



Imagem

Computação Gráfica
Prof. Bernardo Lima



Agenda

- Sobre a reposição
- Trabalhos da disciplina
- Imagem
 - Definições
 - Histograma
 - Quantização
 - Dither
 - Halftone
 - Compressão
- Atividades (Prática de Laboratório)



Sobre a Reposição

- Vamos conversar
 - Quem sou eu?
 - Quem são vocês?
 - O que foi decidido sobre a avaliação da disciplina?
 - Que atividades o professor Érick disponibilizou para vocês?
 - O que vocês decidiram sobre os seus projetos?
 - Caso não tenham, decidam hoje :)



Sobre a Reposição

- Dinâmica das Aulas
 - Exposição e discussão (~50 minutos)
 - Prática de laboratório (~100 minutos)
 - Elaboração do projeto (~50 minutos, ou o tempo que sobrar)
- Tempos das atividades podem variar conforme nossa energia para discutir e resolver os problemas
- Temos quatro encontros neste formato, o último consistirá nos seminários dos projetos



Trabalhos da Disciplina

- N1
 - Vou assumir que todos(as) têm 6,0 até agora.
 - Cada uma das quatro práticas de laboratório subsequentes valem 1 ponto de N1.
- N2
 - Vou assumir que todos(as) fizeram a apresentação parcial do projeto e tiraram a nota máxima
 - Todas as apresentações serão no último encontro
 - 15 minutos de apresentação por projeto
- Relatórios
 - Quem me entregar o relatório final pontua

Imagem

- Uma imagem na computação é uma representação vetorial/matricial, discreta, de cores, luminância ou os dois.
- Por ser vetorial e discreta, trivialmente pode ser representada por um array
- A resolução da imagem indica o tamanho desse array, ou seja, a quantidade de pixels que temos disponíveis para trabalhar



187	163	174	168	160	162	129	163	172	161	165	166
165	182	163	74	75	62	33	17	110	210	380	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	6	124	131	111	120	204	166	16	96	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	180	216	211	168	138	78	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	168	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	103	36	101	255	224
190	214	173	66	103	143	96	50	2	108	249	215
187	196	236	73	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	158	200	136	343	236
195	206	123	207	177	121	123	200	178	13	96	218

167	163	174	168	160	162	129	161	172	161	165	166
165	182	163	74	75	62	33	17	110	210	380	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	6	124	131	111	120	204	166	16	96	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	180	216	211	168	138	78	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	168	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	103	36	101	255	224
190	214	173	66	103	143	96	50	2	108	249	215
187	196	236	73	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	158	200	136	343	236
195	206	123	207	177	121	123	200	178	13	96	218



Imagem

- As imagens em tom de cinza, possuem apenas uma camada.
- Imagens coloridas possuem pelo menos 3 camadas, uma para o vermelho, outra para o verde e outra para o azul.
- Também existe uma quarta camada de transparência (ou alpha), que indica a opacidade do pixel.
- Normalmente, trabalhamos com imagens de 8 bits por pixel, ou seja, temos a potencial variação de valores de 0 a 255 em cada pixel da imagem.
- Também existem imagens de mais bits, como a de 16 bits (variando de 0 até 65536), que é naturalmente reconhecida como imagem em qualquer programa de interpretação ou edição de imagens.

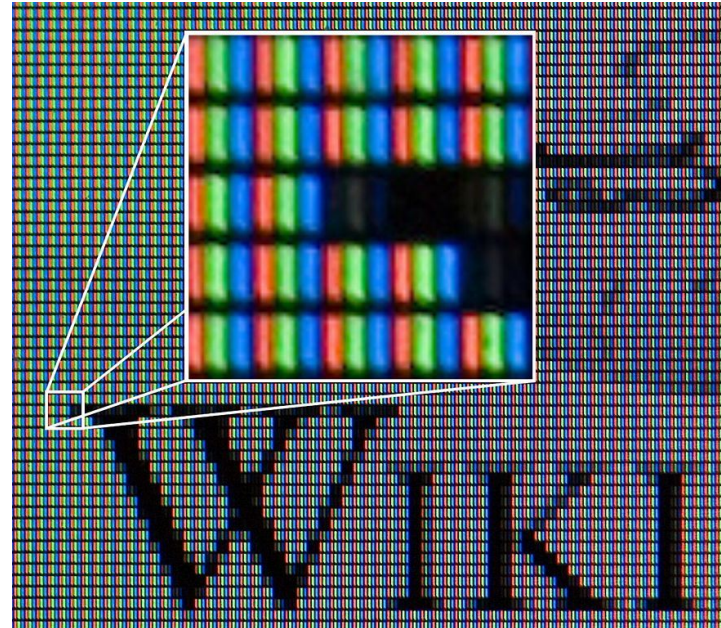


Formatos de Imagem

- Um exemplo de formato de imagem que podemos utilizar para trabalhar é o PGM. Sua especificação está disponível em <https://netpbm.sourceforge.net/doc/pgm.html>
- A representação das imagens PGM é bem próxima de como ela seria representada em formato de matriz, sendo um bom formato para trabalhar com imagens sem se preocupar muito com as nuances do formato
- Os pixels das imagens PGM podem ser de 8 ou 16 bits, isso é configurado no cabeçalho da imagem

