos apercebemos da complexidade envolvida numa simples tarefa como abrir os olhos e visualizar o que nos rodeia. Apesar de usarmos este sentido todos os dias, a visão é na realidade um fenómeno bastante complicado e o olho é provavelmente um dos órgãos mais complexos do corpo humano.

Estudos recentes indicam que cerca de 8% da população europeia apresenta anomalias ao nível de algumas das células (cones) da retina. A nossa ferramenta primária de visão são os olhos, são eles também um instrumento biológico de captura que, como uma simples câmara fotográfica, recolhe, foca e processa a luz. A incapacidade do processo visual diferenciar todas ou algumas cores é denominada de daltonismo; a ausência de um grupo de cones sensíveis aos três comprimentos de onda (Longo, Médio e Curto) é chamado de dicromância e pode apresentar-se sob a forma de protanopia, deuteranopia e tritanopia, respectivamente. Esta perturbação tem normalmente origem genética, ocorre mais frequentemente nos indivíduos de sexo masculino, pois está directamente ligado a uma anomalia no cromossoma X. Até à data não existe cura nem com a ajuda de fármacos nem com intervenções cirúrgicas. Torna-se então evidente desenvolver algoritmos (soluções) e implementá-los em mecanismos que possam melhorar o dia-a-dia destas pessoas. A forte penetração de dispositivos móveis (p.e. telemóveis, PDAs) nos mercados mundiais, equipados com sistemas de captação de vídeo e imagem e com processadores mais rápidos, fazem deles uma opção estratégica, pois acompanham, também eles, as pessoas no seu quotidiano. As tentativas de aumentar a complexidade e o desempenho funcional dos algoritmos de correcção daltónica deparam-se com as limitações ao nível da sua implementação física, este problema agrava-se quando implementados em dispositivos móveis. A dissertação actual apresenta dois métodos de correcção de imagens para deficientes de percepção da cor, no âmbito do processamento de imagens, que permitem, por um lado, manter as cores naturais da imagem original e, por outro lado, tornar perceptíveis as cores que são confundidas. No primeiro método, recorrendo ao algoritmo proposto por Giovane R. Kuhn em 2008, são usados processos de quantização da imagem original e recorre posteriormente a analogias físicas (*mass-spring*) para manter a distância original entre as cores. No segundo método é sugerido um processo de análise de

textura para demarcar as diferentes regiões de uma imagem garantindo assim que os mapeamentos não serão feitos para cores, espacialmente adjacentes, já existentes na imagem original.

Conteúdo

1. Introdução.....	19
1.1 Enquadramento	19
1.2 Objectivos propostos.....	20
1.3 Metodologia usada.....	21
1.4 Contribuições da tese	22
1.5 Organização do documento.....	22
2 Daltonismo.....	23
2.1 Definição de daltonismo	23
2.2 Tipos de daltonismo	27
2.3 Causas do daltonismo	28
2.4 Espectro electromagnético visível	30
2.5 O mundo visto por um dicromata	33
2.6 Resultados/conclusões	34
3 Estado da Arte	35
3.1 Técnicas de Simulação de daltonismo	35
3.2 Técnicas de correcção daltónica	39
3.2.1 Técnicas de re-coloração que requerem intervenção do utilizador.....	39
3.2.2 Técnicas automáticas de re-coloração de imagens que não requerem intervenção dos utilizadores	41
3.2.3 Operações pontuais	44
3.2.4 Métodos de optimização.....	44
4 Processamento Digital de Imagens.....	45
4.1 Geração de imagens digitais	46
4.2 Espaços de cores	46
4.2.1 RGB.....	46
4.2.2 CMY.....	47
4.2.3 HSV	47
4.2.4 CIE L*a*b	48
4.3 Amostragem espacial e quantização de níveis.....	49
4.4 Quantização de imagens – redução de cores.....	50
4.4.1 Algoritmos de quantização.....	52
4.4.1.1 Algoritmos de popularidade.....	52
4.4.1.2 Quantização uniforme.	53
4.4.1.3 Quantização não uniforme.....	53
4.4.1.4 Hierárquicos divisivos.....	54
4.4.1.5 Hierárquicos aglomerativos.....	55

5	Daltonização por <i>mass-spring</i>	56
5.1	Conceitos básicos	56
5.2	Sistema “ <i>mass-spring</i> ”	58
5.3	Dinâmica do sistema “ <i>mass-spring</i> ”	59
5.4	Algoritmo de recoloração de imagem	60
5.5	Modelação do sistema <i>mass-spring</i>	61
5.6	Resultados/conclusões	65
6	Algoritmo proposto	67
6.1	Conceitos básicos	67
6.2	Quantização da imagem original em N cores	68
6.3	Determinação de cores vizinhas.....	70
6.3.1	Matriz de co-ocorrência	70
6.3.2	Distância euclidiana.	74
6.4	Adaptação do sistema <i>mass-spring</i> ao algoritmo proposto.....	76
6.5	Dinâmica das molas (componente 1).....	78
6.5.1	Comprimento das molas nos pares de cores vizinhas.	78
6.5.2	Dinâmica do coeficiente de elasticidade das molas	78
6.5.3	Cálculo das forças	79
6.5.4	Resultados/conclusões.....	80
6.6	Afastar cores vizinhas (componente 2).....	82
6.6.1	Vector distância.....	82
6.6.2	Módulo da distância	83
6.6.3	Versor distância.....	83
6.6.4	Distância a incrementar	83
6.7	Resultados/conclusões	85
7	Resultados	89
8	Conclusões	94
8.1	Trabalhos Futuros	95
9	Apêndice A	1
A.1	Simulação Dicromancia	1
A.2	Algoritmo de Proposto por Kuhn.....	3
A.3	Algoritmo Proposto componente 1 e 2	9
A.3.1	Final.....	9
A.3.2	Matriz de co-ocorrência	23
A.3.3	Afastar cores vizinhas	25

Lista de Tabelas

Tabela 1 - Cronograma da tese.....	21
Tabela 2 - Incidência de daltônicos na população europeia [Rigden 1999].....	28
Tabela 3 - Sensibilidade dos comprimentos de onda medida em nanometros (nm).....	31
Tabela 4 - Imagem na forma de uma matriz de índices de cores $N \times M$ onde N representa a largura e M a altura da imagem.....	70

Lista de Figuras

Figura 1 - Diagrama do olho em corte longitudinal [1].	24
Figura 2 - A estrutura celular da retina [2].	26
Figura 3 - Transmissão genética do daltonismo [3].	29
Figura 4 - Espectro de luz visível [4].	30
Figura 5 - Aproximação das curvas de sensibilidade relativa do olho humano [5].	30
Figura 6 - Percepção dos 3 tipos diferentes de dicromancia [6].	33
Figura 7 - Aproximação de gama de cores plano [Kuhn et al. 2008].	34
Figura 8 - Diagrama de Goethe na primeira placa de Zür Farbenlehre (teoria das cores) [7].	36
Figura 9 - Cores do cubo RGB projectadas no plano de protanopes ao longo do eixo L [8].	37
Figura 10 - Paleta de cores proposta por [Vienot et al. 1999].	39
Figura 11 - Imagem corrigida usando o algoritmo proposto em [Wade 2008].	40
Figura 12 - Estrutura do modelo proposto por Yu Ma.	42
Figura 13 - Rede SOM utilizada neste método [Ma, Gu and Wang 2009].	42
Figura 14 - Representação abstracta de uma imagem digital.	45
Figura 15 - Espaço de cores $L^*a^*b^*$ [9].	49
Figura 16 - Exemplo de uma imagem contínua antes e depois do processo de amostragem e quantização [10].	49
Figura 17 - Dimensão de uma imagem (Estig) [11].	50
Figura 18 - Sequência de efeitos da quantização [10].	51
Figura 19 - Histograma de cores (exemplo).	52
Figura 20 - Exemplo bidimensional da quantização uniforme [13].	53
Figura 21 - Exemplo bidimensional da quantização não uniforme [13].	54
Figura 22 - Histograma equalizado por corte mediano.	54
Figura 23 - Esquema de partição em octree [14].	55
Figura 24 - Sistema <i>mass-spring</i> composto por cinco partículas (A,B,C,D e E) e oito molas (S) com diferentes comprimentos e estados de repouso [15].	59
Figura 25 - Representação de uma imagem de 128 cores quantizadas no espaço de cores $L^*a^*b^*$ [12].	61
Figura 26 - Exemplo de partículas $q(i)$, no espaço $L^*a^*b^*$, percebidas por tricromatas e interligadas por molas.	61
Figura 27 - Partículas P_i alinhadas em b^* no plano $L^*a^*b^*$ [12].	62
Figura 28 - Correção baseada no método de Kuhn segundo os testes de Ishihara [6].	65
Figura 29 - Imagem da esquerda: Representação no espaço de cores Lab; Imagem do centro: Correção Kuhn; Imagem da direita: Teste Ishihara [12].	65
Figura 30 - Correção <i>mass-spring</i> Kuhn (árvores [10]).	66
Figura 31 - Representação espaço de cores $L^*a^*b^*$, Correção Kuhn exemplo jardim.	66
Figura 32 - Exemplo de redução de cores [10].	69
Figura 33 - Redução e indexação para 16 níveis de cor da figura 32.	69
Figura 34 - Máscara de amostragem dos pixéis que compõem uma imagem com a identificação do pixel de interesse (ao centro) e os pixéis de adjacência.	71
Figura 35 - Índices (aplicação da máscara de amostragem) de cores da imagem quantizada.	72
Figura 36 - Matriz de co-ocorrências.	72
Figura 37 - Anulação da vizinhança de índices de cores iguais da MCO.	73
Figura 38 - Representação de 2 pontos no espaço tridimensional.	75
Figura 39 - Cores vizinhas da figura 32.b).	76

Figura 40 - Conjunto de imagens geradas pelo algoritmo proposto para a componente 1	81
Figura 41 - Exemplo das partículas no processo de afastamento de duas cores vizinhas.....	84
Figura 42 - Exemplo prático do afastamento das cores.	85
Figura 43 - Conjunto de imagens geradas pelo algoritmo proposto pela componente 1.	86
Figura 44 - Conjunto de correcções "daltónicas" para os vários algoritmos.....	91

Lista de Siglas

DVC	Deficiente na Visualização da Cor
SVH	Sistema Visual Humano
MCO	Matriz de Co-Ocorrência
RGB	Red Green Blue
CMY	Cyan Magenta Yellow
HSV	Hue Saturation Value
CIE	Commission Internationale d'Eclairage
SOM	Self-Organizing Map
W3C	World Wide Web Consortium
GPU	Graphics Processing Unit

Capítulo 1

1. Introdução

Cerca de 8% da população europeia apresenta anomalias ao nível de algumas das células (cones) da retina, o que se traduz pela incapacidade de perceber visualmente gamas específicas de cor. Isto provoca algumas dificuldades na realização de elementares tarefas do dia-a-dia. Por exemplo, seleccionar fruta num mercado ou escolher uma peça de roupa num pronto-a-vestir. O processo de mapeamento das cores, que não são possíveis de perceber, para as cores percepcionáveis é referido por *daltonização*. Pretende-se que sejam implementados e adaptados algoritmos para tornar eficiente o referido processo em dispositivos móveis. Assim, regiões distintas, que não são possíveis de discriminar, passam, por *daltonização*, a serem diferenciáveis. Para que o processo seja suficientemente eficiente é necessário que o algoritmo de mapeamento seja generalista na sua própria construção. Neste trabalho serão estudadas, adaptadas e aplicadas técnicas de *daltonização* para que sejam integradas em dispositivos móveis.

1.1 Enquadramento

A constante evolução tecnológica têm introduzido novos conceitos e novas estratégias de resoluções de problemas. A área do processamento digital de imagem é certamente uma área em crescimento que pode ajudar na resolução de variadíssimos problemas entre os quais a correcção daltónica de imagens para deficientes na visualização das cores (DVC).

O processamento de imagens está directamente ligado ao processamento de sinais. Os sinais como as imagens estão, na realidade, num suporte físico que porta no seu interior uma determinada informação.

Processar uma imagem consiste em transformá-la sucessivamente com o objectivo de extrair e manipular a informação nela presente. É adequado, neste momento, fazer uma comparação entre Processamento de Imagem e Computação Gráfica, técnica que se encontra frequentemente aplicada a sequências animadas na televisão ou em filmes de cinema. Geralmente, a Computação Gráfica parte de uma informação precisa para obter uma imagem ou um filme. O Processamento de Imagem parte de uma imagem geralmente captada por uma câmara para obtermos a informação. Deste ponto de vista o Processamento de Imagens e a Computação Gráfica são exactamente filosofias opostas, no entanto, não quer isso dizer que as técnicas usadas em cada caso não possam ser as mesmas ou, pelo menos, complementares.

É evidente que no sentido de processar uma imagem, como é feito pelo sistema visual humano (SVH), é extremamente complexo. Realizar as mesmas tarefas que o sistema visual, com a ajuda de máquinas, exige a compreensão amadurecida “filosófica” do mundo ou dos conhecimentos humanos. Esta característica complexa faz com que o processamento de imagens seja, actualmente, uma área com extrema dependência nos limites do conhecimento. Daí, a não existência, até ao momento, de sistemas de análise de imagens complexos e que funcionem para todos os casos de aplicação.

1.2 Objectivos propostos

No presente trabalho pretendemos implementar e adaptar algoritmos de correcção daltónica, recorrendo ao processamento digital de imagens. Pretende-se colmatar a perda de informação adquirida pelo sistema visual humano (SVH), proporcionando uma imagem o mais natural possível e rica em detalhes. Detalhes esses que não são percebidos pelo portador da deficiência de visualização de cores.

Tendo em conta a elevada taxa de penetração de dispositivos móveis (e.g. telemóveis, PDA), quer a nível nacional, quer a nível internacional, e o aparecimento de equipamentos cada vez mais avançados tecnologicamente, capazes de transportar no bolso, poder-se-á afirmar que estes objectos poderão ser um auxílio ao dia-a-dia de pessoas portadoras de daltonismo.

O objectivo principal da presente dissertação é desenvolver, adaptar e aplicar algoritmos de correcção de imagens em tempo real, capazes de serem executados em dispositivos móveis.

A aplicação deve permitir captar a imagem em tempo real (vídeo) através de um dispositivo móvel equipado com câmara, e proporcionar ao daltónico uma imagem alterada mas o mais aproximada possível da original. Deste modo, o daltónico terá uma percepção das cores que, pela visão do seu espectro visível não teria.

1.3 Metodologia usada

Esta tese foi dividida em 7 fases:

Tabela 1 - Cronograma da tese

Ano	1	2	3	4	5	6	7	8	9	10	11	12
2009												
2010												
2011												

	Pesquisa bibliográfica sobre daltonismo
	Estudo do processamento digital de imagens
	Pesquisa do estado da arte
	Pesquisa de ferramentas e linguagens de programação
	Desenvolvimento, testes e implementação em Matlab do algoritmo proposto
	Programação/adaptação em dispositivos móveis
	Elaboração do Relatório

1.4 Contribuições da tese

A contribuição desta tese inclui:

1. Sugestão de algoritmos de correcção daltónica baseados no modelo de “*mass-spring*”.
2. Desenvolvimento de uma aplicação desktop que permite simular em tempo real a visão de daltónicos com dicromatismo.
3. Desenvolvimento de uma aplicação em J2ME de um algoritmo de daltonização de imagens em tempo real.

1.5 Organização do documento

Este documento está organizado em oito capítulos.

Depois do capítulo dedicado à introdução, aborda-se com alguma profundidade a definição e as causas do daltonismo. O capítulo três destina-se ao balanço do actual “estado da arte” do tema em questão. No quarto capítulo fazemos uma introdução aos conceitos básicos de processamento digital de imagens. No quinto capítulo apresentamos o algoritmo proposto por Kuhn que usa a modelação “*mass-spring*”. No sexto capítulo apresentamos com detalhe a nossa proposta para o novo algoritmo de reestruturação de cor em imagens. No sétimo capítulo fazemos uma série de demonstrações com as técnicas estudadas nesta dissertação. No oitavo e último capítulo falamos da conclusão e propomos trabalhos futuros no âmbito do tema da tese.

Capítulo 2

2 Daltonismo

2.1 Definição de daltonismo

As crianças usam os olhos para interagir com o mundo que as rodeia, observam objectos coloridos que chamam a sua atenção, reconhecem rostos familiares e acompanham os movimentos das pessoas. Este tipo de comportamento indica que acontece um processamento complexo da informação e não simplesmente uma resposta aos estímulos apresentados.

Torna-se por isso necessário neste momento fazer a distinção entre ver e perceber. Ver é um processo passivo de obter informação sensorial visual, enquanto que perceber possui a componente de reconhecimento e compreensão. Ou seja podemos ver, porém podemos não perceber.

Estudos recentes indicam que uma em cada doze pessoas apresenta anomalias ao nível de algumas das células (cones) da retina. A nossa ferramenta primária de visão são os olhos, eles também são um instrumento biológico de captura que como uma simples câmara fotográfica recolhe, foca e processa a luz. A incapacidade da percepção visual de diferenciar todas ou algumas cores é denominada de daltonismo, a ausência de um grupo de cones sensíveis aos três comprimentos de onda (Longo, Médio e Curto) é chamado de dicromatismo e pode apresentar-se sob a forma de protanopia, deuteranopia e tritanopia, respectivamente. Esta perturbação tem normalmente origem genética, ocorre mais frequentemente nos indivíduos de sexo masculino, pois está directamente ligado a uma anomalia no cromossoma X. Até à data não existe cura nem com a ajuda de fármacos nem com intervenções cirúrgicas.

Torna-se então necessário desenvolver algoritmos e implementá-los em mecanismos que possam melhorar o dia-a-dia destas pessoas. A forte penetração de dispositivos móveis (e.g. telemóveis, PDA, smatphones, Etc...) nos mercados mundiais, equipados com sistemas de

captação de vídeo e imagem e com processadores mais rápidos, fazem deles uma opção estratégica, pois acompanham, também eles, as pessoas no seu quotidiano. As tentativas de aumentar a complexidade dos algoritmos de correcção daltónica deparam-se com as limitações a nível da sua implementação física; este problema agrava-se quando os algoritmos são implementados em dispositivos móveis. Apesar da evolução de capacidade de processamento destes equipamentos, estes são ainda bastante limitados em cálculos matemáticos necessários para a aplicação de algoritmos de correcção daltónica, o que torna a execução de aplicações locais mais lentas, exigindo assim alguma adaptação dos algoritmos propostos.

As câmaras fotográficas possuem lentes que concentram e focam a luz, um obturador que controla a quantidade da luz que entra e uma área sensível aos fotões que funciona como um registo de estado sensível à luz. A analogia dos olhos com uma simples máquina fotográfica é muito simplista, considerando o sofisticado sistema visual que os olhos representam.

A retina é a estrutura fundamental do olho localizando-se na camada mais interna. É composta por um tecido fino constituída por células fotossensíveis que detectam e respondem à entrada de luz e por uma rede de neurónios que processam os impulsos luminosos. Estes últimos usam o nervo óptico, para encaminhar as informações ópticas para o cérebro.

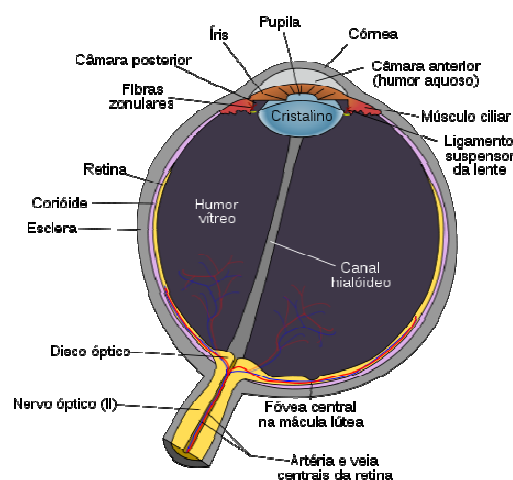


Figura 1 - Diagrama do olho em corte longitudinal [1].

Existem dois tipos de células, os bastonetes e os cones. Os primeiros são responsáveis por captar a percepção de contraste, actuam com baixa luminosidade e participam na detecção de movimentos. Os cones funcionam apenas quando há luz suficiente e são responsáveis pela visão detalhada e pela percepção apurada das cores. Uma ausência ou deficiência destas células provocam a percepção deficiente de cores.

O daltonismo é uma perturbação da percepção visual caracterizada pela incapacidade de diferenciar todas ou algumas cores, manifestando-se muitas vezes pela dificuldade de distinguir a cor verde da cor vermelha. Esta perturbação tem normalmente origem genética, mas pode também resultar de lesão nos órgãos responsáveis pela visão, ou de lesões de origem neurológica.

John Dalton, químico e físico inglês, foi o primeiro cientista a estudar a anomalia de que ele mesmo era portador, a doença recebeu o nome em homenagem a ele. Uma vez que esse problema está geneticamente ligado ao cromossoma X, ocorre mais frequentemente entre homens (no caso das mulheres, será necessário que os dois cromossomas X contendam o gene anómalo).

Os portadores do gene anómalo apresentam dificuldades na percepção de determinadas cores primárias, como o vermelho e o verde, o que se repercute na percepção das restantes combinações de cores primárias. Esta perturbação é causada pela ausência ou pelo menor número de alguns tipos de cones (figura 2) ou por uma perda de função parcial (tricromáticos anómalos) ou total destes (daltónicos). Esta perda está normalmente associada à diminuição de pigmentos nos fotoreceptores que deixam de ser capazes de absorver a informação luminosa de cor.

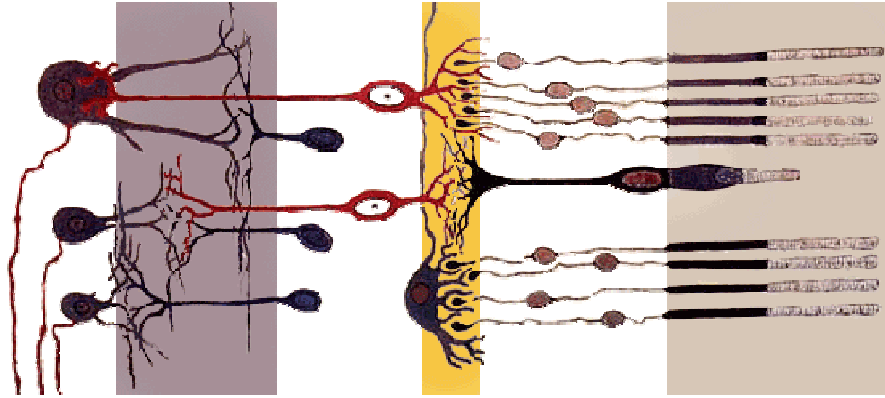


Figura 2 - A estrutura celular da retina [2].

