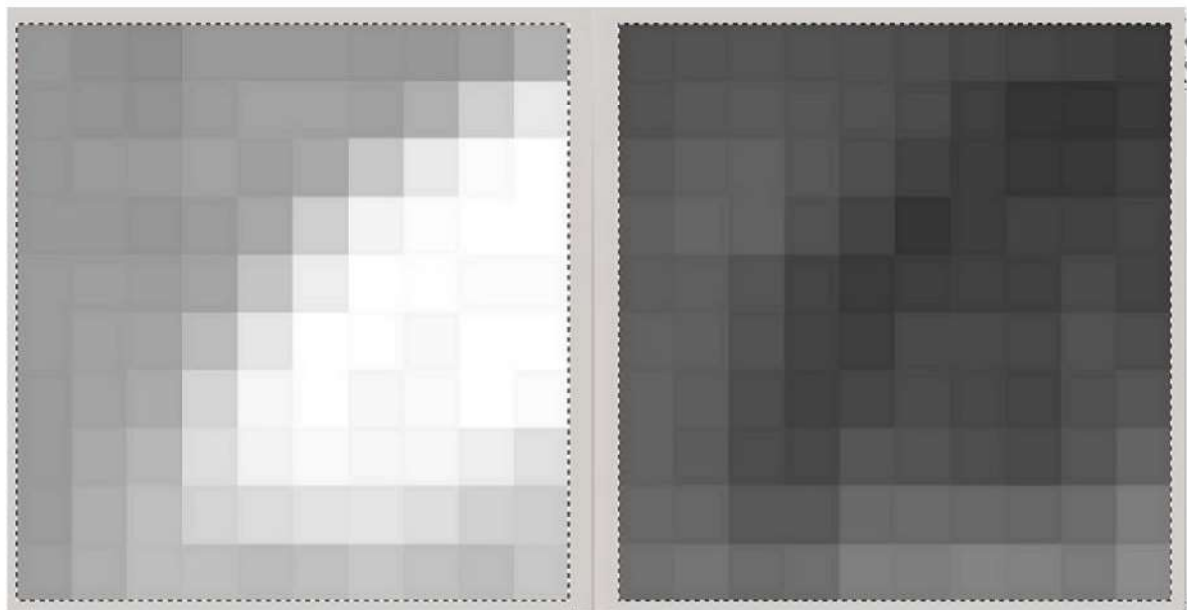


## UNIDADE 3 - Pré-processamento e Filtros

A etapa de pré-processamento de um sistema de visão computacional realiza transformações nas imagens originais buscando novas imagens que sejam mais fáceis de serem tratadas nas etapas posteriores. Basicamente, busca-se nessa fase novas imagens em que o ruído seja de alguma forma reduzido, mesmo que as custas de alguma perda de informação em relação aos objetos de interesse na imagem. É nessa etapa que temos a maior interseção entre visão computacional e processamento digital de imagens pois estamos justamente lidando com a transformação de uma imagem em outra. Do ponto de vista da engenharia, uma imagem nada mais do que um tipo de sinal, e por isto, uma outra área interessante que fornece fundamentos para a visão computacional e também para o processamento digital de imagens é a área chamada processamento digital de sinais (PDS). Muitas das técnicas utilizadas em visão computacional são derivadas de técnicas estudadas inicialmente em PDS e por isto, o termo “filtro”, que é muito usado em PDS acabou sendo também adotado, em alguns casos, na visão computacional. O termo filtro está associado à uma de suas principais aplicações que é “purificar um sinal” (retirar a sujeira ou o ruído), com por exemplo, o sinal sonoro. Técnicas de PDS podem ser utilizadas para melhorar, por exemplo, a qualidade do som reproduzido em um aparelho de som ou em um telefone. Ao interpretarmos uma imagem como um sinal, conceitos como equalização, facilmente compreendidos no universo sonoro, podem também ser aplicados a imagens e portanto, estudar a área de processamento digital de sinais pode ser um complemento bem interessante para os profissionais da área da visão computacional. Como boa parte das técnicas de PDS foram desenvolvidas inicialmente para sinais unidimensionais, como o som, elas precisam ser adaptadas para sinais bidimensionais, como é o caso das imagens que possuem uma largura e uma altura. Para “imagens em movimento” (E.g: vídeos) temos que tratar de sinais tridimensionais (além da largura e da altura de cada quadro ou *frame*, temos o tempo, que nesse caso é a terceira dimensão). A adaptação para mais de uma dimensão pode ou não ser trivial e em alguns casos, como a quantidade de informação geralmente aumenta conforme aumentamos as dimensões, podem ser necessárias otimizações especiais das técnicas para que sua aplicação em imagens seja viável.