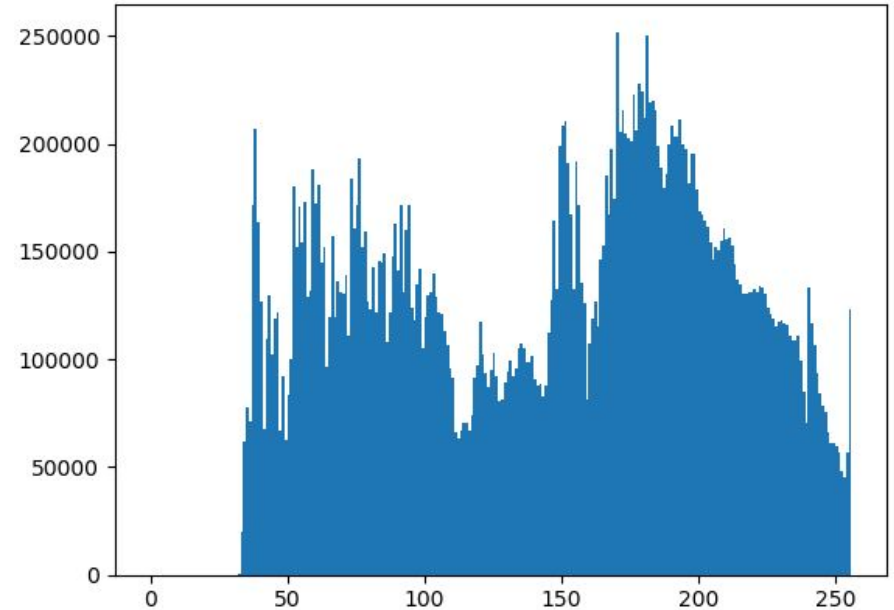
Monitor

- Uma tela de LED, por exemplo, possui 3 LEDs que acendem em cada pixel.
- A “mescla” das intensidades de cada LED dá a cor final ao pixel que percebemos.
- Pelo fato dos LEDs serem muito pequenos, nossos olhos não conseguem distinguir as cores individualmente e enxerga a combinação das intensidades.



Histograma

- O histograma é uma representação das distribuições de tons da imagem (i.e., como os valores dos pixels variam e estão distribuídos).



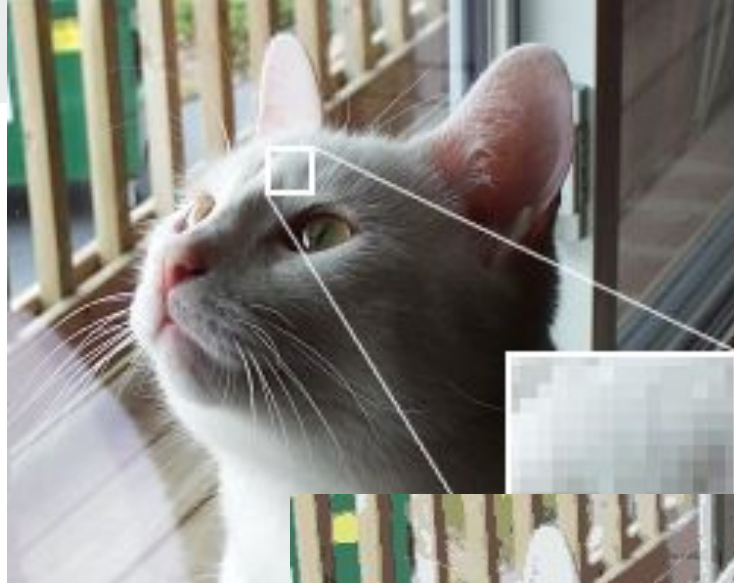


Aplicações dos histogramas

- Visualizar e ajustar brilho e contraste
- Os picos e vales do histograma podem servir como limiares para segmentação
- Correlacionar imagens por meio das frequências das tonalidades e intensidades
- Extrair *features* das imagens, especialmente se as features forem coloridas

Quantização

- A quantização, também vista pelos nomes posterização ou discretização, consiste em representar cores com menos bits
- Esta ação está intimamente ligada com a compressão de dados. Tendo menos informação, conseguimos utilizar menos bits por camada, reduzindo o tamanho final da imagem.
- Podemos utilizar os histogramas para quantizar, mapeando cada intervalo de intensidades a somente uma cor
- De que outras maneiras podemos fazer esse processo?