Os cones são as células do olho humano que têm a capacidade de reagir às cores, segundo a teoria tricromática (teoria de Young-Helmholtz). Já os bastonetes, outro tipo de célula do olho humano, tem a capacidade de reagir à luminosidade. Existem aproximadamente 6 milhões de cones em cada olho humano concentrados na região fóvea. Sendo estes os responsáveis pela percepção das cores, quando existe uma anomalia ou ausência de algum dos fotopigmentos nas terminações dos cones estamos na presença de daltonismo.

A retina humana possui três tipos de células sensíveis à cor. Cada uma delas é sensível a uma determinada faixa de comprimento de onda do espectro eletromagnético visível.

Indivíduos com visão normal (tricromáticos) têm 3 tipos de cones normais chamados de vermelho, verde e azul, que diferem uns dos outros por possuírem fotopigmentos que são sensíveis às altas, médias e baixas frequências, respectivamente, do espectro eletromagnético visível.

Alterações nos fotopigmentos dos cones, provocam sensibilidade a uma faixa diferente do espectro visível, quando comparado com pessoas de visão normal. Estes indivíduos são chamados de tricromatas anormais; quando um destes tipos de cones está em falta os indivíduos denominam-se de daltônicos. A condição mais rara é caracterizada por indivíduos que possuem um único ou mesmo nenhum tipo de cone, que são chamados de monocromatas.

Um indivíduo é daltônico quando um dos pigmentos, nos cones, está em falta, normalmente o vermelho ou o verde, por vezes o azul, mas este último muito raramente acontece.

2.2 Tipos de daltonismo

Podemos considerar que existem três grupos de Discromatopsias: Tricomancias Anómalas, Dicromancias e Monocromancias.

Tricomancias Anómalas – Resultam de uma mutação nos pigmentos dos fotorreceptores dos cones situados na retina e manifestam-se em três anomalias distintas:

- Protanomalia - Presença de uma mutação no pigmento sensível às frequências mais longas (L) correspondente aos cones vermelhos. Resulta numa menor sensibilidade ao vermelho e num escurecimento das cores perto das frequências mais longas, podendo levar à confusão entre vermelho e preto.

- Deuteranomalia - Presença de uma mutação no pigmento sensível às frequências médias (M) correspondente aos cones verdes. Resulta numa maior dificuldade em discriminar o verde.

- Tritanomalia - Presença de uma mutação no pigmento sensível às frequências curtas (S) correspondente aos cones azuis. É a forma mais rara, que impossibilita a discriminação de cores na faixa azul-amarelo.

Dicromancia – Resulta da ausência de um tipo específico de cones, pode apresentar-se sob a forma de:

- Protanopia - Ausência de cones sensíveis às frequências de comprimento de onda longo correspondente aos cones vermelhos; impossibilidade de discriminar cores no segmento verde-amarelo-vermelho do espectro.

- Deuteranopia - Ausência de cones sensíveis às frequências de comprimento de onda médio correspondente aos cones verdes; impossibilidade de discriminar cores no segmento verde-amarelo-vermelho do espectro.

- Tritanopia - Ausência de cones sensíveis às frequências de comprimento de onda curto correspondente aos cones azuis; impossibilidade de discriminar cores no segmento azul-amarelo do espectro.

Monocromancia – É o tipo de daltonismo mais raro em que há uma cegueira completa de cores, não têm subcategorias porque o “mundo” é visto a tons escuros e claros. Existe uma ausência total de dois ou de mais cones sensíveis às frequências de comprimento longo, médio e curto.

Oito por cento dos homens da população europeia têm deficiência visual de cores. Esta é composta por tabela (2):

Tabela 2 - Incidência de daltônicos na população europeia [Rigden 1999].

Classificação	Incidência %	
	Masculino	Feminino
Protanomalia	1 %	0.4 %
Deuteranomalia	4.9 %	
Tritanomalia	-	
Tricromáticos anómalos	5.9 %	
Protanopia	1 %	
Deuteranopia	1.1 %	
Tritanopia	0.002 %	
Daltônicos/Dicromancia	2.1 %	
Monocromáticos	0.003 %	

Relativamente às mulheres, apenas 0.4% têm algum tipo de deficiência de visão de cores, tendo a maioria delas problemas relacionados com o comprimento de onda Longo (L) e Médio (M) do espectro visual.

2.3 Causas do daltonismo

Como se pode verificar na tabela (2), a incidência de daltônicos na população europeia é muito maior no sexo masculino que no feminino, isto resulta de uma falha genética que está directamente associada ao cromossoma X.

O Daltonismo é provocado por genes recessivos localizados no cromossoma X sem alelos no Y. Uma vez que o homem é portador de um cromossoma X e de um Y, enquanto que a

mulher é portadora de dois cromossomas X, se a alguém do sexo masculino a mãe lhe passar um alelo em X que codifica o daltonismo (gene recessivo), a doença do daltonismo vai-se manifestar.

Já nas mulheres, a condição para ser daltónica depende da passagem dos seus progenitores de dois X (com gene recessivo). Denotar ainda que as mulheres podem ser portadoras do gene anómalo que provoca o daltonismo mas este não se manifestar já que o cromossoma não portador da doença compensa o “defeito”.

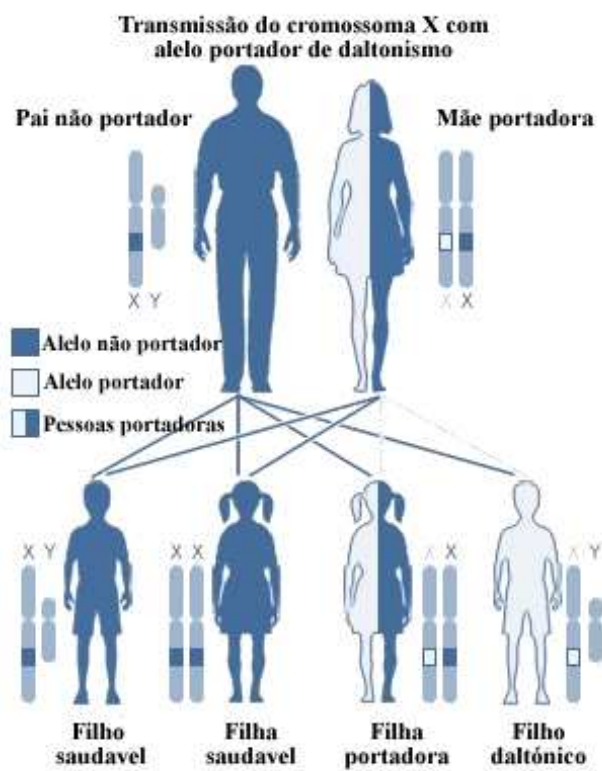


Figura 3 - Transmissão genética do daltonismo [3].

2.4 Espectro electromagnético visível

A visão normal a cores é denominada de tricromática, é obtida pela absorção de fótons por três tipos de cones, cuja sensibilidade de pico se encontra nos comprimentos de onda longo (L), comprimento de onda médio (M) e comprimento de onda curto (S) das regiões do espectro visível (figura (4)).

As formas reduzidas de visão de cores surgem a partir da ausência efectiva/anomalia de um dos fotopigmentos da retina do olho, do tipo L, M ou S.

O espectro electromagnético é o intervalo completo da radiação electromagnética, que contém desde as ondas de rádio, as microondas, os infravermelhos, a luz visível os raios X, até aos raios ultravioletas.



Figura 4 - Espectro de luz visível [4].

Maxwell, físico e matemático Britânico, concluiu que a luz visível é constituída por diferentes bandas espectrais que são ondas electromagnéticas, semelhantes umas às outras, com diferenças nas suas frequências.

Os diferentes comprimentos de onda estão directamente ligados à percepção da cor feita pelo olho humano - figura (5).

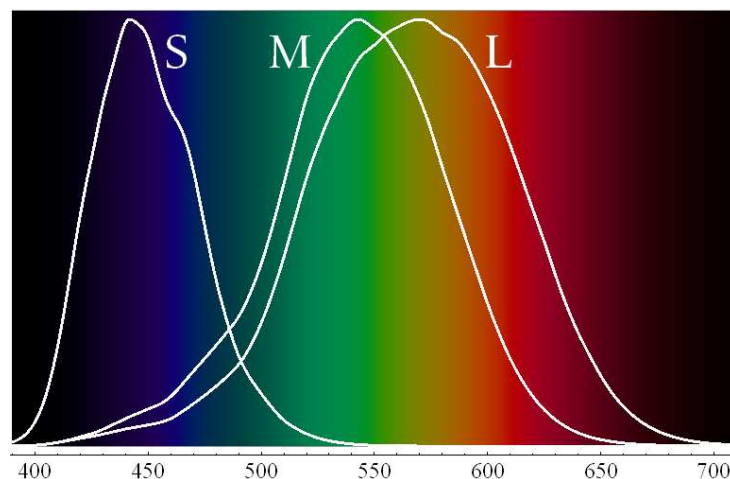


Figura 5 - Aproximação das curvas de sensibilidade relativa do olho humano [5].

Existem no entanto, duas teorias de visão de cores, a teoria tricromática e a teoria do processo oponente. A teoria tricromática afirma que a retina do olho humano possui três tipos de cones sensíveis, são eles:

- Vermelho, sensível ao comprimento de onda Longo (L)
- Verde, sensível ao comprimento de onda Médio (M)
- Azul, sensível ao comprimento de onda Curto (S).

A teoria do processo oponente afirma que o sistema visual interpreta a cor num modo oposto:

- Vermelho versus verde
- Azul versus amarelo
- Preto versus branco.

No entanto, ambas as teorias estão correctas pois descrevem diferentes fases da fisiologia visual. A tabela 3 indica o intervalo de sensibilidade dos três tipos diferentes de cones.

Tabela 3 - Sensibilidade dos comprimentos de onda medida em nanometros (nm).

Tipos de cones	Faixa	Picos de comprimento de onda
L	500-700 nm	564 – 580 nm
M	450-630 nm	534 – 545 nm
S	400-500 nm	420 – 440 nm

Dificuldades de um portador de daltonismo

Cenário pessoal:

Um indivíduo portador de DVC (Deficiência de Visualização de Cores) têm algumas limitações no dia-a-dia, a escolha de uma simples peça de fruta no supermercado, a conjugação de cores para a roupa a vestir, interpretar um mapa da linha do autocarro, do comboio quando estas estão normalmente representadas por cores, distinguir o vermelho ou o verde de um semáforo, distinguir a bandeira vermelha ou verde de uma praia, podem ser tarefas complicadas de levar a cabo.

Cenário Profissional:

Também neste campo a execução de uma profissão cuja principal característica seja a de distinguir cores e conjugá-las, tal como designer de indústria, da moda, da decoração, do cinema etc, torna-se uma tarefa impossível de concretizar sem a ajuda de terceiros.

Cenário do Lazer:

Um outro segmento importante que limita em termos visuais as pessoas afectadas por esta limitação é a área do entretenimento como vídeos jogos em que a identificação das cores se pode tornar praticamente impossível de fazer. Visualizar um filme pode retirar qualidade quando a pessoa não se consegue aperceber de algumas cores presentes.

2.5 O mundo visto por um dicromata

São inúmeras as situações de confusão de cores vividas no dia-a-dia de um dicromata. Neste conjunto de imagens que se seguem (figura 6) pretendemos demonstrar essa perda de informação causada pela doença:

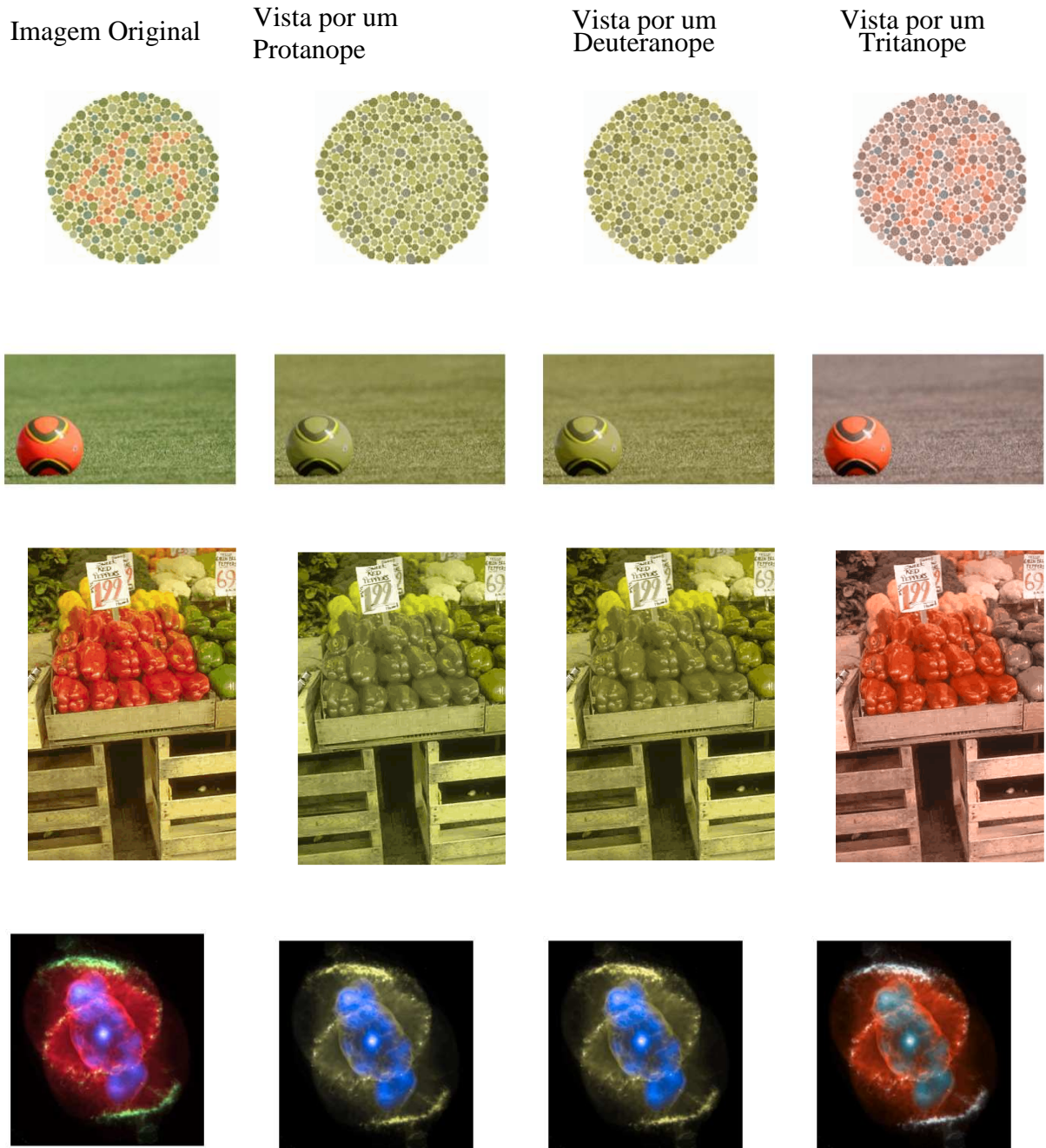


Figura 6 - Percepção dos 3 tipos diferentes de dicromancia [6].

Cada imagem original está representada com 128 cores, as imagens simuladas geradas pelo anexo em apêndice A.1 (simulação_dicromancia.m), a aproximação de gama de cores visíveis para os três tipos de dicromancia segundo Kuhn, no plano de cores $L^*a^*b^*$ representado na figura (7).

Este método foi implementado em Matlab e encontra-se em anexo no apêndice A.1.

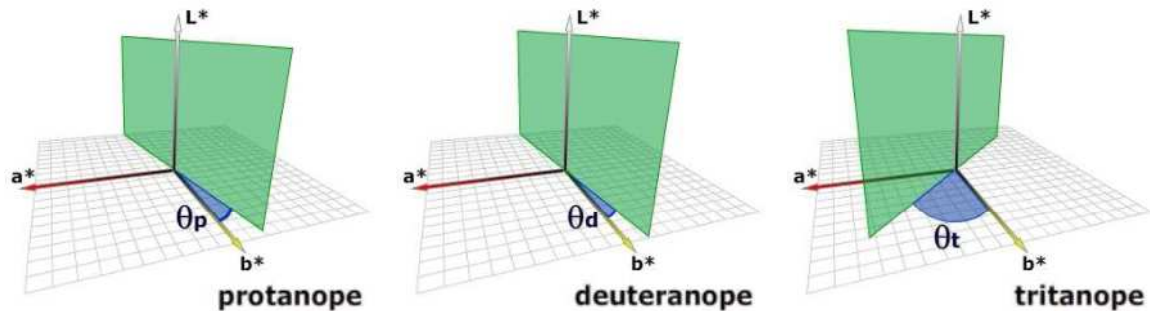


Figura 7 - Aproximação de gama de cores plano [Kuhn et al. 2008].

$\theta_p = -11.48^\circ$, para protanopes $\theta_d = -8.11^\circ$, para deuteranopes e $\theta_t = 46.37^\circ$, para tritanopes

2.6 Resultados/conclusões

O problema daltonismo consiste na redução da dimensionalidade das cores do espectro visível de três para duas dimensões, essa projecção implica a redução do espaço de cores que representam uma imagem e por isso a correspondente perda de qualidade na percepção das cores com conseqüente confusão na distinção de diferentes regiões/objectos da imagem.

Capítulo 3

3 Estado da Arte

Neste capítulo serão abordadas várias técnicas de simulação da percepção de portadores de algum tipo de dicromância, assim como alguns algoritmos propostos de correção daltônica de imagens.

3.1 Técnicas de Simulação de daltonismo

Para melhor entender como é visto o mundo pelos portadores da doença existem, na literatura várias abordagens e algoritmos que simulam a visão dos daltônicos. Algumas delas remontam já ao século XIX, por exemplo quando o alemão Goethe produziu uma tela (figura 8) em que demonstrou o problema da visualização de cores por pessoas com problemas na percepção da cor azul. Provou-se assim que a sensação de cor que chega ao cérebro humano é também formada pelos mecanismos físicos da visão humana e o modo como o cérebro a processa ¹ [Art 2010].

¹ <http://www.webexhibits.org/colorart/ch.html>

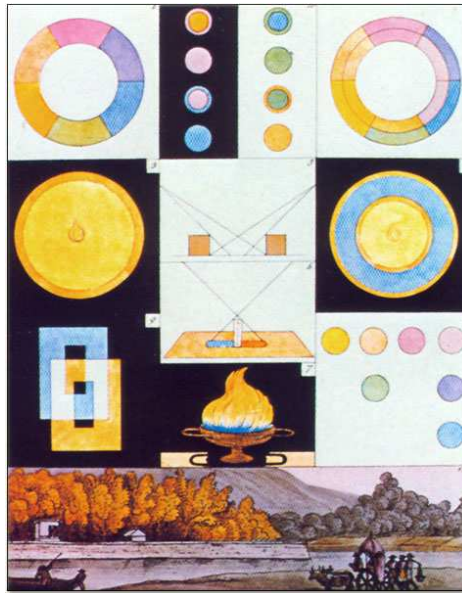


Figura 8 - Diagrama de Goethe na primeira placa de Zür Farbenlehre (teoria das cores) [7].

Mais recentemente Hans Brettel [Brettel et al. 1997] propõem a implementação de algoritmos computacionais que simulam a visão dos daltônicos, permitindo assim às pessoas com visão tricromática (saudáveis) visualizar as dificuldades dos portadores da doença.

De acordo com o que já foi referido anteriormente, indivíduos com visão normal têm três tipos de cones presentes chamados de vermelho, verde e azul, que diferem uns dos outros por possuírem fotopigmentos que são sensíveis aos comprimentos de onda longo (L), médio (M) e curto (S). A energia absorvida pelos cones L, M e S pode ser calculada pela integração numérica sobre o comprimento de onda λ [Brettel, at 1997].

$$[L, M, S] = \int E(\lambda) [\bar{l}(\lambda), \bar{m}(\lambda), \bar{s}(\lambda)] d\lambda \quad (3.1)$$

Onde $E(\lambda)$ é a distribuição da potência espectral da luz e $\bar{l}(\lambda), \bar{m}(\lambda), \bar{s}(\lambda)$ é a função espectral de absorção para os cones L, M e S

Brettel apresenta um método computacional expresso no espaço de cores tridimensional LMS com três tipos de projecções, uma por cada tipo de daltonismo.

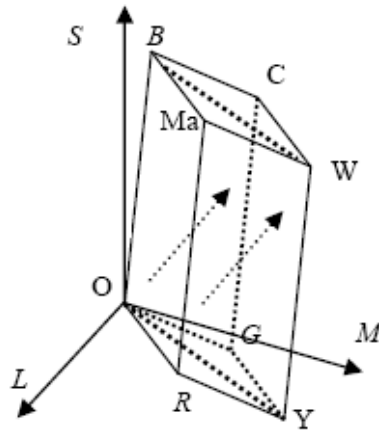


Figura 9 - Cores do cubo RGB projectadas no plano de protanopes ao longo do eixo L [8].

Os dados da imagem normalizados e representados pelo espaço de cores RGB são convertidos para o espaço de cores LMS para simular a visão humana representando em cada um dos eixos ortogonais os três tipos de cones existentes no olho que captam a luz, essa transformação é conseguida pela equação 3.2.

$$\begin{pmatrix} L \\ M \\ S \end{pmatrix} = U \times \begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} L_R & L_G & L_B \\ M_R & M_G & M_B \\ S_R & S_G & S_B \end{pmatrix} \times \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (3.2)$$

U representa a matriz de transformação do espaço de cores RGB para LMS.

As cores vistas pelos protanopes são distribuídas no plano com valores de R igual a de G, é por isso chamado plano de cor de protanopes. Como as cores ao longo da linha L serão projectadas para um ponto do plano estas tornam-se por isso confusas. Por exemplo vermelho e verdes são cores que se encontram ao longo do eixo L, quando estas são projectadas tornam-se quase indistinguíveis para protanopes. Para deuteranopes as cores do cubo RGB são projectadas para o plano com o valor de R igual ao de G ao longo do eixo M, que também é chamado de cores de plano deuteranope. Assim, os resultados das cores vistas por protanopes e deuteranopes são muito semelhantes.

Para o caso dos tritanopes as cores do cubo RGB são projectados sobre o plano com B igual ao G ao longo do eixo S, chamado de plano de cor tritanope.

O algoritmo é dividido em três partes:

- Cálculo dos valores tridimensionais do LMS a partir dos dados RGB de cada pixel que compõe a imagem.
- Aplicar geometricamente as projecções calculadas no ponto anterior, tendo em conta as limitações dos daltónicos.
- Calcular os valores RGB a partir das coordenadas representadas no eixo tridimensional LMS.

Este algoritmo foi implementado na linguagem C++ e encontra-se em anexo digital na pasta Brettel.

Vienot [Vienot et al. 1999] propõem uma correspondência de um mapa de cores que permite indicar se uma determinada cor será correctamente visível para protanopes e deuteranopes.

Ele constrói um mapa de cores para substituir na paleta standard de 256 cores, incluindo 216 (figura (10)) que são mais comuns em ambientes de desenvolvimento de sistemas MS Windows e Macintosh, mostrando quais são efectivamente as cores vistas pelos portadores da doença.

Neste trabalho Vienot adopta um método de sete passos para simular o mapa de cores visto pelos portadores de doença protanopes e deuteranopes. O método é também baseado no sistema de cores tridimensional LMS, especifica as cores em termos de relação com os diferentes comprimentos de onda Longa (L), Média (M) e Curta (S). Ou seja, partindo dos valores RGB originais de cada pixel, indicados por 8 bits, corresponderá um valor RGB no respectivo campo de visão do dicromata, Vienot cria ,assim um procedimento computacional para simular a visualização de cores. O resultado é uma paleta de cores com as respectivas cores vista pelos dicromatas, conforme mostra a figura (10). À esquerda de cada coluna encontramos a cor original seguida da cor visualizada por protanopes e deuteranopes

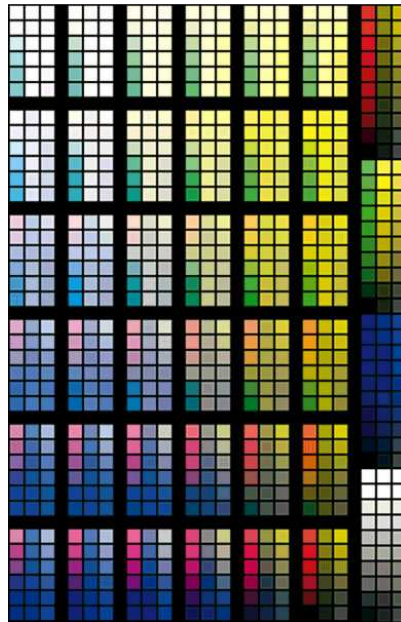


Figura 10 - Paleta de cores proposta por [Vienot et al. 1999].

3.2 Técnicas de correcção daltónica

Mais recentemente, uma série de técnicas de adaptação automática têm sido propostas para modificar o conteúdo de uma determinada imagem, proporcionando assim uma imagem melhorada às pessoas portadoras de daltonismo.

A rápida evolução tecnológica tem permitido desenvolver diferentes técnicas e abordagens ao problema da correcção daltónica de imagens para DVC. A literatura não é, de facto, muito rica relativamente a este tema, no entanto, as técnicas podem-se subdividir em duas classes diferentes, uma que requer a intervenção do utilizador e outra que funciona de forma automática.

3.2.1 Técnicas de re-coloração que requerem intervenção do utilizador.

Bob Dougherty e Alex R. Wade [Wade 2008] propõem um algoritmo a que chamam “daltonize” técnica que usa o sistema de cores $L^*a^*b^*$ (será abordado no capítulo (4.2.4) com mais detalhe) o sistema funciona para protanopes e deuteranopes, Para alterar a imagem original são necessários três parâmetros fornecidos pelo utilizador. Basicamente o utilizador

especifica quanto do vermelhos e verde da imagem original deve ser modificado puxado/separado, projectado no plano de luminância e no plano azul/amarelo plano a^* , respectivamente essa simulação pode ser visualizada na figura (11).

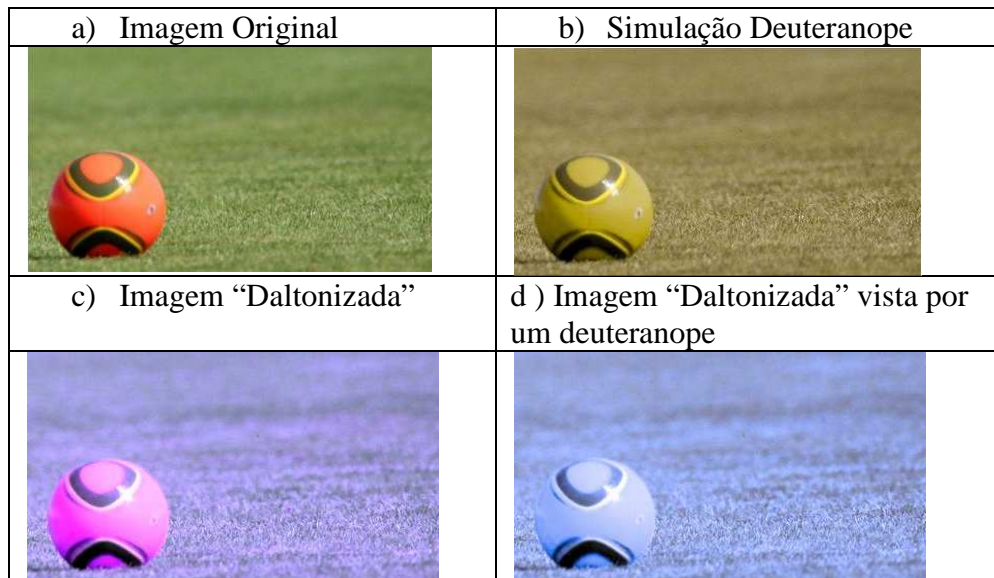


Figura 11 - Imagem corrigida usando o algoritmo proposto em [Wade 2008].

Gennaro Iaccarino [G. Iaccarino 2006] – usa o sistema de cores HSL onde H representa a Matiz (Hue), L a Luminosidade e S a Saturação. A Matiz descreve as cores individuais (a porção do espectro visível que contém as cores). O valor da saturação representa a intensidade de cada cor específica, enquanto a luminosidade representa a intensidade de brilho percebido. O Algoritmo é personalizável e requer a intervenção do utilizador especificando cada um destes três parâmetros anteriores, a qualidade dos resultados obtidos é altamente dependente dos dados fornecidos pelo utilizador.

Ou-Yang [Ou-Yang and Huang 2007] propõem um método para alterar a imagem de uma forma adaptativa de acordo com os diferentes graus da deficiência de visão de cores. Para resolver o problema da discriminação entre duas cores que se confundem aos olhos do portador da doença, o autor propõem a adaptação da saturação entre essas cores que se confundem. A saturação da cor é reduzida para dar a discriminação entre a tonalidade das cores alteradas relativamente à tonalidade das cores originais. Com isto, o autor diminui as componentes de cores cromáticas. A adaptação para DVC é feita pela equação (3.3).

$$H' = H + \Delta H, S' = S + \Delta S, I' = I \quad (3.3)$$

Onde H, S e I são a Tonalidade, Saturação e Intensidade da cor. H', S' e I' serão as cores adaptadas, a ΔH e ΔS são as variações da tonalidade e da saturação. Nesta adaptação é usada a relação de cores entre o magenta, cyan e amarelo representando-se assim a quantidade de cada uma das cores anteriores numa determinada cor.

Miguel Neiva no âmbito da sua tese de mestrado desenvolveu um código gráfico monocromático. Para cada cor primária (vermelho, amarelo e azul) criou um símbolo (respectivamente triângulo, barra diagonal, triângulo invertido) fácil de apreender, com custos de aplicação reduzidos, e passível de ser adaptado a vários cenários desde etiquetas de roupa a transportes públicos. A partir da conjugação dos três símbolos é possível identificar cores, tons, brilhos e misturas, a ideia consiste em aplicar estes símbolos directamente nos objectos coloridos [JN 2008]. O problema deste método é não oferecer uma representação de cor directa e completa sobre o objecto em observação.

3.2.2 Técnicas automáticas de re-coloração de imagens que não requerem intervenção dos utilizadores

Seungji Yang [Yang and Ro 2003] propõem uma abordagem baseada em operações pontuais que altera a cor de cada pixel independente das restantes cores. No entanto, esta técnica não efectua nenhuma análise na imagem para determinar o conjunto óptimo de cores para substituir. Têm um baixo custo computacional logo a possibilidade de execução em tempo real, no entanto o uso deste tipo de estratégia pode levar a que o DVC confunda outras cores na imagem resultante.

Karl Rache [Rasche et al. 2005] apresentam uma técnica de re-coloração que consiste numa optimização que tenta preservar a diferença entre todas as cores. No entanto essa transformação não capta as variações de cor e não garante que as cores sejam correctamente mapeadas para a gama disponível. Num trabalho posterior Rache [Rasche et al. 2005] abordou as limitações do procedimento anterior propondo um procedimento de optimização restrita multi-variada para um conjunto reduzido de cores quantizados. O conjunto das cores quantizadas é então usado para otimizar toda a imagem original.

Yu Ma [Ma et al. 2006] propõe um algoritmo capaz de proporcionar a percepção das cores aos daltônicos com um bio-dispositivo interno, usando, para isso, redes neurais e partindo do pressuposto que o córtex visual, apesar de não ser estimulado pelos cones em falta, ainda estará “vivo”. Segundo o autor, o modelo transforma sinais de entrada anormal correspondentes a dois tipos de cones em três sinais de saída correspondentes aos três tipos de comprimentos de onda L,M e S, proporcionando assim aos daltônicos a percepção de cores de um tricromata.

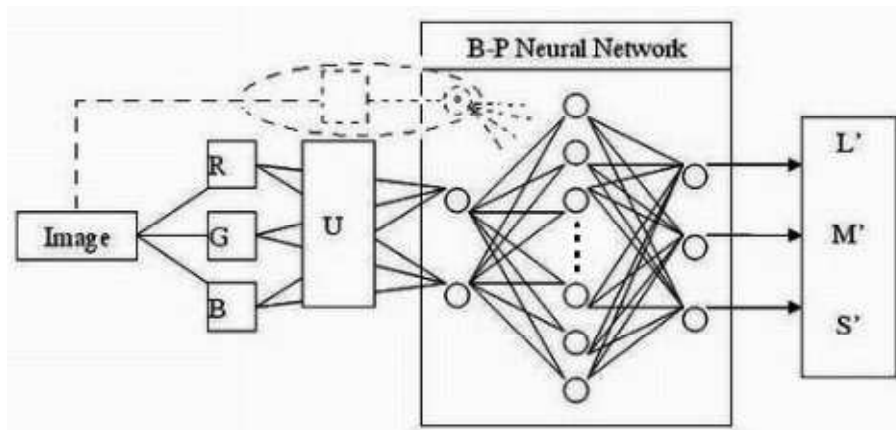


Figura 12 - Estrutura do modelo proposto por Yu Ma.

O mesmo autor do método anterior faz uma nova abordagem ao problema em [Ma et al. 2009] propondo um método de transformação de cor baseado em redes neurais SOMs (Self-Organizing Map). Utiliza para isso a redundância de informação das cores, alterando-as para que estas sejam perceptíveis pelos daltônicos. É usado um algoritmo para construir um mapa de cores não-linear, mantendo a relação de vizinhança entre as cores. Segundo o autor o método pode aumentar a discriminação de cores para daltônicos.

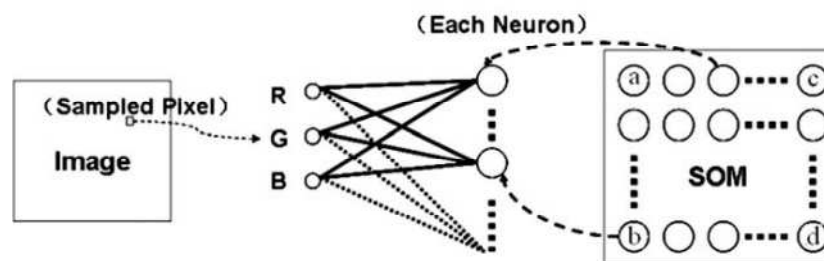


Figura 13 - Rede SOM utilizada neste método [Ma, Gu and Wang 2009].

Jefferson [Jefferson et al. 2007] formula o problema da adaptação das cores para portadores da deficiência como uma compensação baseada em otimização. O objectivo é modelar as

cores da imagem de modo a que estas possam ser vistas pelos DVC, a diferença percebida entre um determinado par de cores será da mesma magnitude que a diferença percebida por um observador normal. Este algoritmo usa os critérios definidos pela *World Web Consortium* (W3C) para melhorar a visualização de cores para pessoas com DVC. O algoritmo é composto por quatro passos:

1. Seleccionar um subconjunto (quantização) das cores principais da imagem
2. Calcular as diferenças de cor usando a diferença de brilho entre as cores principais.
3. Optimizar a adaptação das cores para DVC.
4. Interpolar as cores resultantes com as restantes cores da imagem usando a distância inversa ponderada.

Kuhn [Kuhn, Oliveira and Fernandes 2008] Apresentaram uma técnica eficiente e automática de re-coloração de imagem para daltónicos que preserva, na medida do possível, a naturalidade das cores da imagem original. Apresentam também uma extensão desta técnica que realça o contraste entre as cores com base num procedimento não linear. Segundo os autores a técnica proposta possui as seguintes características:

- 1- Pode ser executada em tempo real
- 2- Garante a consistência global das cores.
- 3- Preserva as diferenças da percepção entre as cores da imagem original.
- 4- Procede à alteração das cores quando estas não são percebidas pelo daltónico, preserva também assim a naturalidade das cores originais.

As técnicas de recoloração apresentadas anteriormente têm como grande objectivo alterar a imagem original de forma a proporcionar ao daltónico uma imagem de maior qualidade, que lhe possa tornar visíveis pormenores que antes passariam despercebidos.

As abordagens utilizadas para resolver o problema podem ser classificadas em dois sub-grupos: operações pontuais e métodos de optimização.

3.2.3 Operações pontuais

Os métodos que não fazem uma análise prévia na imagem original, fazem alterações à cor de cada pixel de forma independente, não tendo em conta nem a distância espacial nem a vizinhança das cores, são denominados de métodos pontuais. Este tipo de abordagem tem a vantagem de ter um custo computacional mais baixo e tem a possibilidade de ser executada em tempo real. O uso destas técnicas poderá fazer com que os portadores da doença confundam outras cores na imagem resultante. Assim a imagem produzida apesar de tornar visíveis algumas das regiões que antes não o eram, podem por outro lado criar confusão com outras já existentes, porque no seu mapeamento não foi tida em conta uma análise da imagem original.

3.2.4 Métodos de optimização

Este tipo de abordagem é por norma a estratégia, das técnicas automáticas descritas anteriormente, que tende a gerar melhores resultados sem a necessidade de intervenção do utilizador. Usam por norma, métodos de optimização, com o objectivo de gerar uma imagem que preserve o contraste das cores da imagem original. É comum nestas técnicas serem usados métodos que usam apenas um sub-conjunto de cores da imagem original diminuindo assim o custo computacional associado ao processamento dos algoritmos. Apesar desta simplificação nem todas as técnicas de re-coloração funcionam com tempos aceitáveis para aplicações interactivas. Das referidas anteriormente, destacamos o algoritmo proposto por Kuhn [Kuhn, Oliveira and Fernandes 2008] que consegue processar imagens com tempos aceitáveis para aplicações interactivas.

Capítulo 4

4 Processamento Digital de Imagens

As técnicas de processamento de imagens digitais são actualmente utilizadas para contribuir na resolução de variadas questões relacionadas com a análise e com a manipulação de imagens.

Uma imagem digital em duas dimensões é uma função $f(x,y)$, onde x e y são coordenadas espaciais e o valor de f em qualquer ponto (x,y) é proporcional ao nível de cinzento da imagem no ponto. O olho humano é menos sensível às variações de cor do que às mudanças de tons de cinzento, conseguindo discriminar centenas de cores mas apenas três dezenas de tons de cinzento.

Uma imagem digital pode ser considerada como uma matriz (equação (4.1)) cujos índices de linha e coluna identificam a posição de cada ponto e o valor do elemento de matriz (pixel) correspondente identifica o nível de cinza. A imagem em forma de matriz poderia ser representada pela figura (14).

$$f(x,y) = \begin{bmatrix} f(0,0) & \dots & f(0,N-1) \\ f(1,0) & \dots & f(1,N-1) \\ \vdots & \dots & \vdots \\ f(M-1,0) & \dots & f(M-1,N-1) \end{bmatrix} \quad (4.1)$$

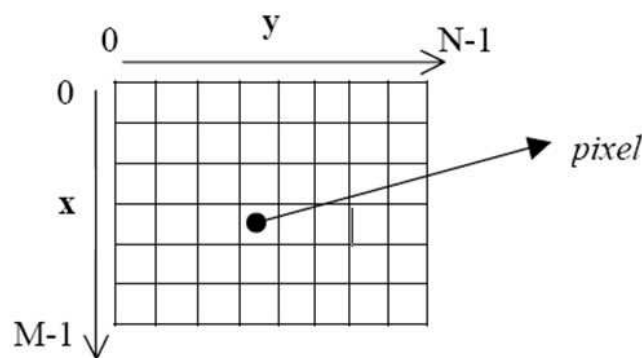


Figura 14 - Representação abstracta de uma imagem digital.

O sentido dos eixos numa imagem, bem como a variação dos índices e o sentido de leitura é feito no sentido *raster* (ou seja, linha a linha).

4.1 Geração de imagens digitais

As imagens digitais geradas por dispositivos electrónicos tiveram origem na tecnologia que globalmente se designa de digital. Este processo possibilita que sinais analógicos, provenientes da energia que chega ao sensor do digitalizador, sejam transformados em informação discretizada.

Para ser digitalizada, a informação analógica precisa de ser quantizada num certo número de estados digitais. Quanto maior o número de estados digitais melhor será a representação da informação analógica. Porém, será necessário um número maior de bits para representar a imagem e, conseqüentemente, será necessária mais memória para armazenar essa mesma imagem. O número de instruções necessário para processar a imagem será também maior.

4.2 Espaços de cores

Um espaço de cores é um meio pelo qual podemos especificar, criar e representar a cor. É um modelo abstracto matemático para descrever cada cor a partir de modelos matemáticos. Existem vários espaços de cores, cada um com as suas características, destacamos aqui alguns deles.

4.2.1 RGB

O *RGB* é o sistema de cores aditivo que permite obter as diversas cores derivadas através da composição aditiva das cores primárias:

- R – Red
- G – Green
- B – Blue

A cor preta, por exemplo, é formada pela ausência destas três componentes, enquanto a cor branca pura é gerada pela mistura dos valores máximos dos componentes (R,G,B). Assim, a representação de cada cor é modelada como uma combinação linear de intensidades das componentes R, G e B a cor final é calculada pela equação (4.2).

$$Cor\ final = r \times R + g \times G + b \times B \quad (4.2)$$

Onde, r, g e b são percentagens ou proporções das componentes primárias R, G e B.

4.2.2 CMY

É um espaço de cores que se baseia nas propriedades substractivas da cor. As cores representadas são C (*Cyan*), M (*Magenta*) e Y (*Yellow*). Este modelo tem por base o fenómeno de reflexão/absorção que se verifica quando a luz incide em superfícies. Estas podem absorver e reflectir a luz de forma desigual consoante o comprimento de onda. As cores ciano, magenta e amarelo são designadas por cores primárias complementares ou cores primárias substractivas, dada a forma de como resultam da subtracção do vermelho, verde e amarelo respectivamente, a cor final é obtida pela equação (4.3).

$$Cor\ final = 1 - R + 1 - G + 1 - B \quad (4.3)$$

As coordenadas do espaço *CMY* passam a ser as cores primárias substractivas e, tal como no espaço *RGB* a cor é modelada como uma combinação linear de intensidades *CMY*.

4.2.3 HSV

Neste espaço de cores, normalmente, as cores obtêm-se através de uma transformação não-linear do espaço *RGB* e é baseado na maneira de como as pessoas seleccionam as cores. É formado pelos componentes H (*Hue*), S (*Saturation*) e V (*Value*).

- *Hue* (Matiz) – Estabelece a natureza da cor, abrangendo todas as cores do espectro. Pode ter valores que variam de 0° a 360°.
- *Saturation* (Saturação)– Quanto menor for este valor, maior será o nível de cinzento, a sua representação pode ser em percentagem.
- *Value* (brilho) – define o brilho da cor, normalmente em percentagem.

4.2.4 CIE L*a*b

Em 1931, o *CIE* (*Commission Internationale d'Eclairage*) elaborou o sistema colorimétrico *xyY*, que respresenta as cores conforme a sua cromaticidade (eixos *x* e *y*) e a sua luminância (eixo *Y*). O diagrama cromático, com origem em transformações matemáticas, representa a periferia das cores puras, ou seja, as radiações monocromáticas que correspondem às cores do espectro, localizadas pelo seu comprimento de onda. Contudo, este modo de representação meramente matemático não têm em conta os factores fisiológicos da percepção da cor realizada pelo olho humano, o que resulta por exemplo num diagrama de cromaticidade com um espaço demasiado largo para os tons verdes.

É criado em 1976, pela mesma instituição, o espaço de cores **L*a*b***, este espaço proporciona uma representação tridimensional relativa à percepção do estímulo de cores. Funciona com três canais diferentes. O canal **L** (que varia de 0-preto a 100-branco), que representa a luminosidade de uma imagem, os canais **a** e **b** que representam a cor. Valores positivos em **a** indicam o vermelho e valores negativos indicam a cor verde. Valores positivos de **b** indicam azul e os negativos indicam o amarelo. Variando ambos entre -120 e 120 unidades. É um espaço com cores uniformemente distribuídas e de cromaticidade uniforme. As cores têm uma distribuição perceptualmente uniforme, a luminosidade esta separada da cromaticidade e detém a representação dos contrastes cromáticos naturais percebidos pelo Ser Humano, reproduzindo praticamente todas as cores do espectro visível.

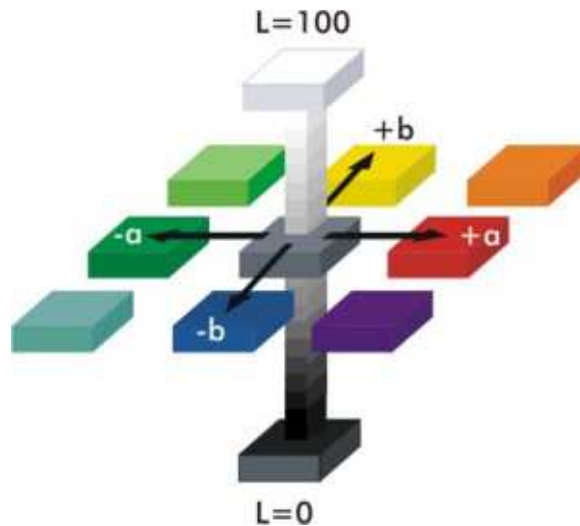


Figura 15 - Espaço de cores $L^*a^*b^*$ [9].