Quantização

- É possível clusterizar as cores originais da imagem de acordo com a cor final mais próxima (dentro das possibilidades).
- Como definir quais são as cores finais?
- Existem diversas formas de realizar essa operação, e os resultados serão diferentes (visualmente).

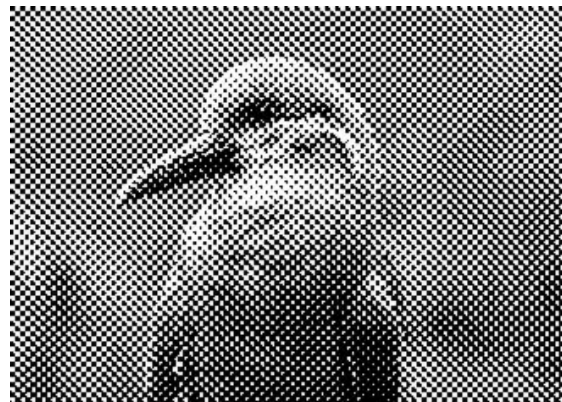


Algoritmo de Populosidade

- Esse método constrói inicialmente o histograma de frequência da imagem e, em seguida, escolhe para os K níveis de quantização as K cores que correm com maior frequência.
- O problema dessa abordagem é que visualmente o resultado pode não ser tão interessante, existem outras abordagens.
- Também é possível escolher outras cores e não necessariamente as mais frequentes. Quem sabe algumas que estão entre as mais frequentes?

Dither

- O dither era muito utilizado, especialmente em jogos antigos, para produzir impressão de “mais cores” sem utilizar mais cores, reduzindo, portanto, o tamanho da imagem.
- Técnicas parecidas são utilizadas em impressão de imagens em papel, para economizar tinta.
- O dither é baseado na percepção humana, explorando o fato de que nosso cérebro tende a interpretar certas coisas de forma “contínua”.



Dither

(Original)



Threshold



Random



Halftone



Ordered (Bayer)



Ordered (void-and-cluster)



Halftone

- O halftone é um algoritmo particular de dithering que usa “pontos” de diferentes tamanhos.
- Isso implica em bastante economia de tinta.
- É bastante utilizado na impressão de jornais, por exemplo.
- É um dos métodos de dither que apresenta um visual agradável.





Halftone

Dividir a imagem em quadrados menores

Para cada quadrado:

- Calcular a média da intensidade dos pixels da região

- Desenhar na imagem de saída um círculo, cujo diâmetro é definido pela intensidade média dos pixels (quanto mais intensidade, maior o círculo)

Dithering de Bayer

- O algoritmo ordenado de Bayer utiliza padrões de dither pré-definidos, e calcula o erro de pedaços da imagem com relação a esses padrões, como pode ser visto abaixo:



- O algoritmo void and cluster pega essa ideia e adiciona um certo ruído para uma cobertura mais uniforme.

Compressão

- Existem diferentes tipos de algoritmos para compressão, o mais básico talvez seja o run-length encoding (i.e., encoding de comprimento corrido, para vetores 1d, ou texto).
- aaabb vira 3a2b
- Em sua versão de imagens, se chama run length matrix.

0	3	2	0	0	1	1
0	1	0	3	4	0	0
3	4	1	2	0	3	4
2	0	0	2	0	3	0
2	4	2	4	1	1	4
1	2	2	0	0	0	0
2	1	2	1	2	3	1

Input Image

Run Length Matrix(0^0)					$\leftarrow \ell$
	1	2	3	4	
0	6	3	0	1	
1	6	2	0	0	
2	9	1	0	0	
3	6	0	0	0	
4	6	0	0	0	

\uparrow
 r



Atividades (Prática de Laboratório)

1. Criem um programa que recebe uma imagem como entrada, e como saída disponibiliza um arquivo json contendo seu histograma
2. Criem um programa que recebe uma imagem como entrada, aplica o algoritmo halftone sobre a imagem e disponibiliza o resultado em uma imagem de saída
3. Criem um programa que comprime uma imagem utilizando o run length matrix, recebendo uma imagem como entrada e disponibilizando a matriz resultante em um arquivo json
4. Criem um programa que recebe o arquivo gerado pelo programa feito no exercício anterior e desfaz a compressão

Para fazer os exercícios, podem utilizar imagens PGM ou outro formato mais confortável. No exercício de halftone, seria interessante ter uma imagem PGM de entrada e uma PBM de saída.