A conversão de RGB para $L^*a^*b^*$ passa pelo modelo base de desenvolvimento *CIE XYZ*.

As características descritas acima fazem o espaço de cores $L^*a^*b^*$ ideal para a sua utilização na modelação do sistema *mass-spring* (explicado nos capítulos posteriores) e para o desenvolvimento do algoritmo de re-coloração de imagens para dicromatas.

4.3 Amostragem espacial e quantização de níveis.

O grau de detalhe de uma imagem está directamente ligado às técnicas de amostragem e quantização. A amostragem é a resolução espacial e a a quantização de níveis está relacionada com o número de níveis de cinzento, figura (16).

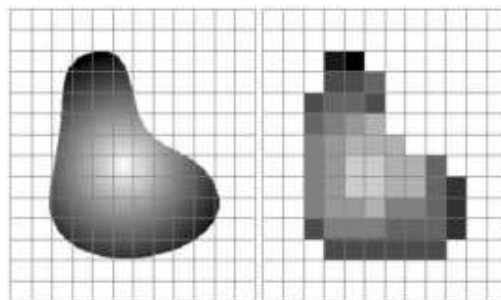


Figura 16 - Exemplo de uma imagem contínua antes e depois do processo de amostragem e quantização [10].

A resolução espacial de uma imagem figura (17) envolve a quantidade de linhas e colunas utilizadas. Ou seja, uma boa resolução de uma imagem depende do tamanho da matriz que representa essa mesma imagem.

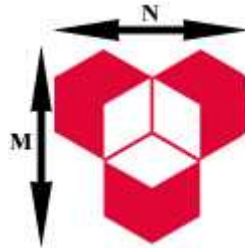


Figura 17 - Dimensão de uma imagem (Estig) [11]

A quantização de níveis envolve o número de níveis de cinzento (G) que $f(x,y)$ pode assumir. No geral, usam-se potências de base dois, nas quais o expoente, z , representa o número de bits que codificam o pixel, pela equação (4.4).

$$G = 2^z \quad (4.4)$$

O número de bits total (b) requerido para guardar uma imagem digital é dado pela equação (4.5).

$$b = N \times M \times z \quad (4.5)$$

4.4 Quantização de imagens – redução de cores

A quantização de uma imagem digital para redução de cores é um processo que implica perda de informação, mas é um método de redução do espaço de representação para minimizar a complexidade computacional do processo.

Efeitos da quantização de cor (512, 256, 128, 64, 32, 16, 8 e 4 níveis)



Figura 18 - Sequência de efeitos da quantização [10].

Conforme se pode verificar pelas imagens acima, as sucessivas reduções de cores resultam numa menor qualidade das mesmas.

O erro no processo de quantização é dado equação (4.6).

$$e_q = c - q(c) \quad (4.6)$$

Onde c representa a cor inicial, $q(c)$ cor quantizada, e $e(q)$ o erro de quantização. O melhor algoritmo de quantização é aquele que minimiza este erro.

4.4.1 Algoritmos de quantização

Existem na literatura inúmeros algoritmos de quantização entre eles destacamos:

4.4.1.1 Algoritmos de populalidade.

É um algoritmo adaptativo que define numa primeira fase os níveis de quantização determinando à posterior as células a quantizar, no geral para caracterizar as cores de uma imagem usa o histograma de frequências (figura (19)) de ocorrência de cores.

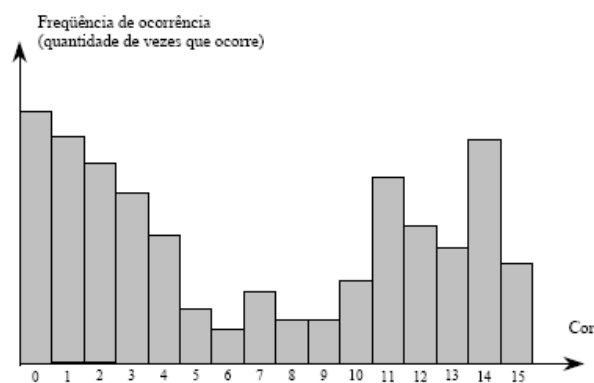


Figura 19 - Histograma de cores (exemplo).

4.4.1.2 Quantização uniforme.

Este método divide o espaço de cores em células uniformes (figura (20)) e toma o centro (valor médio) da célula como nível de quantização. É de fácil implementação e apropriado para imagens com distribuição de cores uniforme

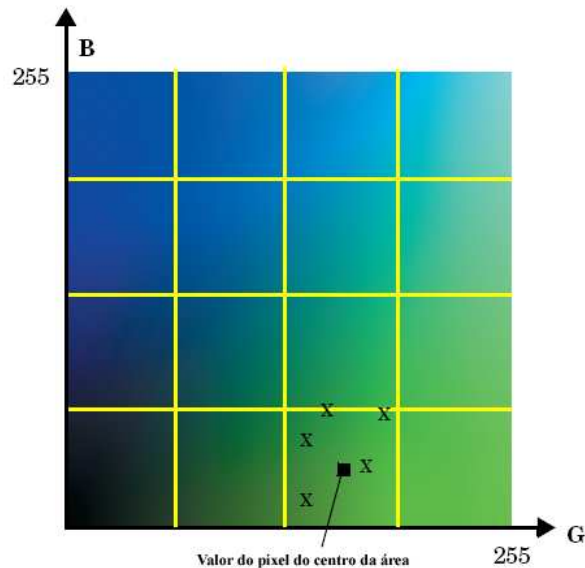


Figura 20 - Exemplo bidimensional da quantização uniforme [13].

Nota: no espaço RGB, as células seriam cubos

4.4.1.3 Quantização não uniforme.

Neste tipo de algoritmos destacamos o de quantização pela minimização da variância (figura (21)). O algoritmo divide o espaço de cores em células não uniformes, é ideal para reduzir cores em imagens que não estão distribuídas de uma forma uniforme, adaptando o tamanho das células à estatística da imagem original. Produz melhores resultados que a quantização uniforme mas tem tempos de execução ligeiramente mais lentos.

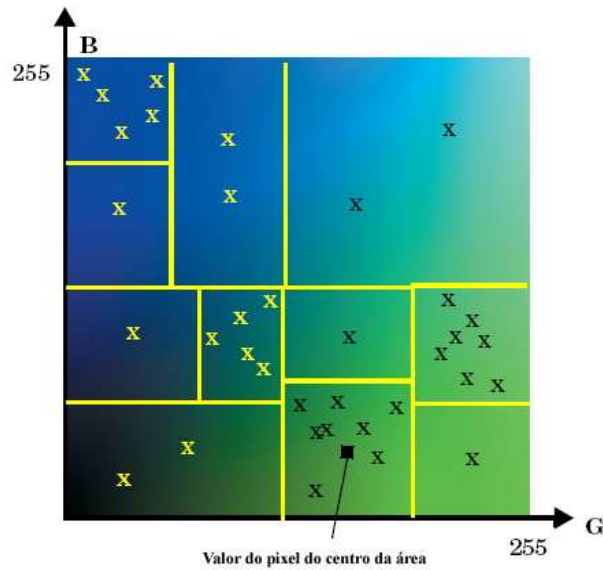


Figura 21 - Exemplo bidimensional da quantização não uniforme [13].

4.4.1.4 Hierárquicos divisivos

No geral, fazem partições do espaço de cores RGB mas adaptadas à distribuição de cores da imagem; um exemplo é o corte mediano, baseia-se na medida estatística denominada mediana, estabelece em primeiro lugar as células de quantização para depois calcular o nível de quantização associado a cada célula com base no seu valor mediano, procura assim definir células de quantização de forma a equalizar o histograma figura (22).

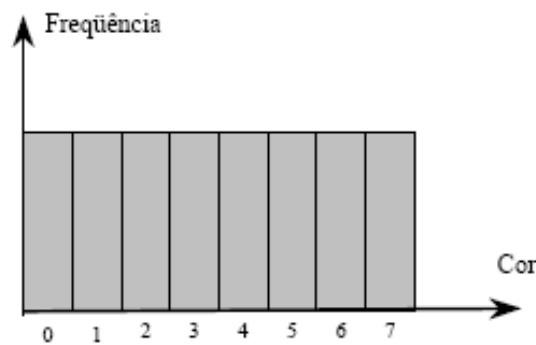


Figura 22 - Histograma equalizado por corte mediano.

4.4.1.5 Hierárquicos aglomerativos

Estes métodos recorrem a decisões hierárquicas de agrupamento de cores para obter uma redução da quantidade de cores na imagem, são exemplo disso as estratégias Octree figura (23).

É um algoritmo que estrutura dados em árvore em que cada nó interno têm exactamente oito filhos, são frequentemente usados para particionar espaços tridimensionais como é o caso do espaço de cores RGB sub-dividido recursivamente por oito nós até nove níveis de profundidade.

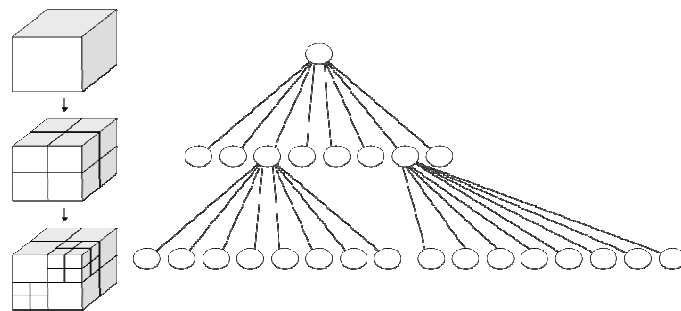


Figura 23 - Esquema de partição em octree [14].

À esquerda da imagem é visível a subdivisão recursiva de um cubo em oito partes e à direita a *octree* correspondente²

² http://en.wikipedia.org/wiki/Octree#Application_to_color_quantization

Capítulo 5

5 Daltonização por *mass-spring*

Os processos como a daltonização carecem do aumento da complexidade dos métodos para a sua resolução funcionalmente adequada. Este aumento de complexidade produz, normalmente, uma elevação exponencial na carga computacional necessária à implementação do processo. Nesta secção será abordado com maior detalhe o algoritmo proposto por Kuhn [Kuhn, Oliveira and Fernandes 2008] por ser a abordagem que preserva a naturalidade da imagem original e por ter tempos de execução consideravelmente mais baixos que abordagens do mesmo tipo.

Este capítulo detalha os aspectos fundamentais do sistema *mass-spring*, como uma ferramenta de optimização com baixo custo computacional que pode ser implementado e adaptado ao problema. A compreensão e a dinâmica do sistema *mass-spring* são fundamentais para a modelação e optimização da re-coloração de imagens, tanto na abordagem feita por Kuhn como na proposta por nós.

5.1 Conceitos básicos

A abordagem proposta utiliza uma optimização do sistema de base física “*mass-spring*”, este sistema consiste num conjunto de partículas (cada uma com uma determinada massa) ligadas por um conjunto de molas “*spring*” (estas, com um determinado tamanho e em estado relaxado). Quando aplicadas forças externas (exemplo gravidade, atrito, inércia) as molas exercem forças opostas para restaurarem as distâncias entre as diferentes partículas. As molas

tendem assim a chegar ao seu estado relaxado garantindo que as distâncias iniciais entre as partículas sejam as mesmas.

O algoritmo proposto é então constituído por três grandes etapas:

- (i) Quantização da imagem
- (ii) Optimização do sistema *mass-spring* sobre as cores quantizadas no ponto (i).
- (iii) Reconstrução das cores finais

Na primeira etapa é usado um processo de quantização para extrair um conjunto Q de cores da imagem de entrada.

Na segunda etapa, inicia-se o sistema *mass-spring* com o conjunto Q de cores quantizadas, onde a cada cor deste conjunto é associada uma partícula do sistema. É criada uma mola entre cada par de partículas associadas, tendo em conta que o tamanho relaxado de cada mola é definido pela diferença na percepção entre cada par de cores do ponto de vista de um indivíduo com visão normal. Durante o processo de optimização do sistema *mass-spring*, a posição actual de cada partícula é definida pela posição da sua cor em associação com a cor percebida pelo indivíduo com DVC. Devido às forças restauradoras das molas, o sistema tende a estabilizar quando as molas voltam ao seu estado de relaxamento, isso acontece quando as distâncias entre as partículas, do ponto de vista dos DVC, são iguais as diferenças entre as cores do conjunto Q das cores quantizadas (correspondentes à percepção de um indivíduo normal).

Na terceira e última etapa, reconstrói-se a imagem final utilizando o conjunto de cores resultante do processo de optimização e que deverá corresponder à melhor *daltonização* possível.

5.2 Sistema “*mass-spring*”

O sistema *mass-spring* é um modelo utilizado na física para o estudo de oscilações, em rede, de partículas.

Um sistema *mass-spring* é constituído por um conjunto de partículas ligados entre si por molas que se deformam com a presença de forças externas, quando o sistema é comprimido ou esticado, as molas aplicam forças de reacção internas, com a finalidade de restabelecer o estado de repouso.

O sistema tende a estabelecer-se quando as forças externas são compensadas por forças de oposição internas. Estas características fazem do sistema *mass-spring* uma técnica de optimização flexível que permite optimizar um conjunto de parâmetros. Tais como, a posição de cada partícula relativamente ao antes e ao depois da aplicação das forças externas, satisfazendo a restrição como a soma das forças internas e externas ser igual a zero.

Adaptação do sistema *mass-spring* às diferentes fases do algoritmo proposto:

- (i) Posição das Partículas
- (ii) Massa das Partículas
- (iii) Forças aplicadas a cada partícula
- (iv) Comprimentos das molas no estado de repouso.
- (v) Comprimento actual das molas.

A massa de cada uma das partículas poder ser determinada por forma a terem valores diferentes fazendo assim com que as partículas com massas maiores se mexam menos aquando do restabelecimento das forças internas.

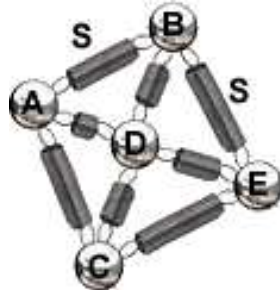


Figura 24 - Sistema *mass-spring* composto por cinco partículas (A,B,C,D e E) e oito molas (S) com diferentes comprimentos e estados de repouso [15].

5.3 Dinâmica do sistema “*mass-spring*”

Considerando um conjunto de partículas, todas elas ligadas por molas, então o sistema deverá ser capaz de simular a atribuição de massas a todas as partículas, assim como os respectivos estados a cada uma das molas. Todo o sistema deverá por isso obedecer à segunda lei de Newton equação (5.1):

$$F = m * a \quad (5.1)$$

Onde m é a massa da partícula P , a é a aceleração causada pela força F_i que traduz a composição de forças internas e externas ao sistema. Assim sendo, a força aplicada à partícula P pode ser obtida pela lei de *Hooke* (equação (5.2)) somando as tensões das molas de todas as partículas a que se encontra ligada a partícula P .

$$F_i = \sum_{j \in N} K_{ij} \left(1 - \frac{l_{ij}}{l_{ij}^0}\right) (p_j - p_i) \quad (5.2)$$

K_{ij} representa a constante elástica da mola que traduz a rigidez da mola. Quanto maior for a constante elástica da mola, maior será a sua reacção.

Onde N é o número de vizinhos ligados a P_i , l_{ij} e l'_{ij} são o comprimento de repouso e o comprimento actual da mola entre P_j e P_i .

5.4 Algoritmo de recoloração de imagem

Nesta secção é apresentada a técnica proposta por Kuhn por ser a base da abordagem adoptada para o nosso algoritmo.

Segundo os autores, a técnica foi projectada para atingir os seguintes objectivos:

- Desempenho em tempo real;
- Preservar a naturalidade das cores;
- Manter a consistência da cor global;
- Manter a luminância;
- Manter a cromaticidade;
- Manter o contraste de cores.

O algoritmo utiliza o sistema *mass-spring* para otimizar as cores da imagem de entrada conforme apresentado na secção anterior. A gama de cor de cada classe de dicromancia pode ser representado por um único plano que passa pelo eixo da Luminosidade Vienot [Vienot et al 1999] e cujo ângulo, no espaço de cores, foi determinado com dados experimentais e com recurso ao método dos mínimos quadrados. Os ângulos entre os planos de simulação e o plano de L^*b^* são $\theta_p = -11.48^\circ$, $\theta_d = -8.11^\circ$, $\theta_t = 46.37^\circ$, para protanopes, deuteranopes e tritanopes respectivamente.

Estes ângulos são depois usados para alinhar os planos de daltonismo com os correspondentes planos L^*b^* simplificando o processo de optimização ao eixo b^* (os valores de luminância são mantidos sem alterações).

5.5 Modelação do sistema *mass-spring*

Usando o espaço de cores $L^*a^*b^*$, a cada cor quantizada $q_i \in Q$ (figura (25)) é associada uma partícula P_i com massa m_i . As partículas são ligadas entre si por molas $S_{(i,j)}$ figura (26).

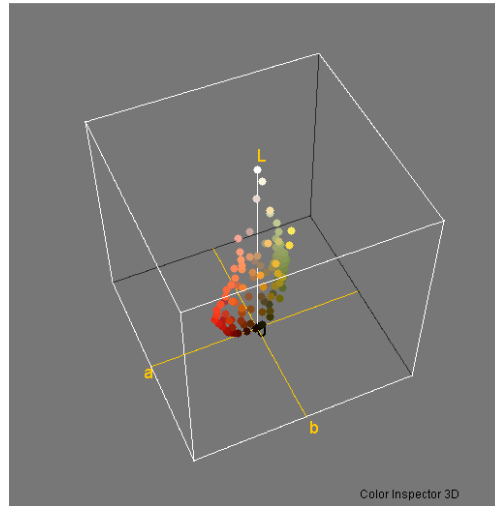


Figura 25 - Representação de uma imagem de 128 cores quantizadas no espaço de cores $L^*a^*b^*$ [12].

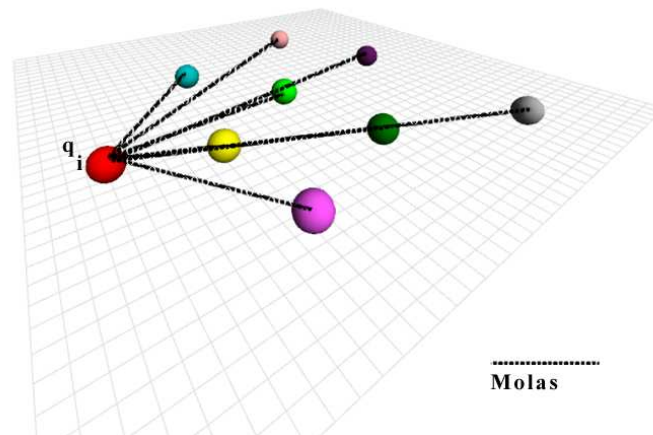


Figura 26 - Exemplo de partículas $q(i)$, no espaço $L^*a^*b^*$, percebidas por tricromatas e interligadas por molas.

Cada par de partículas P_i e P_j é ligado por uma mola S_{ij} com um coeficiente de elasticidade $K_{ij} = 1$ (equação (5.2)) e com um comprimento de repouso definido por $l_{ij} = \|q_i - q_j\|$.

Este valor é definido pela distância euclidiana das partículas representadas no espaço de cores $L^*a^*b^*$, notar que q_i e q_j são as cores quantizadas percebidas por tricromatas (visão normal).

A posição p_i , no eixo b , no sistema $L^*a^*b^*$, da partícula P_i é determinada e iniciada com as coordenadas correspondentes às cores q percebidas pelo dicromata após a projecção, seguida de rotação, desses pontos de cor original q . A projecção faz-se de os pontos q para o plano de dicromatas. A rotação faz-se do plano de dicromatas para o plano que alinha com o plano L^*b^* . Ao fazer-se este alinhamento simplifica-se o processo posterior de optimização, pois o deslocamento dos pontos far-se-á apenas numa coordenada. A equações (5.3) descrevem o processo:

$$\vec{p}_i = M_\theta D \vec{q}_i \quad (5.3)$$

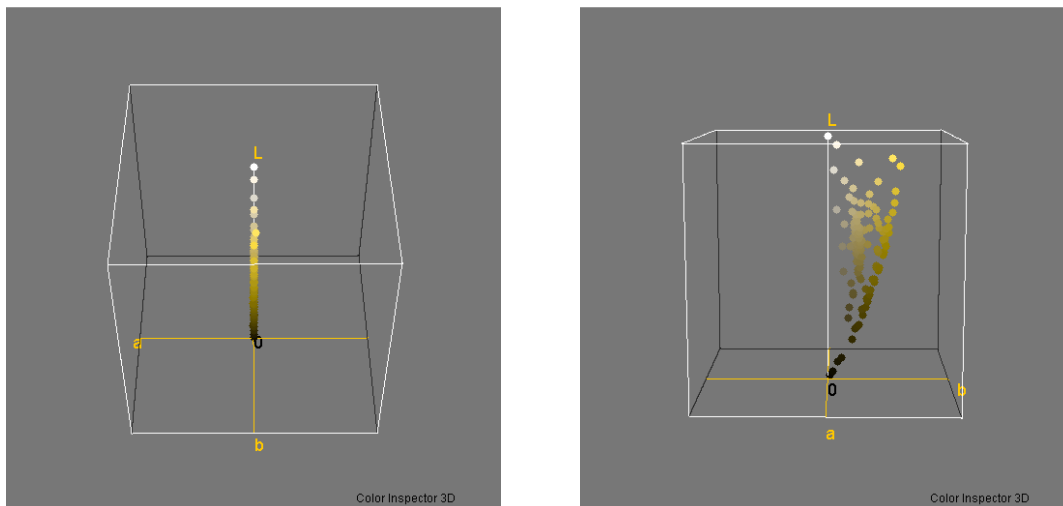


Figura 27 - Partículas P_i alinhadas em b^* no plano $L^*a^*b^*$ [12].

Onde M_d é a matriz de projecção $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

M_θ é a matriz de rotação $\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}$ que alinha as partículas no plano L^*b^*

Em cada iteração da otimização são actualizadas as posições de p_i e p_j das partículas P_i e P_j e calculados os comprimentos actuais das molas S'_{ij} . Este comprimento é dado pela distância euclidiana entre as partículas p_i e p_j tal como descrito na equação (5.4).

$$l'_{ij} = \|p_i - p_j\| \quad (5.4)$$

Dada a força restauradora das molas, o sistema irá tentar convergir para uma configuração em que os comprimentos de mola para dicromatas igualam os comprimento de mola das cores originais quantizadas, $l'_{ij} = l_{ij}$, para todas as molas S_{ij} . Após a estabilização ou a iteração máxima for atingida, a distância perceptual entre todos os pares de cores será aproximadamente igual aos inicialmente calculados aquando da quantização das cores iniciais q_i e q_j do conjunto Q .

Com a finalidade de reforçar a preservação das cores naturais, os autores definem a massa m_i a partir da distância perceptual recíproca (no espaço de cores $L^* a^* b^*$) entre \vec{q}_i e $D \vec{q}_j$ calculado pela equação (5.5).

$$m_i = \frac{1}{\|\vec{q}_i - D \vec{q}_j\|} \quad (5.5)$$

Onde D é calculado pela equação (5.6).

$$D = M_\theta^{-1} M_d M_\theta \quad (5.6)$$

Esta estratégia garante que qualquer cor percebida da mesma forma por tricromatas e dicromatas terá massas maiores fazendo por isso com que as suas partículas correspondentes se mexam menos na modelação *mass-spring*. Pelo contrário, distâncias maiores, entre a cor quantizada e a cor correspondente, projectada no plano de dicromatas, conduzirá a uma partícula p de massa pequena e consequentemente permitirá o deslocamento da posição partícula para a obtenção do contraste original de cores.

O cálculo das novas posições das partículas é dado pela equação (5.7)

$$p_i(t + \Delta t) = \frac{F_i(t)}{m_i} (\Delta t)^2 + 2p_i(t) - p_i(t - \Delta t) \quad (5.7)$$

O último conjunto T de cores é obtido aplicando a cada cor p_i a rotação inversa utilizada na equação (5.3). O resultado corresponde às cores já optimizadas e visíveis pelo dicromata pela equação (5.8).

$$\vec{t}_i = M_\theta^{-1} \vec{p}_i \quad (5.8)$$

A optimização pelo sistema *mass-spring* pode ficar limitada pela retenção em mínimos locais no espaço de representação do problema. Isto pode, em algumas situações, gerar imagens com poucas melhorias significativas. Na prática, isto acontece quando a imagem de entrada tem cores que são percebidas da mesma maneira ou muito próximas entre tricromatas e dicromatas, tendo por isso grandes massas. (equação (5.5)) Situações deste tipo impedem o deslocamento de partículas ao longo do eixo b^* . Os autores propõem um método não linear para resolver o problema. O método consiste na troca de sinal das coordenadas b^* , modificando em rebatimento posicional (no espaço de cores) as cores cuja coordenada a^* é positiva e cuja percepção de distância entre a cor em si e como ela é percebida por dicromatas é maior que um limite L. Esse valor é definido, pelos autores do trabalho, para o 15 como sendo o que melhores resultados apresenta. Desta forma, cores que efectivamente detêm erro e que estão próximas de cores (no plano dicromático) situadas no semi-volume esquerdo (mas no mesmo semi-volume, inferior ou superior), são diferenciadas pela passagem de uma delas para o semi-volume oposto, inferior ou superior pela equação (5.9).

$$p_i^{b^*} = \begin{cases} -p_i^{b^*} & \text{if } (qr_i^{a^*} > 0) \text{ E } (\|\vec{q}_i - D\vec{q}_i\| > L) \\ p_i^{b^*} & \text{caso contrário} \end{cases} \quad (5.9)$$

Onde $\overline{qr}_i = M_\theta D \vec{q}_i$ e representa a rotação e projecção dos pontos Q no plano dicromático de ângulo θ

5.6 Resultados/conclusões

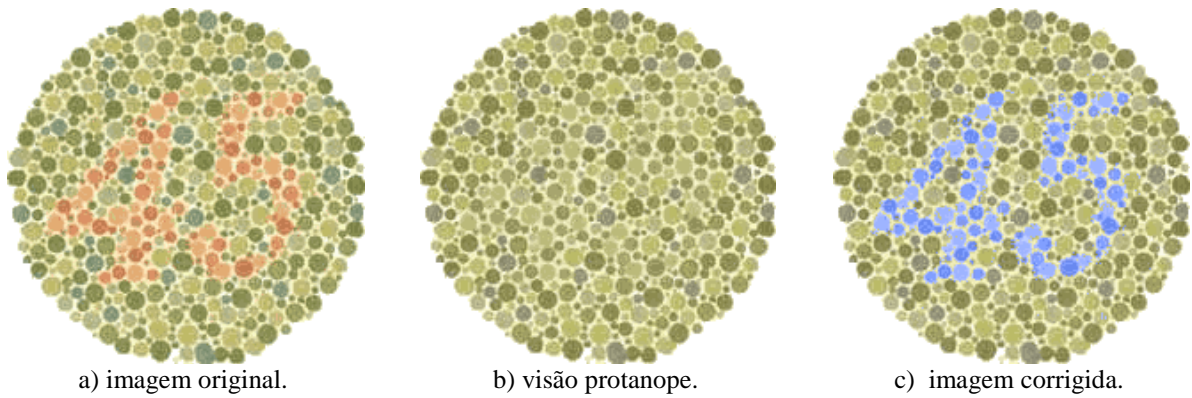


Figura 28 - Correção baseada no método de Kuhn segundo os testes de Ishihara [6].

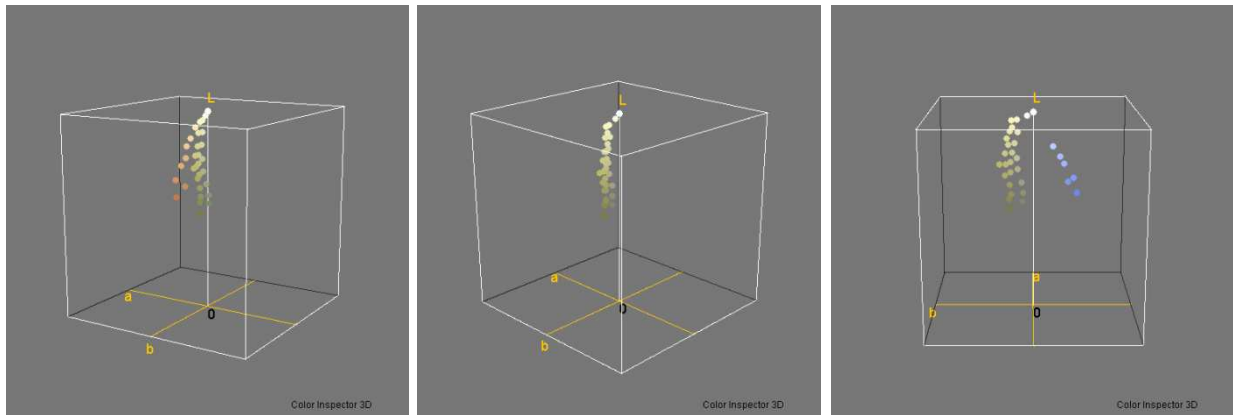


Figura 29 - Imagem da esquerda: Representação no espaço de cores Lab; Imagem do centro: Correção Kuhn; Imagem da direita: Teste Ishihara [12].

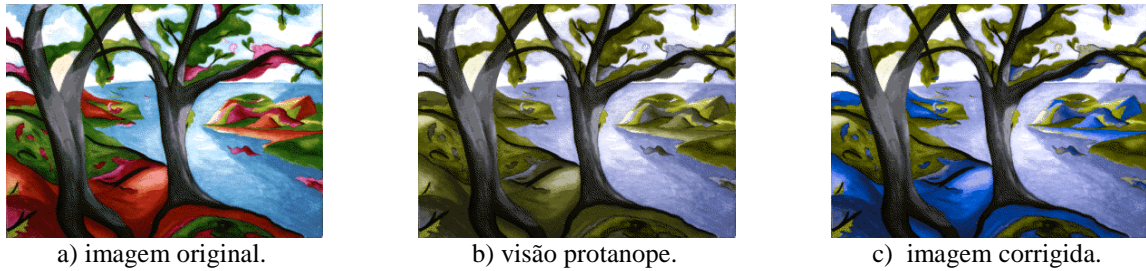


Figura 30 - Correção *mass-spring* Kuhn (árvores [10]).

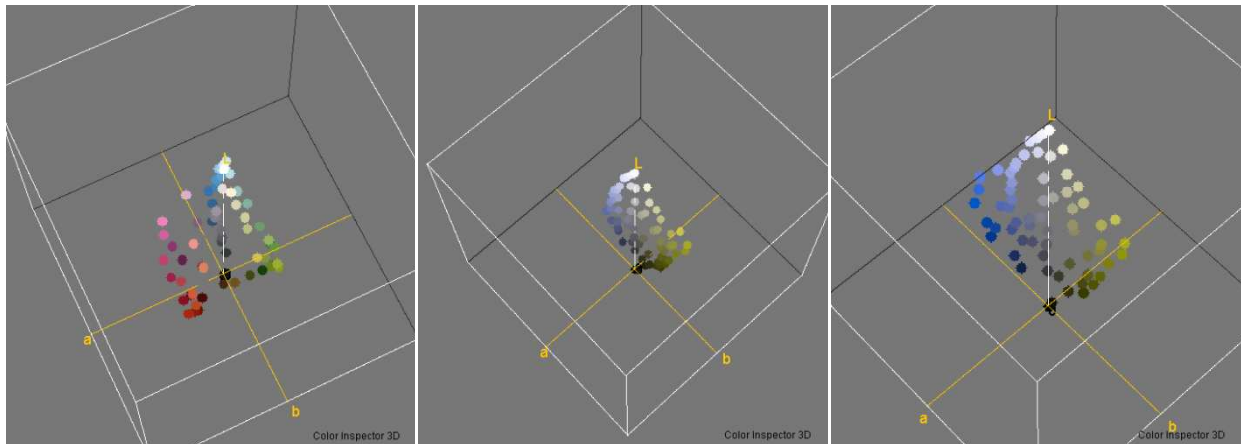


Figura 31 - Representação espaço de cores $L^*a^*b^*$, Correção Kuhn exemplo jardim.

Na figura (29) o algoritmo proposto por Kuhn cria uma imagem mais informativa para o protanope, pois ele consegue agora visualizar mais detalhes.

No entanto, na figura (30) o mapeamento das cores vermelhas para as cores azuis cria na imagem corrigida confusão com cores existentes na original. Ou seja, a solução apresentada neste método não garante que a imagem resultante não venha a destruir outras cores e regiões já existentes na imagem original mas que eram possíveis de visualizar pelo protanope.

Este método foi implementado em Matlab e encontra-se em anexo no apêndice A.2.

Capítulo 6

6 Algoritmo proposto

O capítulo anterior apresenta uma estratégia de re-coloração de imagens para dicromatas, mantendo, tanto quanto possível, a naturalidade das cores da imagem original. No entanto, não é feita uma análise espacial da distribuição das cores.

A heurística de mínimos locais, apresentada no algoritmo anterior, poderá em alguns casos não gerar imagens com os resultados esperados, exemplo da figura (30).

Com o nosso algoritmo pretendemos colmatar essa falha apresentando dois tipos de abordagens que são descritas nas secções seguintes.

6.1 Conceitos básicos

Nesta secção iremos apresentar uma abordagem que fará a análise da distribuição de vizinhança de cor entre pixéis. Essa análise residirá em métodos estatísticos de análise de textura, que será complementada com um conjunto de heurísticas actuadoras. Pretende-se garantir que o mapeamento final de cores não colide com as cores já existentes na imagem original mas que são percebidas pelos dicromatas.

Impede-se assim a confusão cromática entre regiões originalmente diferentes.

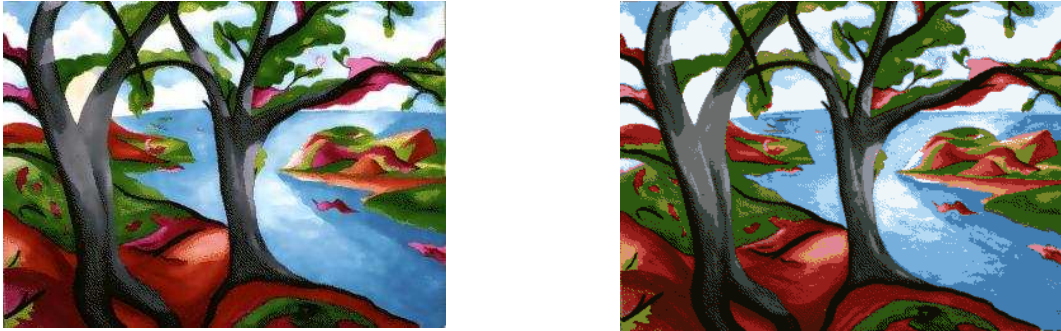
O algoritmo proposto apresenta uma nova abordagem (em duas componentes) de re-coloração de imagem para dicromatas, está organizado em 6 etapas fundamentais:

- i) Quantização da imagem original em N cores;
- ii) Análise da textura da imagem resultante de i);
- iii) Cálculo dos índices de vizinhança das cores;
- iv) Utilização do sistema *mass-spring*;
- v) Utilização dinâmica da constante de elasticidade da mola e aumento do comprimento inicial das molas das cores vizinhas em iii) componente 1;
- vi) Afastamento das cores vizinhas componente 2.

6.2 Quantização da imagem original em N cores

Nesta primeira fase e à semelhança do sugerido por Kuhn, pode ser usado um qualquer método de quantização (median cut, oct tree, uniforme, não uniforme) para reduzir o número de cores da imagem original diminuindo a quantidade de operações e cálculos sobre cada imagem e aumentando a liberdade de deslocamento das cores no espaço respectivo durante o processo de optimização

O algoritmo de quantização (figura (32)) usado na implementação Matlab foi o não uniforme (que estabelece a divisão do espaço de cores com base na minimização da variância) conforme referido no capítulo (4.3.3.1). A sua escolha foi a preferida, por ser de rápida implementação, por representar as cores mais representativas de imagens não lineares, por executar em tempos aceitáveis e por se obter bons resultados globais face ao problema em questão. Na implementação, do algoritmo, em *java* foi usada a *octree* por apresentar resultados a nível de tempo e de agrupamento de cores, mais adequados às condicionantes da implementação em dispositivos móveis.



a) Imagem original 54547 cores.

b) Imagem quantizada 16 cores.

Figura 32 - Exemplo de redução de cores [10].

Indexação de cores quantizadas figura (33), tabela (5).

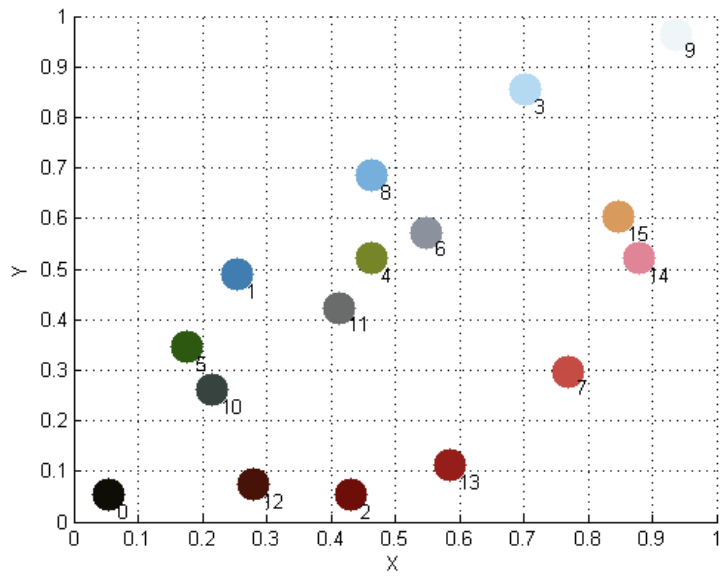


Figura 33 - Redução e indexação para 16 níveis de cor da figura 32.

Tabela 4 - Imagem na forma de uma matriz de índices de cores $N \times M$ onde N representa a largura e M a altura da imagem.

	1	2	3	4	5	6	7	→
1	6	9	6	6	6	6	6	..
2	3	3	3	8	3	8	3	..
3	3	3	8	3	3	8	8	..
4	8	8	8	8	8	3	8	..
5	3	8	3	8	8	8	8	..
6	8	8	8	3	8	3	8	..
7	8	3	8	8	3	8	3	..
↓	:	:	:	:	:	:	:	:

6.3 Determinação de cores vizinhas

Para determinar quais as cores que são vizinhas e que precisam efectivamente de ser distinguidas usamos um método estatístico de análise de textura que considera a relação espacial de cada pixel comparando os índices de cor quantizadas de cada um dos pixéis em análise. O método é também conhecido como o método da matriz de dependência espacial. Este método utiliza uma matriz de co-ocorrência que caracteriza numericamente a textura de uma imagem, calculando quantas vezes um determinado par de pixéis ocorre num relacionamento espacial determinado.

6.3.1 Matriz de co-ocorrência

A matriz de co-ocorrência quantifica quantas vezes um pixel com a intensidade de valor de i ocorre em uma relação específica espacial com um pixel com o valor j . Esta relação de vizinhança entre pixéis é calculada em várias direcções e com uma distância de D pixéis conforme ilustra a figura (34).

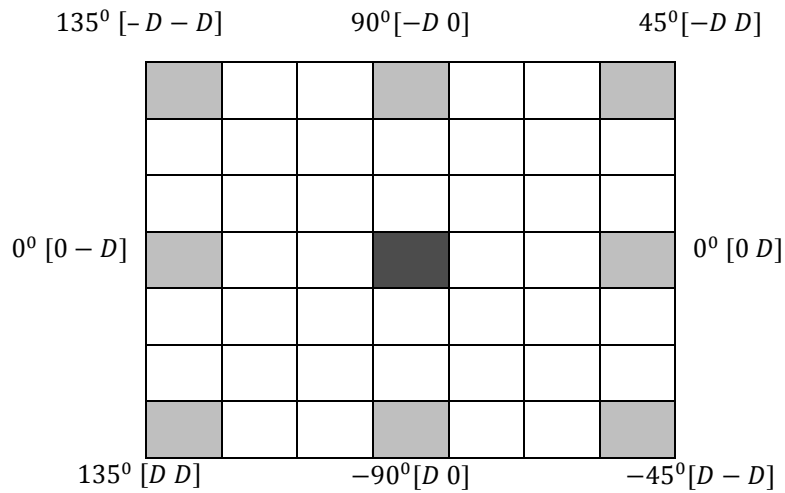
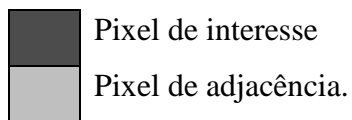


Figura 34 - Máscara de amostragem dos pixels que compõem uma imagem com a identificação do pixel de interesse (ao centro) e os pixels de adjacência.



Onde D representa a distância do pixel de interesse ao pixel de adjacência, no caso da figura (34) está estabelecido com o valor três ($D=3$).

O tamanho da Matriz de co-ocorrência (MCO) é definido pelo número de cores quantizadas no passo 6.1 i). Esta matriz passa a ser designada por MCO.

A figura (35) corresponde aos valores adquiridos pela máscara de amostragem, quando posicionada centralmente num pixel da imagem a analisar.

	1	2	3	4	5	6	7	→
1	6	9	6	6	6	6	6	..
2	3	3	3	8	3	8	3	..
3	3	3	8	3	3	8	8	..
4	8	8	8	8	8	3	8	..
5	3	8	3	8	8	8	8	..
6	8	8	8	3	8	3	8	..
7	8	3	8	8	3	8	3	..
↓	:	:	:	:	:	:	:	:

Figura 35 - Índices (aplicação da máscara de amostragem) de cores da imagem quantizada.

Nesta figura (36) são calculadas as co-ocorrências do pixel de posição de máscara (4,4) e com índice de cor 8. O processo é concluído quando todos os pixels da imagem quantizada foram percorridos e contabilizados na matriz de co-ocorrências figura (36).

I	1	2	3	4	5	6	7	8	→
1									..
2									..
3									..
4									..
5									..
6									..
7									..
8			1			3		4	..
↓	:	:	:	:	:	:	:	:	∴

Figura 36 - Matriz de co-ocorrências.

A contagem de co-ocorrência do pixel de posição, de índices (figura 35), (4,4) indexado de cor 8 na célula (linha, coluna) (8,3) fornece o valor 1 porque na máscara de amostragem o par (8,3) apenas se repete uma vez, já o par (8,6) repete-se 3 vezes.

Só as cores diferentes é que podem ser consideradas no processo de análise de cores vizinhas. É assim necessária a anulação da contabilidade das vizinhanças de uma cor com ela própria. Esse processo é conseguido pela equação (6.1) e representado na figura (37).

$$MCO_{(i,i)} = 0 \quad (6.1)$$

	1	2	3	4	5	6	7	8	→
1	0								..
2		0							..
3			0						..
4				0					..
5					0				..
6						0			..
7							0		..
8			1			3		0	..
↓	:	:	:	:	:	:	:	:	0

Figura 37 - Anulação da vizinhança de índices de cores iguais da MCO.

É necessário conhecer as posições da *MCO* que correspondem aos pares de cores mais vizinhas.

O cálculo das posições (*p*) das cores que mais se “tocam” é conseguido pela equação (6.2).

$$TPos = DSort(MCO, Index2D(MCO)) \quad (6.2)$$

Onde *TPos* representa a Tabela de Posições ordenadas, posições essas que correspondem aos índices, linha coluna, originais de *MCO* que referenciam as co-ocorrências, de cor, em ordem decrescente. O operador *Index2D* devolve uma matriz com duas colunas. O número de linhas devolvido, em matriz, por este operador, corresponde ao número de células da matriz *MCO*. Cada linha identifica as coordenadas, em *MCO*, de um determinado par de cores que ocorreu *n* vezes. O operador *DSort*, ordena decrescentemente, com base no conteúdo das células *MCO* a matriz de coordenadas dada por *Index2D*. Como essas coordenadas correspondem aos pares de cores vizinhas, os pares de cores, de cores que mais vezes estão próximas, aparecerão nas

primeiras posições da matriz $TPos$. Esta informação é importante para as fases posteriores do algoritmo, pois indica quais os pares de cores que devem ser diferenciados. Se os pares de cores correspondem a cores diferentes e que estão espacialmente próximas, devem ser tratados com mais peso diferenciador do que pares de cores, de cores que estão espacialmente distantes.

6.3.2 Distância euclidiana.

Quando se trabalha com processamento de imagens é muito frequente a necessidade de medir distâncias entre pontos situados em referenciais euclidianos, no nosso algoritmo esta necessidade prende-se com o facto de obtermos uma relação espacial da proximidade das cores.

O cálculo final de determinação do conjunto de cores vizinhas a processar deriva do quanto essas cores são semelhantes. Essa informação é dada pela distância tridimensional no plano $L*a*b*$ de cores. Cores vizinhas, determinadas pela MCO, podem ser eliminadas do processamento posterior, se forem parecidas. Assim, apenas cores vizinhas, com um certo grau de dissemelhança, serão consideradas no posterior processo de separação de cores.

Se C_1 e C_2 formam o par de cores que mais se toca na MCO, então a distância tridimensional desse par de cores é calculada pela equação (6.3) conforme ilustra a figura (38).

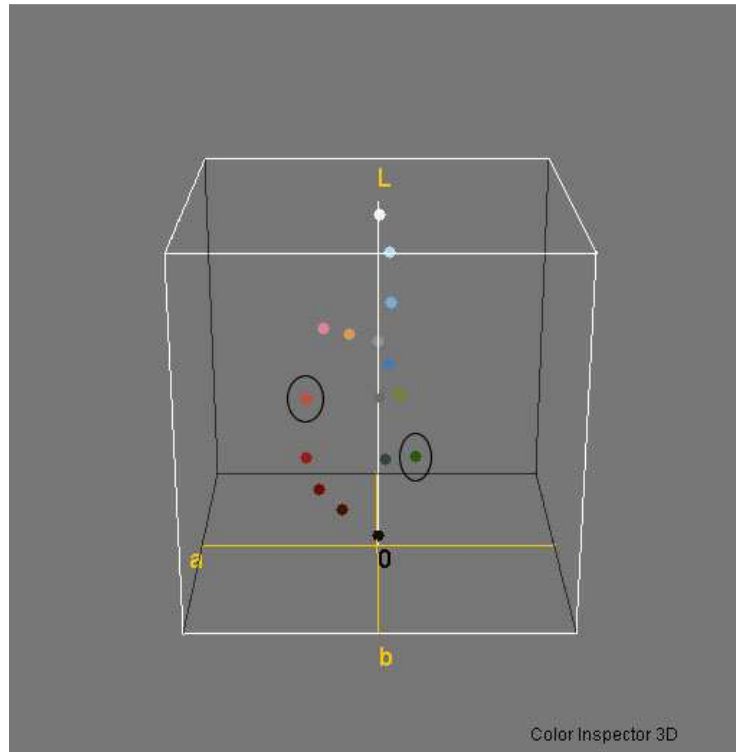


Figura 38 - Representação de 2 pontos no espaço tridimensional.

$$C_1 = (L_1, a_1, b_1) \text{ e } C_b = (L_2, a_2, b_2)$$

$$Dist_{(1,2)} = \sqrt{(L_2 - L_1)^2 + (a_2 - a_1)^2 + (b_2 - b_1)^2} \quad (6.3)$$

Onde Dist representa a Distância do par de cores vizinha 1 e 2.

O par de cores é então dado como vizinho se a distância (Dist) for superior a 40, equação (6.4), conforme ilustra a figura (39)

$$\text{If } Dist_{(p1,p2)} > Y \text{ then separate} \quad (6.4)$$

O limite de Y força a que só os pares de cores da MCO que assumem um valor de distância superior a 40 possam ser especialmente processados por separação. Pelas experiências efectuadas, este valor garante que são classificadas como cores a separar as cores que originalmente estão espacialmente próximas mas que se diferenciam no seu tom. A figura (39) representa os quatro pares de cores com maiores pesos na MCO e com $Y > 40$.

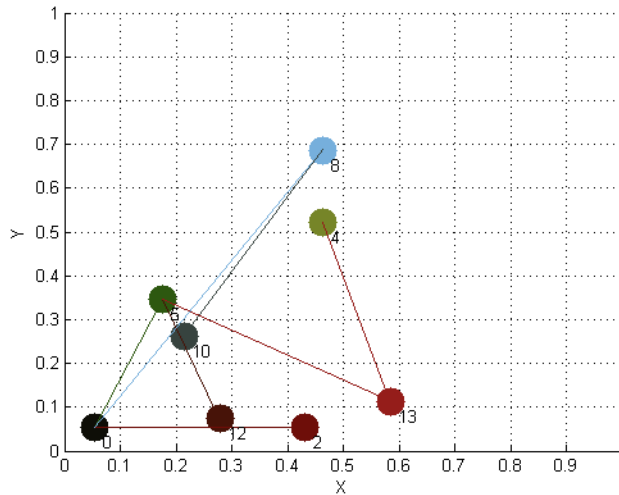


Figura 39 - Cores vizinhas da figura 32.b).

6.4 Adaptação do sistema *mass-spring* ao algoritmo proposto

A adaptação do sistema *mass-spring* ao algoritmo proposto é feito conforme descrito pelo capítulo 5.5, com a exceção do cálculo das forças e das cores finais.

A soma das forças causada pelas tensões das molas que ligam cada uma das partículas (cores) P_i é calculada pelo conjunto de equações (6.5).

Se $l_{ij} > l'_{ij}$ e $l_{ij} \neq 0$ (6.5)

$$F_i = \sum_{j \in N} -K_{(i,j)} \left(1.0 - \frac{l'_{(i,j)}}{l_{(i,j)}}\right) (p_j - p_i)$$

Se $l_{ij} = 0$

$$F_i = \sum_{j \in N} K_{(i,j)} (l'_{(i,j)}) (p_j - p_i)$$

Se $l'_{(i,j)} \neq 0$

$$F_i = \sum_{j \in N} K_{(i,j)} \left(1.0 - \frac{l_{(i,j)}}{l'_{(i,j)}}\right) (p_j - p_i)$$

Se $l'_{(i,j)} = 0$

$$F_i = \sum_{j \in N} K_{(i,j)} (l_{(i,j)}) (p_j - p_i)$$

Estas equações garantem que todas as situações possíveis da modelação *mass-spring* são representadas pelas suas forças, permitindo também a convergência e a estabilidade de todo o sistema realimentado.

As novas posições das partículas P_i é calculado pela equação (6.6).

$$p_i(t + \Delta t) = \frac{F_i(t)}{m_i} (\Delta t)^2 + \frac{1}{2} p_i(t) - \frac{1}{2} p_i(t - \Delta t) \quad (6.6)$$

As novas posições são calculadas, iterativamente, até que o sistema converge ou até que o número de iterações seja alcançado.

6.5 Dinâmica das molas (componente 1)

Nesta abordagem a estratégia é aumentar o comprimento original das molas e introduzir um novo coeficiente de elasticidade para cada par de partículas calculadas como vizinhas. Obrigando assim à separação de cores que confundem o dicromata.

6.5.1 Comprimento das molas nos pares de cores vizinhas.

Seja C_1 vizinha de C_2 e correspondentemente representadas por (L_1, a_1, b_1) e (L_2, a_2, b_2) , o novo comprimento de mola das cores vizinhas diferentes calculado pela equação (6.7)

$$l_{new(i,j)} = l(i,j)^2 \quad (6.7)$$

Onde $l_{new(i,j)}$ é o novo comprimento de mola do par de cores vizinhas, $l(i,j)$ comprimento de mola de cada par de partículas $\vec{q}_i \in Q$.

6.5.2 Dinâmica do coeficiente de elasticidade das molas

Seja C_1 vizinha de C_2 e correspondentemente representadas por (L_1, a_1, b_1) e (L_2, a_2, b_2) , o novo $kl_{(i,j)}$ coeficiente de elasticidade da mola é calculado pela equação (6.8).

$$kl_{(i,j)} = k_{new} * k_{(i,j)} \quad (6.8)$$

Onde $k_{(i,j)} = 1$, [Kuhn, Oliveira and Fernandes 2008] e $k_{new} = 1.3$. O valor foi calibrado por tentativas, para garantir a estabilidade e a convergência de todo o sistema. O kl é o novo valor da constante de elasticidade a integrar no sistema *mass-spring* para cada par de cores vizinhas diferentes.

6.5.3 Cálculo das forças

Calculadas as novas variáveis no ponto 6.5.1 e 6.5.2 é agora necessário introduzi-las no sistema *mass-spring*; isso é conseguido no cálculo das forças pelo sistema de equações (6.9).

Se $l_{new_{ij}} > l'_{ij}$ e $l_{new_{ij}} \neq 0$ (6.9)

$$F_i = \sum_{j \in N} -Kl_{(i,j)} \left(1.0 - \frac{l'_{(i,j)}}{l_{new_{(i,j)}}}\right)(p_j - p_i)$$

Se $l_{new_{ij}} = 0$

$$F_i = \sum_{j \in N} Kl_{(i,j)} (l'_{(i,j)})(p_j - p_i)$$

Se $l'_{(i,j)} \neq 0$

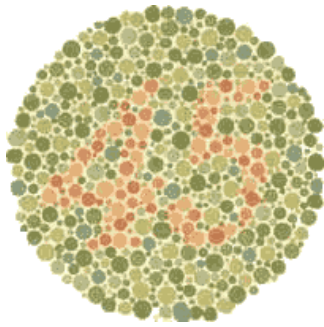
$$F_i = \sum_{j \in N} Kl_{(i,j)} \left(1.0 - \frac{l_{new_{(i,j)}}}{l'_{(i,j)}}\right)(p_j - p_i)$$

Se $l'_{(i,j)} = 0$

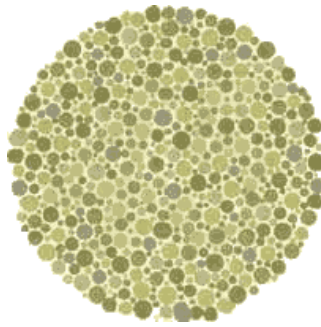
$$F_i = \sum_{j \in N} K_{(i,j)} (l_{new_{(i,j)}})(p_j - p_i)$$

6.5.4 Resultados/conclusões

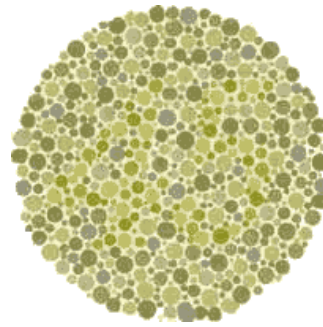
Imagem Original



Vista por um protanope



Corrigida pelo algoritmo de componente 1



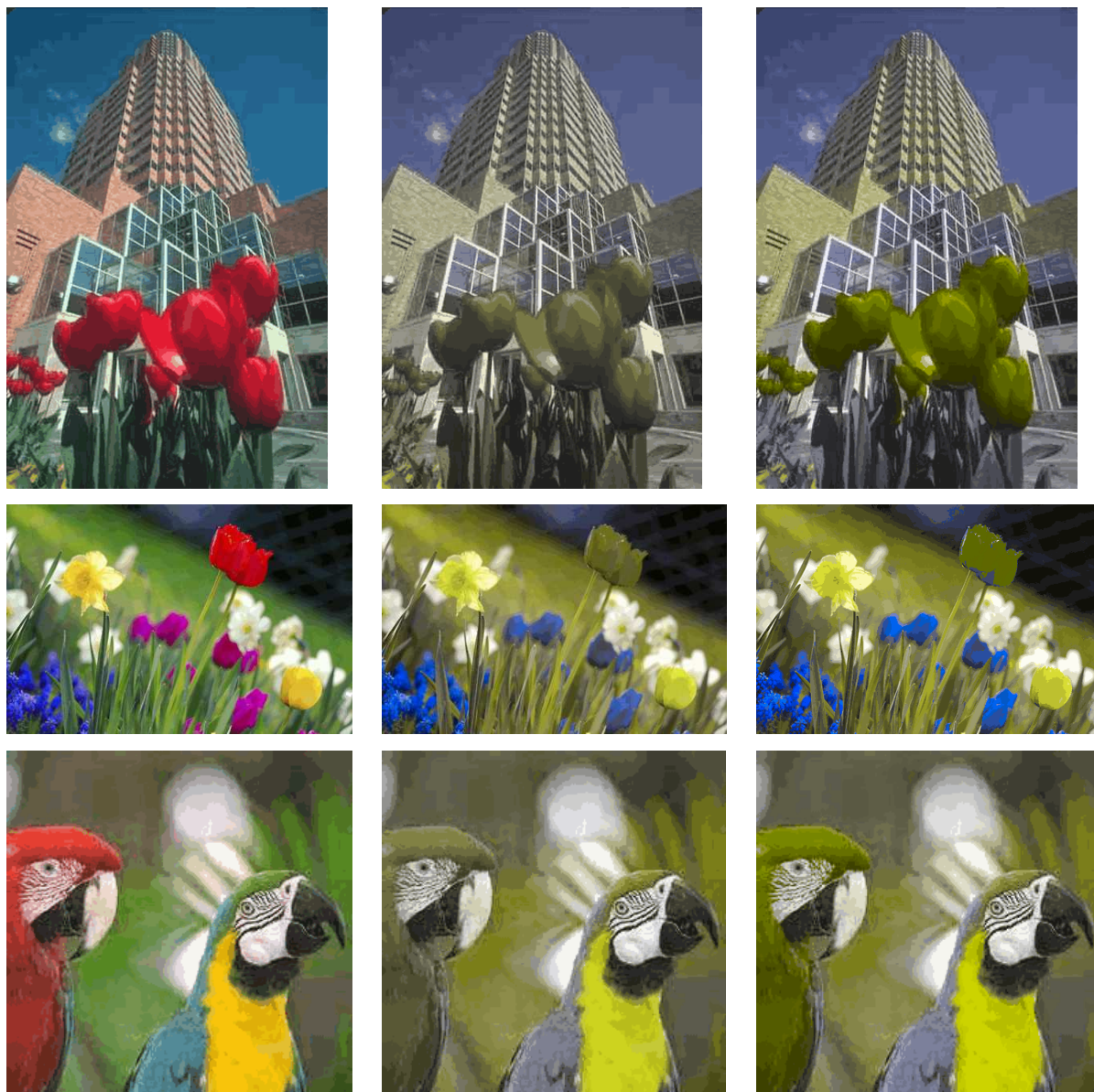


Figura 40 - Conjunto de imagens geradas pelo algoritmo proposto para a componente 1

As imagens quantizadas foram geradas para 64 níveis (Apêndice (A.3.1), (A.3.2), (A.3.3)), a estratégia implementada no componente 1 cria contraste nas cores que são vizinhas, garante por isso a percepção das regiões, embora em alguns casos essa percepção não seja muito evidente, Figura 40.

6.6 Afastar cores vizinhas (componente 2)

Nos processos anteriormente apresentados, o otimizador baseado em *mass-spring*, tenta conseguir obter a melhor distribuição possível de cores, no plano de daltonismo, para se igualar a relação de cores originais. Evidentemente, esse mecanismo está condicionado pela menor magnitude espacial da representação daltónica face à representação tricromática. A tentativa de obedecer ao contraste de cores original é reflectida entre todos os pares de cores presentes na imagem a processar. Este facto inibe a possibilidade de tratar o contraste de certas cores que seriam mais importantes. A estratégia apresentada a seguir pretende corrigir o processo global, de forma a dar um contraste final às cores mais importantes.

Seja C_1 vizinha e diferente de C_2 , correspondentemente representadas por (L_1, a_1, b_1) e (L_2, a_2, b_2) , no plano daltónico de ângulo θ . Propomos nesta abordagem o afastamento dessas cores de uma forma percentual, após actuação do sistema *mass-spring*. Basicamente é traçado o vector distância entre as cores do par de cores e a cada uma das cores é deslocada uma percentagem desse mesmo vector, forçamos assim, o contraste entre essas cores, tornando-as mais perceptíveis para o dicromata.

6.6.1 Vector distância

Seja C_1 vizinha de C_2 representadas por (L_1, a_1, b_1) e (L_2, a_2, b_2) o vector distância $(\overline{u}_{(1,2)})$ é calculado pela equação (6.10).

$$\overline{u}_{(1,2)} = C_2 - C_1$$

Daqui deriva

$$\overline{u}_{(1,2)} = (L_2 - L_1, a_2 - a_1, b_2 - b_1) \quad (6.10)$$

$$\overline{u}_{(1,2)} = (u_{(L)}, u_{(a)}, u_{(b)}) \quad (6.11)$$

6.6.2 Módulo da distância

O módulo da distância fornece o valor absoluto associado à magnitude do vector distância e é calculada usando a relação quadrática expressa na equação (6.12).

$$v = \sqrt{(L_2 - L_1)^2 + (a_2 - a_1)^2 + (b_2 - b_1)^2} \quad (6.12)$$

6.6.3 Versor distância

O versor é um vector unitário que contém a informação espacial das propriedades de direcção e sentido, pode ser calculado pela equação (6.13) da seguinte forma:

$$\vec{w} = \left(\frac{u(L)}{v}, \frac{u(a)}{v}, \frac{u(b)}{v} \right) \quad (6.13)$$

6.6.4 Distância a incrementar

A magnitude a incrementar ao vector distância para afastar as cores é calculado pela equação (6.14) conforme ilustra a figura (41).

$$\overrightarrow{Dist_{new}} = Dist + v * \frac{\%afastamento}{2} \quad (6.14)$$

Onde $\%afastamento = \frac{factor_{afastamento}}{quantidade\ de\ vizinhos}$, com $factor_{afastamento} = 1.8$, calibrado para afastar as cores dos pares de cores da imagem resultante com uma percentagem proporcional à quantidade de cores vizinhas. Este último factor vai garantir que o afastamento só terá lugar se o número de cores vizinho for diminuto, diminuindo a probabilidade de o próprio método promover novas colisões de cores.

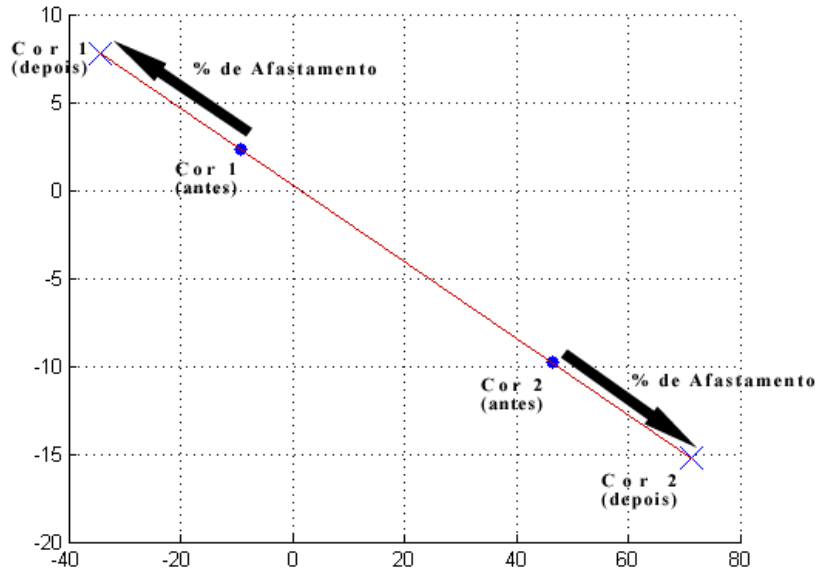
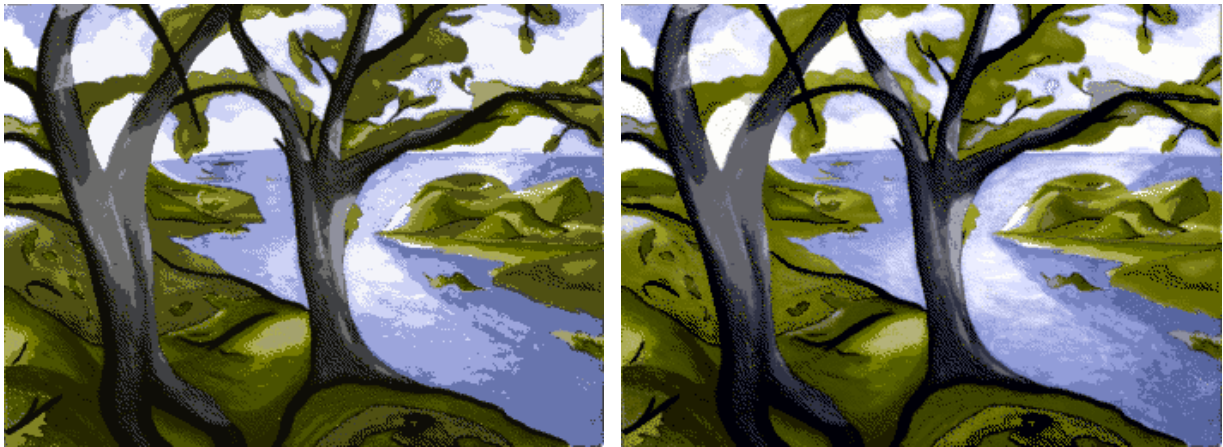


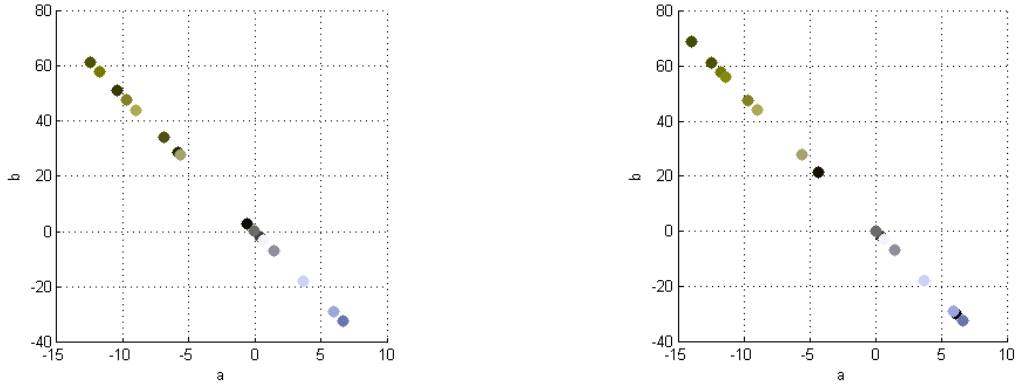
Figura 41 - Exemplo das partículas no processo de afastamento de duas cores vizinhas.

O resultado do afastamento das cores é mostrado no conjunto de imagens figura (42).



a1) imagem quantizada com 16 cores projectadas em sem afastamento de pares de cores vizinhas.

b2) imagem quantizada com 16 cores projectadas em com afastamento de pares de cores vizinhas.

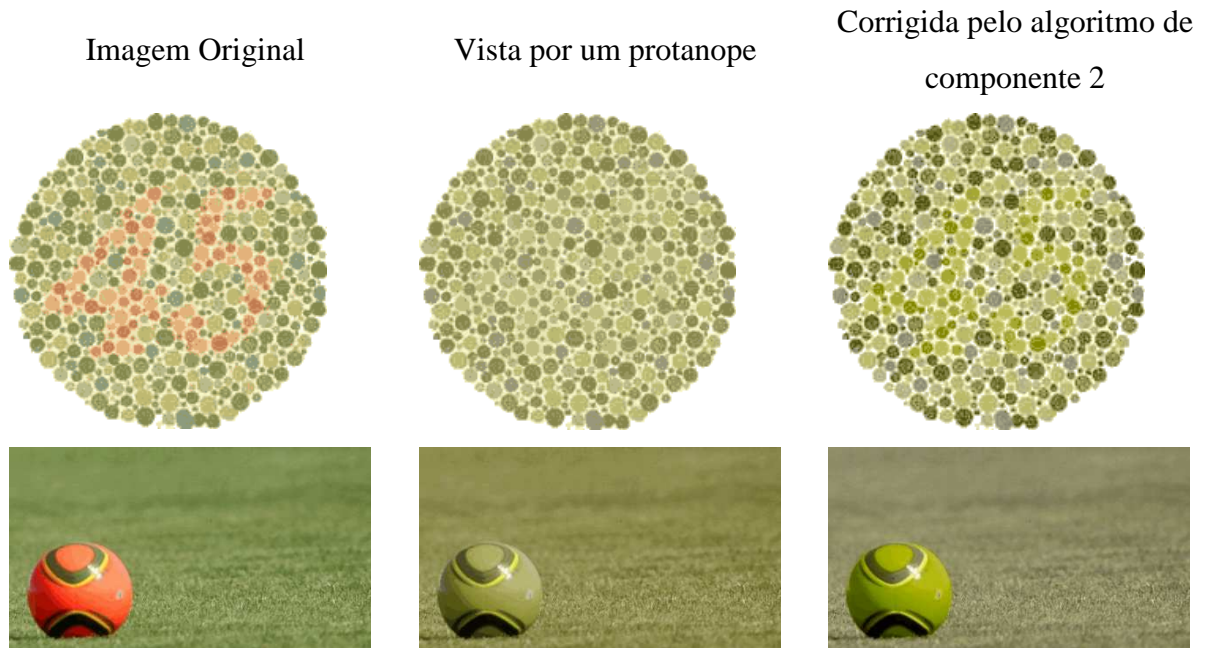


a2) representação gráfica no eixo a^* b^* das partículas quantizadas antes do afastamento de cores vizinhas

b2) representação gráfica no eixo a^* b^* das partículas quantizadas depois do afastamento de cores vizinhas

Figura 42 - Exemplo prático do afastamento das cores.

6.7 Resultados/conclusões



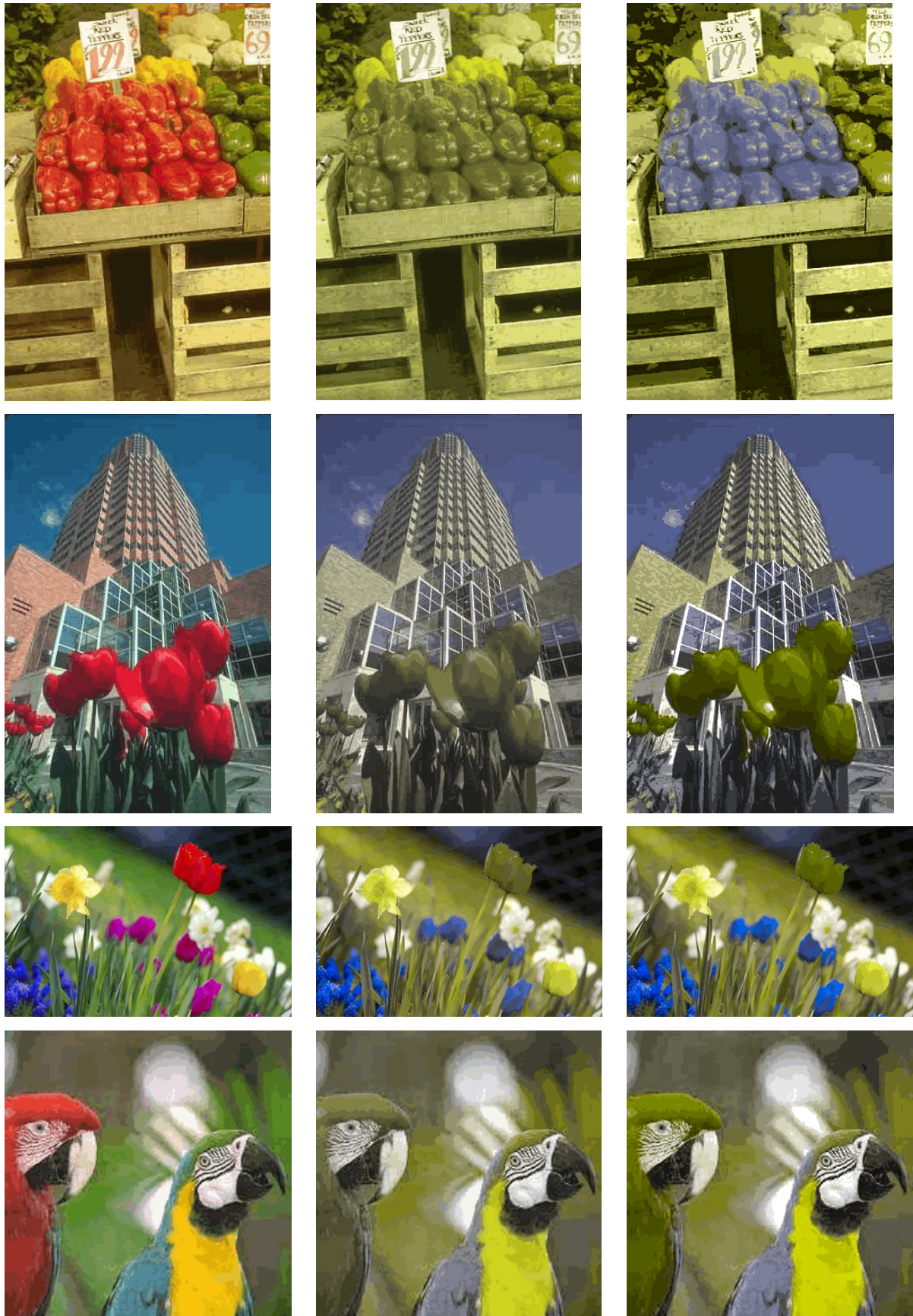


Figura 43 - Conjunto de imagens geradas pelo algoritmo proposto pela componente 2.

As imagens quantizadas foram geradas para 64 níveis (Apêndice (A.3.1), (A.3.2), (A.3.3)), a estratégia implementada no componente 2 cria contraste nas cores que são vizinhas, garante a diferenciação por parte do dicromata de regiões que antes não eram visíveis. Mantendo na medida do possível a naturalidade das cores.

Capítulo 7

7 Resultados

Neste capítulo apresentamos uma série de imagens comparativas, geradas pelo algoritmo proposto por Kuhn e pelo algoritmo proposto por nós para as 2 componentes.

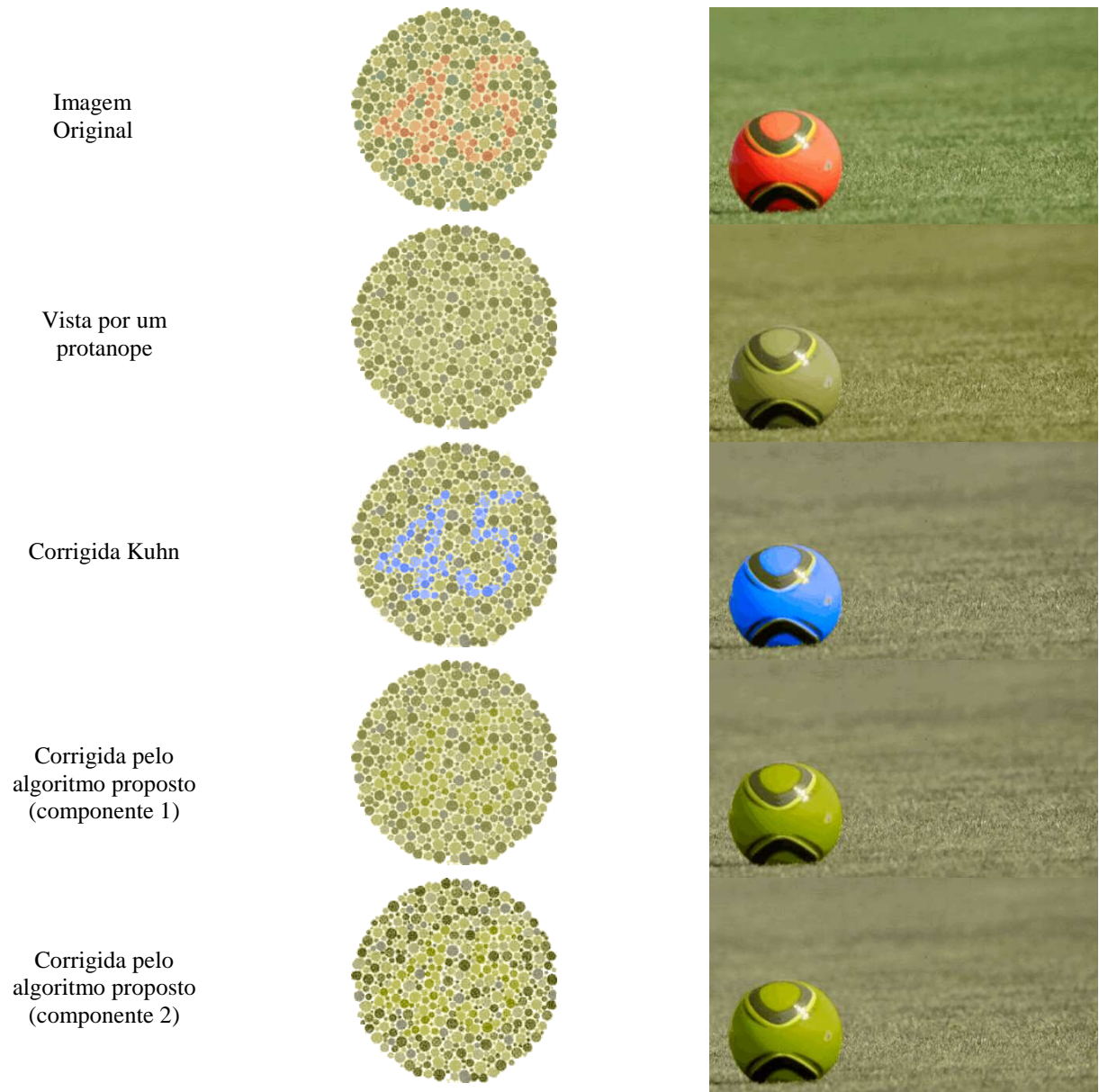


Imagem Original



Vista por um protanope



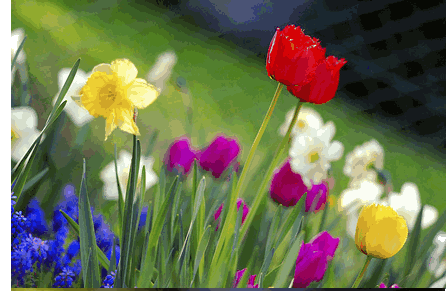
Corrigida Kuhn



Corrigida pelo algoritmo proposto (componente 1)



Corrigida pelo algoritmo proposto (componente 2)



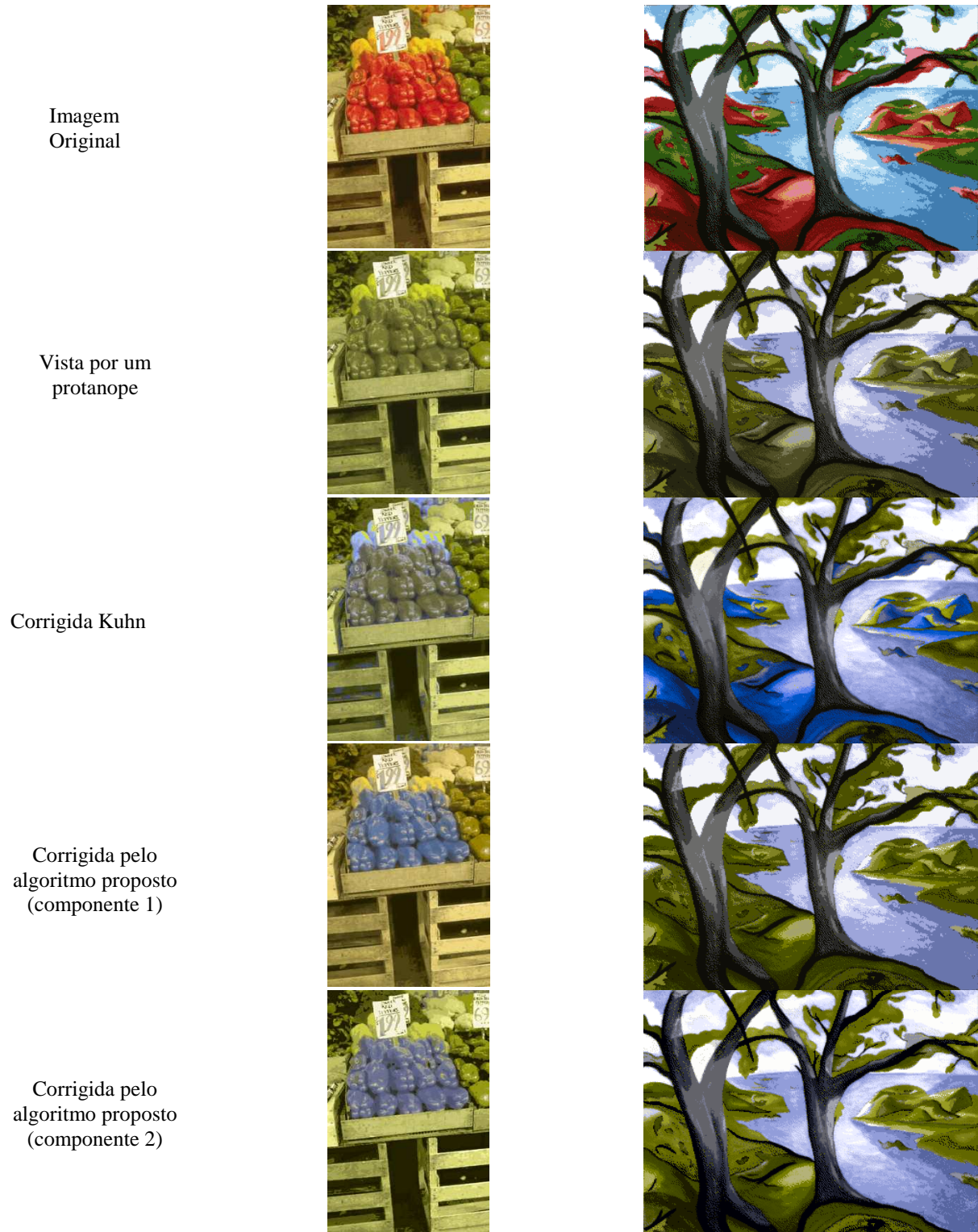


Figura 44 - Conjunto de correções "daltônicas" para os vários algoritmos.

As imagens quantizadas foram geradas para 64 níveis pelo algoritmo não-uniforme (Variância Mínima) (Apêndice (A.2), (A.3.1), (A.3.2), (A.3.3)), esta sequência de imagens (Figura (44))

mostra os diferentes resultados criados por cada um dos algoritmos. A principal evidência é a seguinte:

A implementação pela componente 2 cria ainda mais contraste relativamente ao contraste gerado pela implementação da componente 1 nas cores que são vizinhas, garante a diferenciação por parte do dicromata de regiões que antes não eram visíveis. Mantendo na medida do possível a naturalidade das cores. Estes resultados adicionam uma maior utilidade prática à solução proposta por Kuhn.

Capítulo 8

8 Conclusões

Nesta dissertação é proposto um novo algoritmo de re-coloração de imagens para DVC, na sua base está incorporado um processo de análise de textura da imagem que permite determinar quais as cores que terão prioridade de afastamento no plano visível do daltónico, com base em critérios de vizinhança espacial. Este processo é baseado num método de optimização, apresentado por Kuhn, orientado pelo contraste das cores originais.

Esse método funciona razoavelmente bem para algumas imagens. Porém, em alguns cenários de distribuições de cores, a própria correcção colide com cores já existentes. As duas componentes algorítmicas desenvolvidas pretendem evitar esses problemas de colisão no mapeamento das cores. Os testes efectuados mostram que as duas componentes desenvolvidas acrescentam desempenho funcional ao processo, permitindo aumentar a naturalidade de contraste de cores, nas imagens vistas pelos daltónicos, relativamente ao obtido pelo método de Kuhn isoladamente.

Os resultados permitem concluir que a componente 1 produz bons resultados, incorporando novos coeficientes de elasticidade, às representações das cores vizinhas, na fórmula de *hooke* (equação(5.2)). No entanto, este mecanismo, na tentativa de aumentar o contraste, de todas as cores, fica limitado pela pequena magnitude do espaço de cores daltónico. Já a componente 2 pretende afastar as cores, de uma forma ainda mais dirigida e prioritária, após a modelação baseada em *mass-spring*. Esta abordagem mostra ser a melhor na generalidade das imagens testadas. O número de pares de cores que se devem afastar, limita a capacidade de processamento do algoritmo e também limita a liberdade de deslocamento dessas cores no espaço de cores daltónico, impossibilitando correcções completas relativamente às representações das imagens originais que residem num espaço de dimensão mais elevada.

É interessante notar que para imagens específicas, o algoritmo de componente 1 devolve resultados superiores ao algoritmo de componente 2. Isto significa que a utilização de uma ou de outra componente estará indexada ao tipo de imagem a corrigir.

Num modo mais geral, a principal conclusão que resulta desta dissertação é que jamais poderá existir um algoritmo universalista de solução a este problema e que a solução concreta passa pela adição de múltiplos modos que tratem de forma diferente o mapeamento das cores mediante a semântica visual contida na imagem.

8.1 Trabalhos Futuros

Como trabalho futuro, este algoritmo poderá ser melhorado em algumas vertentes de desempenho e paralelização, de forma a acelerar a sua execução, possibilitando a implementação em GPUs (*Graphics Processing Unit*), possibilitando também a adequada execução em dispositivos móveis e até a sua distribuição por redes de servidores de correcção daltónica.

Bibliografia

[Art 2010]

ART, C.V. 2010. Goethe's Color Theory. *Vision Science & the emergence of modern art* <http://www.webexhibits.org/colorart/ch.html>.

[Bao, Wang, Ma and Gu 2008]

BAO, J.B., WANG, Y.Y., MA, Y. and GU, X.D. 2008. Re-coloring images for dichromats based on an improved adaptive mapping algorithm. *2008 International Conference on Audio, Language and Image Processing, Vols 1 and 2, Proceedings*, 152-156.

[Brettel, Vienot and Mollon 1997]

BRETTEL, H., VIENOT, F. and MOLLON, J.D. 1997. Computerized simulation of color appearance for dichromats. *Journal of the Optical Society of America a-Optics Image Science and Vision 14*, 2647-2655.

[G. Iaccarino 2006]

G. IACCARINO, D.M., MARCO DEL PERCIO AND VITTORIO SCARANO 2006. Efficient Edge-Services for Colorblind Users. *www' 2006* 919-920.

[Jefferson, Harvey and Acm 2007]

JEFFERSON, L., HARVEY, R. and ACM 2007. *An Interface to Support Color Blind Computer Users*. Assoc Computing Machinery, New York.

[JN 2008]

JN 2008. Código ajuda daltónicos a decifrar cores. *Jornal de Noticias*.

[Kuhn, Oliveira and Fernandes 2008]

KUHN, G.R., OLIVEIRA, M.M. and FERNANDES, L.A.F. 2008. An Efficient Naturalness-Preserving Image-Recoloring Method for Dichromats. *Ieee Transactions on Visualization and Computer Graphics 14*, 1747-1754.

[Ma, Gu and Wang 2006]

MA, Y., GU, X.D. and WANG, Y.Y. 2006. A new color blindness cure model based on BP neural network. *Advances in Neural Networks - Isnn 2006, Pt 3, Proceedings 3973*, 740-745.

[Ma, Gu and Wang 2009]

MA, Y., GU, X.D. and WANG, Y.Y. 2009. Color discrimination enhancement for dichromats using self-organizing color transformation. *Information Sciences 179*, 830-843.

[Machado and Oliveira 2010]

MACHADO, G.M. and OLIVEIRA, M.M. 2010. Real-Time Temporal-Coherent Color Contrast Enhancement for Dichromats. *Computer Graphics Forum 29*, 933-942.

[Ou-Yang and Huang 2007]

OU-YANG, M. and HUANG, S.W. 2007. Design considerations between color gamut and brightness for multi-primary color displays. *Journal of Display Technology 3*, 71-82.

[Rasche, Geist and Westall 2005]

RASCHE, K., GEIST, R. and WESTALL, J. 2005. Detail preserving reproduction of color images for monochromats and dichromats. *Ieee Computer Graphics and Applications* 25, 22-30.

[Rasche, Geist and Westall 2005]

RASCHE, K., GEIST, R. and WESTALL, J. 2005. Re-coloring images for gamuts of lower dimension. *Computer Graphics Forum* 24, 423-432.

[Vienot, Brettel and Mollon 1999]

VIENOT, F., BRETTEL, H. and MOLLON, J.D. 1999. Digital video colourmaps for checking the legibility of displays by dichromats. *Color Research and Application* 24, 243-252.

[Wade 2008]

WADE, B.D.E.A.R. 2008. vischeck.

[Yang and Ro 2003]

YANG, S. and RO, Y.M. 2003. Visual contents adaptation for color vision deficiency. *2003 International Conference on Image Processing, Vol 1, Proceedings*, 453-456.

Referências de figuras

[1]

http://pt.wikipedia.org/wiki/Ficheiro:Schematic_diagram_of_the_human_eye_pt.svg

[2]

http://pt.wikipedia.org/wiki/Ficheiro:Fig_retine.png

[3]

<http://upload.wikimedia.org/wikipedia/commons/a/a3/XlinkRecessive.jpg>

[4]

<http://pt.wikipedia.org/wiki/Ficheiro:Spectrum4websiteEval.png>

[5]

<http://en.wikipedia.org/wiki/File:Cone-fundamentals-with-srgb-spectrum.png>

[6]

imagem 1 - teste ishihara

http://www.uam.es/personal_pdi/medicina/algvilla/fundamentos/nervioso/Daltonismo/ishihara.htm

imagem 2 – Bola sob a relva

<http://www.zerozero.pt>

imagem 3 – legumes

Software Imaje J

Imagem 4 – aurora

Software Imaje J

[7]

<http://www.webexhibits.org/colorart/ch.html>

[8]

BRETTEL, H., VIENOT, F. and MOLLON, J.D. 1997. Computerized simulation of color appearance for dichromats. *Journal of the Optical Society of America a-Optics Image Science and Vision* 14, 2647-2655.

[9]

<http://corisectelmo.blogspot.com/2011/01/aula-21-espaco-de-cor-lab.html>

[10]

Software Matlab R2010 a toolbox\images\imdemos

[11]

<http://www.estig.ipb.pt>

[12]

Software livre ImageJ 1.45q

[13]

<http://www.mathworks.com/help/toolbox/images/f8-15484.html>

[14]

http://en.wikipedia.org/wiki/Octree#Application_to_color_quantization

[15]

<http://en.wikipedia.org/wiki/File:MassSpringNetwork.jpg> (Imagem Adaptada)

9 Apêndice A

A.1 Simulação Dicromancia

```
% arquivo anexo simulacao_dicromata.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Iniciar Constantes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
N_Q = 128; % Numero de cores quantizadas

theta = -11.48*(pi/180.0); % protanope
%theta = -8.11*(pi/180.0); % deuteranope
%theta = 46.37*(pi/180.0); % tritanope

p = zeros(3, N_Q);
umap = zeros(3, N_Q);

MB = [1 0 0; 0 0 0; 0 0 1]; % matriz de projecção no plano tridimensional

M_THETA = [1 0 0; 0 cos(theta) -sin(theta); 0 sin(theta) cos(theta)];% matriz de rotação no plano tridimensional

D = M_THETA\ MB * M_THETA; % matriz de projecção orthografic no plano dos dicromatas

lims = [-100 100 -100 100 0 100]; % LIMITES ORIGINAL

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Carregar imagem
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```

RGB = imread('bola_relva.jpg');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Quantitization NÃO UNIFORME da Imagem de entrada
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[X,map] = rgb2ind(RGB,N_Q,'nodither');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Converte RGB 2 LAB
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

cform = makecform('srgb2lab');
lab = applycform(RGB, cform);
lab_map = applycform(map, cform);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Trasnssforma Pontos q em p
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:N_Q
    p(1:3, i) = (MB * M_THETA) * lab_map(i, 1:3)';
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Simula Imagem Vista pelo Daltonico
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:N_Q
    umap(1:3, i) = M_THETA\ p(1:3, i);
end

umap = umap';

cform = makecform('lab2srgb');

```

```

map_rgb = applycform(umap, cform);
RGBB = ind2rgb(X, map_rgb);
figure(1), imshow(X,map_rgb);

```

A.2 Algoritmo de Proposto por Kuhn

```

% ficheiro anexo mass_spring_kuhn_final.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Iniciar Constantes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

N_Q = 16; % N° cores a quantizar
ITER = 300;
k= 1;
tau = 15;
theta = -11.48*(pi/180.0); % protanope
%theta = -8.11*(pi/180.0); % deuteranope
%theta = 46.37*(pi/180.0); % tritanope

p = zeros(3, N_Q);
umap = zeros(3, N_Q);
m = zeros(N_Q);
l = zeros(N_Q, N_Q);
ll = zeros(N_Q, N_Q);
F=zeros(N_Q);

tol = 0.1;

MB = [1 0 0; 0 0 0; 0 0 1]; % Projecção

M_THETA = [1 0 0; 0 cos(theta) -sin(theta); 0 sin(theta) cos(theta)]; % Rotação

D = M_THETA\ MB * M_THETA; %Matriz de Projecção Ortogonal

lims = [-100 100 -100 100 0 100]; % Limites Lab

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Carregar Imagem
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

RGB = imread('45.png');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Quantitization NÃO UNIFORME da Imagem de entrada
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[X,map] = rgb2ind(RGB,N_Q,'nodither');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Mostra Imagem Carregada Quantizada
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%RGB = ind2rgb(X,map);
figure(1), imshow(X,map);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Converte RGB 2 LAB
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

cform = makecform('srgb2lab');
lab = applycform(RGB, cform);
lab_map = applycform(map, cform);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Transformação dos Pontos q em p
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:N_Q
    p(1:3, i) = M_THETA * D * lab_map(i, 1:3)';
end

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Mostrar Imagem Vista Pelo Daltonico
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:N_Q
    umap(1:3, i) = M_THETA\ p(1:3, i);
end

umap = umap';

cform = makecform('lab2srgb');
map_rgb = applycform(umap, cform);
RGBB = ind2rgb(X, map_rgb);
figure(2), imshow(X,map_rgb);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Calculo de Máximos, Minimos e Média Locais dos Pontos P
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

p_max=max(p(3,:));
p_min=min(p(3,:));
mean_p = mean(p(3,:));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Calculo das massas de cada Particula
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:N_Q
    d = [lab_map(i, 1:3); (D*lab_map(i, 1:3))'];
    m(i) = 1.0/(pdist(d));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Comprimento das Molas das cores quantizadas

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:N_Q
    for j=1: N_Q
        l(i,j) = pdist( [lab_map(i, 2:3); lab_map(j, 2:3)] );
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Iniciar variáveis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
p_old_old = p(3, 1:N_Q);
p_old = p(3, 1:N_Q);
t=0;

while(t<ITER)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Comprimento Actual das Molas
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:N_Q
    for j=1: N_Q
        ll(i,j) = pdist( [p(2:3, i)'; p(2:3, j)'] );
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Calculo das Forças
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

for i=1:N_Q
    F(i) = 0;
    for j=1:N_Q

        if( l(i,j) > ll(i,j) )
            if( l(i,j) ~= 0 )
                F(i) = F(i) -k * ( 1.0 - ll(i,j)/l(i,j) ) * (p(3, j)-p(3, i));
            end
        else
            if(ll(i,j) ~= 0 )
                F(i) = F(i) + k * ( 1.0 - l(i,j)/ll(i,j) ) * (p(3, j)-p(3, i));
            end
        end

    end

end
end

```

```
end
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Calculo das Novas posições
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

for i=1:N_Q
    p_v = F(i)/m(i) + 0.5*p(3, i) + 0.5*p_old(i);
    p_old(i) = p(3,i);
    p(3, i) = p_v;
end

```

```

t = t+1;
end % while

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%Evitar os Minimios Locais

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:N_Q
    rd = pdist( [lab_map(i, 1:3); (D*lab_map(i, 1:3)')]' );

    if(lab_map(i, 2)>0 && rd>tau)
        p(3, i) = -p(3,i);
    end

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% inversion of tranformation to see the dichromatic image
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear umap;

    umap(1:3, :) = M_THETA\p(1:3, :);

umap = umap';
cform = makecform('lab2srgb');
map_rgb = applycform(umap, cform);
RGB = ind2rgb(X, map_rgb);
figure(3), imshow(X,map_rgb);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%Verificar se os pontos P são colineares
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(33);

    hold on;
    grid on;

    for i=1:N_Q

        plot3(umap(i,2), umap(i,3), umap(i,1), '.', 'MarkerSize', 20);
    end

```

```
end
```

A.3 Algoritmo Proposto componente 1 e 2

A.3.1 Final

```
% ficheiro anexo final.m

close all
addpath([docroot ' /techdoc/creating_plots/examples' ])

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Iniciar Constantes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

N_Q = 128;
ITER = 300;
k= 1;

theta = -11.48*(pi/180.0);
%theta = -8.11*(pi/180.0); % deuteranope
%theta = 46.37*(pi/180.0); % tritanope

p = zeros(3, N_Q);
umap = zeros(3, N_Q);
umapp = zeros(3, N_Q);
umapdicromatasimulacao = zeros(3, N_Q);
m = zeros(N_Q);
l = zeros(N_Q, N_Q);
ll = zeros(N_Q, N_Q);
F=zeros(N_Q);
```

```

MB = [1 0 0; 0 0 0; 0 0 1]; % Projecção
M_THETA = [1 0 0; 0 cos(theta) -sin(theta); 0 sin(theta) cos(theta)]; % Rotação
D = M_THETA\ MB * M_THETA; %Matriz de Projecção Ortogonal
lims = [-100 100 -100 100 0 100]; % Limites Lab

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Carregar Imagem
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

RGB = imread('jardim.png');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Quantitization NÃO UNIFORME da Imagem de entrada
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[X,map] = rgb2ind(RGB,N_Q,'nodither');

%quantização uniforme
%tol= 0.5;
%n = (floor(1/tol)+1)^3;
%[X,map] = rgb2ind(RGB,n);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Iniciar processo de análise sde textura da imagem quantizada
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Obter n cores vizinhas
nPrimeiras = 8;%40;%2%40;
fatorAfastamento = 1.8;

abordagem1_k = false ; % activar abordagem 1 coeficiente de elasticidade

```

```
abordagem1_1 = false ; % activar abordagem 1 comprimento das molas
abordagem2 = true ; % activar abordagem 2 afastar pares uma % para cada lado
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Atribuição das constantes de k e l
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
kn = 1.3; %1.3;%1.23;          esquisitos % 1.9
ln = 2; %2.0;%2.1;           esquisitos % 20
dist = 3;
```

```
direccoes = [ [-dist -dist]; [-dist 0]; [-dist dist]; [0 dist]; [dist -dist]; [dist 0]; [dist dist]; [0 -dist] ]
coresVizinhas = [];
```

```
for i=1:8
[ tabelaDePesos, mapaAlterado, mapaDeCores , coresVizinhasTemp] = Segment( X, map,nPrimeiras,direccoes(i,:) );
    coresVizinhas = cat(1,coresVizinhas,coresVizinhasTemp)
end
```

```
coresVizinhas = Unique(coresVizinhas,'rows') %os valores podem ter repetido em diferentes direções, pegando apenas
únicos
```

```
%coresVizinhas =[2 5;4 14;5 12];;%5 7]
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Mostra Imagem Carregada Quantizada
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
RGB = ind2rgb(X,map);
figure(1), imshow(X,map);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Grafico de Particulas Pares de cores vizinhas da imagem quantizada
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure(11)

xlabel('X');
ylabel('Y');

grid on;
hold on
axis([0 1 0 1]);
size(coresVizinhas)
for i=1:size(coresVizinhas)
    indice1 = coresVizinhas(i,1);
    indice2 = coresVizinhas(i,2);
    cor1 = map(indice1+1,:);
    cor2 = map(indice2+1,:);

    pos1 = dsxy2figxy([cor1(1,1) cor1(1,2) 0 0]);
    pos2 = dsxy2figxy([cor2(1,1) cor2(1,2) 0 0]);
    [l1 l2] = dsxy2figxy(gca,[cor1(1,1) cor1(1,2)],[cor2(1,1) cor2(1,2)]);

    plot( cor1(1,1), cor1(1,2), '.', 'MarkerSize',50, 'color', cor1);
    plot( cor2(1,1), cor2(1,2), '.', 'MarkerSize',50, 'color', cor2);
    annotation('textbox',pos1, 'String', indice1, 'LineStyle','none');
    annotation('textbox',pos2, 'String', indice2, 'LineStyle','none');
    annotation('line',[pos1(1,1) pos2(1,1)],[pos1(1,2) pos2(1,2)], 'color', cor1);
    %cor(1,1), cor(1,2),'String','a');

```

```

end

hold off;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Grafico de Particulas da imagem quantizada
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure(111);
xlabel('X');
ylabel('Y');

grid on;
hold on
axis([0 1 0 1]);

for i=1:N_Q
    indice1 = i;%coresVizinhas(i,1);

    cor1 = map(indice1,:);

    pos1 = dsxy2figxy([cor1(1,1) cor1(1,2) 0 0]);

    plot( cor1(1,1), cor1(1,2), '.', 'MarkerSize',50, 'color', cor1);
    annotation('textbox',pos1, 'String', indice1-1, 'LineStyle','none');

    %cor(1,1), cor(1,2),'String','a');

end

hold off

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Convert RGB 2 LAB
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

cform = makecform('srgb2lab');
lab = applycform(RGB, cform);
lab_map = applycform(map, cform);

% grafico para verificar se os pontos são colineares
figure(1111);

xlabel('X');
ylabel('Y');

grid on;
hold on

    for i=1:N_Q

        plot3(lab_map(i,2), lab_map(i,3), lab_map(i,1), '.', 'MarkerSize', 20);

    end

    hold off;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Transformação dos Pontos q em p
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:N_Q
    p(1:3, i) = M_THETA * D * lab_map(i, 1:3)';
end

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Mostra imagem aproximanda vista pelo daltonico com angulo Theta
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:N_Q

    umapdicromatasimulacao(1:3, i) = M_THETA\ p(1:3, i);

end

umapdicromatasimulacao = umapdicromatasimulacao';

cform = makecform('lab2srgb');
map_rgb = applycform(umapdicromatasimulacao, cform);
RGBB = ind2rgb(X, map_rgb);
figure(2), imshow(X,map_rgb);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% GRAFICO - Abordagem 1 - Grafico simulação daltonico com angulo Theta
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure(22);

    hold on;
    grid on;

    for i=1:N_Q

        plot3(umapdicromatasimulacao(i,2), umapdicromatasimulacao(i,3), umapdicromatasimulacao(i,1),
            '.', 'MarkerSize', 20);

    end

    hold off;

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  Calculo de Máximos, Minimos e Média Locais dos Pontos P
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

p_max=max(p(3,:));
p_min=min(p(3,:));
mean_p = mean(p(3,:));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Calculo das massas de cada Particula
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:N_Q
d = [lab_map(i, 1:3); (D*lab_map(i, 1:3)')'];
m(i) = 1/(pdist(d));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Comprimento das Molas das cores quantizadas
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:N_Q

    for j=1: N_Q

        l(i,j) = pdist( [lab_map(i, 2:3); lab_map(j, 2:3)] );

    end

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Alterar o comprimento das molas para as cores vizinhas Abordagem 1

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tamanho = size(coresVizinhas)

if(abordagem1_1)
    for contador = 1:tamanho(1)

        i=coresVizinhas(contador,1)+1;
        j=coresVizinhas(contador,2)+1;
        l(i,j)= l(i,j)*l(i,j);
        l(j,i)= l(j,i)*l(i,j);

    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Iniciar variáveis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
p_old_old = p(3, 1:N_Q);
p_old = p(3, 1:N_Q);
t=0;

while(t<ITER)

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %% Comprimnto Actual das Molas
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    for i=1:N_Q

        for j=1: N_Q

            ll(i,j) = pdist( [p(2:3, i)'; p(2:3, j)'] );
        end
    end
end

```

```
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Calculo das Forças
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:N_Q
    F(i) = 0;
    for j=1:N_Q

        kl = k;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Alterar o coeficiente de elasticidade para as cores vizinhas abordagem 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        if(abordagem1_k)
            for cont=1:tamanho(1)
                cori=coresVizinhas(cont,1)+1;
                corj=coresVizinhas(cont,2)+1;

                if((i==cori)&&(j==corj))
                    kl=kn*k;
                end
                if((i==corj)&&(j==cori))
                    kl=kn*k;
                end
            end
        end

        if( l(i,j) > ll(i,j) )
            if( l(i,j) ~= 0 )
                F(i) = F(i) -kl *( 1.0 - ll(i,j)/l(i,j) ) * (p(3, j)-p(3, i));
            else
                F(i) = F(i) + kl *( ll(i,j)) * ( p(3, j)-p(3, i) );
            end
        end
    end
end
```

```

else
  if(ll(i,j) ~= 0 )
    F(i) = F(i) + k1 * ( 1.0 - l(i,j)/ll(i,j) ) * (p(3, j)-p(3, i));
  else
    F(i) = F(i) + k1 * (l(i,j)) * ( p(3, j)-p(3, i) );
  end
end

end

end

```

```
end
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Calculo das Novas posições
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

for i=1:N_Q

  p_v = F(i)/m(i) + 0.5*p(3, i) + 0.5*p_old(i);
  p_old(i) = p(3,i);
  p(3, i) = p_v;

```

```
end
```

```

t = t+1;
end % while

```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```

% Mostrar imagem com pontos P alinhados com a*=0
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:N_Q
    umapp(1:3, i) = p(1:3, i);
end

umapp = umapp';
cform = makecform('lab2srgb');
map_rgbp = applycform(umapp, cform);
RGBBP = ind2rgb(X, map_rgbp);

figure(3), imshow(X,map_rgbp);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% GRAFICO - Abordagem 1 - pontos P alinhados com a*=0
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure(33);

    hold on;
    grid on;

    for i=1:N_Q

        plot3(umapp(i,2), umapp(i,3), umapp(i,1), '.', 'MarkerSize',20);

    end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Imagem Projectada no plano SEM AFASTAMENTO Abordagem 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear umapp;
clear cform;

```

```

clear RGB;
umap(1:3, :) = M_THETA\ p(1:3, :);
umap = umap';

cform = makecform('lab2srgb');
map_rgb = applycform(umap, cform);
RGB = ind2rgb(X, map_rgb);

figure(4), imshow(X,map_rgb);

%antes de afastar
figure(44);
hold on
axis square;
xlabel('a');
ylabel('b');
zlabel('L');
grid on;
for i=1:N_Q
    l_a_b = [umap(i,1) umap(i,2) umap(i,3)];
    cform = makecform('lab2srgb');
    r_g_b = applycform(l_a_b, cform);
    plot3( umap(i,2), umap(i,3), umap(i,1), '.', 'MarkerSize',25, 'color', r_g_b );
end
hold off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Imagem Projectada no plano COM AFASTAMENTO Abordagem 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if(abordagem2)

umapAfastado = umap; % MAPA PARA AFASTAR CORES

```

```

percentualAfastamento = fatorAfastamento/size(coresVizinhas,1);

tamvis = size(coresVizinhas,1);

for i=1:size(coresVizinhas)

    %coresVizinhas eh zero-based
    nCor1 = coresVizinhas(i,1) +1;
    nCor2 = coresVizinhas(i,2) +1;
    [novaCorA novaCorB] = Afastar(umapAfastado(nCor1,:), umapAfastado(nCor2,:), percentualAfastamento);
    umapAfastado(nCor1,:) = novaCorA;
    umapAfastado(nCor2,:) = novaCorB;

end

cform = makecform('lab2srgb');
map_rgbAfastado = applycform(umapAfastado, cform);
RGBAfastado = ind2rgb(X, map_rgbAfastado);

figure(5), imshow(X,map_rgbAfastado);

%apos afastar

figure(55);
clf;
hold on
axis square;
xlabel('a');
ylabel('b');
zlabel('L');
grid on;
for i=1:N_Q
    l_a_b = [umapAfastado(i,1) umapAfastado(i,2) umapAfastado(i,3)];
    cform = makecform('lab2srgb');
    r_g_b = applycform(l_a_b, cform);
    plot3( umapAfastado(i,2), umapAfastado(i,3), umapAfastado(i,1), '.', 'MarkerSize',25, 'color', r_g_b );
end

```

```
end  
hold off
```

```
end
```

A.3.2 Matriz de co-ocorrência

```
% ficheiro anexo segment.m
```

```
function [ tabelaDePesos, mapaAlterado, mapaDeCores , coresVizinhasLocais] = Segment( image, mapa,nPrimeiras,  
direcao)
```

```
%SEGMENT Recebe uma imagem como entrada (mapa e cores quantizadas) e  
%devolve as cores que precisam ficar mais distantes em uma tabela com  
%colunas cor1, cor2 e distância
```

```
% A distância das cores será dada pela segmentação de regiões da imagem
```

```
tamanhoMapa = size(mapa);  
numeroCores = tamanhoMapa(1);  
tamanho = size(image);  
mapaCores = zeros(numeroCores,2);
```

```
[glcma,SI] = graycomatrix(image,'Offset',direcao,'NumLevels',numeroCores,'GrayLimits',[ ]);
```

```
tabelaDePesos = glcma;  
mapaAlterado = SI;
```

```
for i=1:numeroCores  
    tabelaDePesos(i,i)=0;  
end
```

```
%mapa de cores
```

```
for i=1:tamanho(1)  
    for j=1:tamanho(2)  
        mapaCores( image(i,j)+1, 1) = image(i,j);
```

```

        mapaCores( image(i,j)+1, 2) = SI(i,j);
    end
end

%cores mais vizinhas:
[tabelaOrdenada,tabelaIndex] = sort(tabelaDePesos(:),'descend');
coresVizinhasLocais = zeros(1,2);
j=0;

cform = makecform('srgb2lab');

for i=1:size(tabelaIndex)

    [posr,posc] = ind2sub(size(tabelaDePesos),tabelaIndex(i));
    if(posr==posc)
        continue;
    end

    [r,~] = find(mapaCores(:,2)==posr);
    cor1 = mapaCores(r,1);
    [r,~] = find(mapaCores(:,2)==posc);
    cor2 = mapaCores(r,1);

    lab1 = applycform( mapa(cor1+1,:), cform);
    lab2 = applycform( mapa(cor2+1,:), cform);
    distancia = norm( lab1 - lab2)

    %todo fazer o threshold funcionar
    if (distancia > 40)
        j = j+1;
        coresVizinhasLocais(j,1) = cor1;
        coresVizinhasLocais(j,2) = cor2;
        coresVizinhasLocais(j,3) = distancia; % guarda a distancia na coluna 3
    end
    if(j==nPrimeiras)

```

```
    break;
end
```

```
end
%[~,pos] = max(tabelaDePesos(:));
%[posr,posc] = ind2sub(size(tabelaDePesos),pos);

%[r,~] = find(mapaCores(:,2)==posr);
%cor1 = mapaCores(r,1);
%[r,~] = find(mapaCores(:,2)==posc);
%cor2 = mapaCores(r,1);

%coresVizinhas = zeros(2,1);
%coresVizinhas(1) = cor1;
%coresVizinhas(2) = cor2;
mapaDeCores = mapaCores;
%coresMaisVizinhasOriginais = [mapaCores(pos(2)

end
```

A.3.3 Afastar cores vizinhas

```
% ficheiro anexo afastar.m
function [ novaCorA, novaCorB ] = Afastar( corA, corB, percentual )
%Afastar Dadas duas cores, faz com que a sua distância aumente de
%percentual, mantendo a direção do vetor distância
    vectorDistancia = corB - corA;
    versorDistancia = vectorDistancia/norm(vectorDistancia);
    moduloDistancia = norm(vectorDistancia);
    distanciaAIncrementar = moduloDistancia*percentual/2; %dividir por 2, pois é uma vez para cada lado
    novaCorB = corB + distanciaAIncrementar*versorDistancia;
```

```
novaCorA = corA - distanciaAIncrementar*versorDistancia;
% figure;
%
% hold on;
% grid on;
% plot3(corA(1,1), corA(1,2), corA(1,3), '.', 'MarkerSize', 15);
% plot3(corB(1,1), corB(1,2), corB(1,3), '.', 'MarkerSize', 30);
% plot3(novaCorA(1,1), novaCorA(1,2), novaCorA(1,3), 'x', 'MarkerSize', 15);
% plot3(novaCorB(1,1), novaCorB(1,2), novaCorB(1,3), 'x', 'MarkerSize', 30);
%
% hold off;
end
```