152	144	142	154	154	154	152	152	159	178
151	149	147	157	164	163	161	176	208	233
152	155	156	164	159	169	200	234	252	255
153	153	150	157	168	207	244	253	255	255
157	159	158	165	197	238	255	254	252	252
157	162	164	188	230	255	254	249	255	255
156	162	171	213	247	255	247	250	255	252
156	170	183	222	243	250	246	247	239	225
157	175	191	212	220	227	229	222	210	206
161	176	191	192	188	193	200	194	189	200

84	84	85	83	81	78	73	69	67	62
85	89	91	87	82	76	64	54	52	58
90	97	98	89	79	67	62	56	58	65
96	102	98	82	68	54	62	66	66	69
94	95	85	70	58	63	67	65	75	68
98	96	83	68	63	74	74	72	82	78
99	94	78	68	71	76	73	69	81	87
99	93	77	73	86	84	79	75	87	102
105	103	88	89	106	107	105	103	109	125
113	118	108	111	128	127	132	131	124	140



Figura 15: Histogramas de tons de cinza

### 3.1 Histogramas

Um histograma ajuda na contagem e na comparação das contagens de diferentes “coisas”, que em visão computacional, podem ser, por exemplo, tonalidades, cores, ângulos, etc. Ele tanto pode ser utilizado como informação para outras etapas do sistema como também alterado para gerar algum tipo de transformação na imagem. No caso de um histograma de tonalidades (ou de tons de cinza) para uma imagem em preto e branco (que em visão computacional chamamos de imagem em tons de cinza) temos uma contagem da quantidade de vezes que diferentes tonalidades ocorrem na imagem. Um histograma desse tipo pode ser apresentado como um gráfico de barras (ou colunas), onde no eixo X temos os tons de cinza e no eixo Y as quantidades. Em uma imagem digital, a representação mais comum de tonalidades é através de um número que varia de 0 a 255, sendo que 0 representa o tom preto e números maiores representam tons mais claros, até chegar ao branco, que tem valor 255. Nas duas últimas linhas da Figura 11, na Seção 2.1, é fácil observar a relação entre os tons e valores que aparecem na matriz (e que correspondem justamente aos tons ou níveis de cinza). Um gráfico de barras para um histograma de tons de cinza teria portanto 256 barras (uma para cada tom). Podemos reduzir a quantidade de barras realizando uma requantização dos tons de cinza. Neste caso, ao invés de mapear cada tom em uma barra podemos mapear um grupo de tons para uma mesma barra. Por exemplo, trabalhando com 32 tons por barra, teríamos na primeira barra a contagem do total de pixels com tons variando de 0 a 31, na segunda os tons de 32 a 63, na terceira de 64 a 95 e assim por diante até a faixa que vai de 224 a 255.

Na Figura 15 temos as duas últimas imagens da Figura 11 reproduzidas, juntamente com suas matrizes e agora também, na linha de baixo, seus respectivos histogramas de tons de cinza com 8 barras (32 tons por barra) ou 8 faixas de valores, como dizemos em estatística. Comparando os histogramas das duas imagens é possível perceber como a imagem do lado esquerdo é, de forma geral mais clara (mesmo se não tivéssemos vendo as imagens), pois temos mais tons de cinza nas faixas mais altas de valores. Realizando modificações nestes histogramas

podemos, por exemplo, deixar as imagens mais claras ou escuras ou ainda distribuir melhor os tons quando temos problemas de nitidez. Na Figura 16 temos um quadro de um vídeo explicando o conceito de histogramas usando o ImageJ.

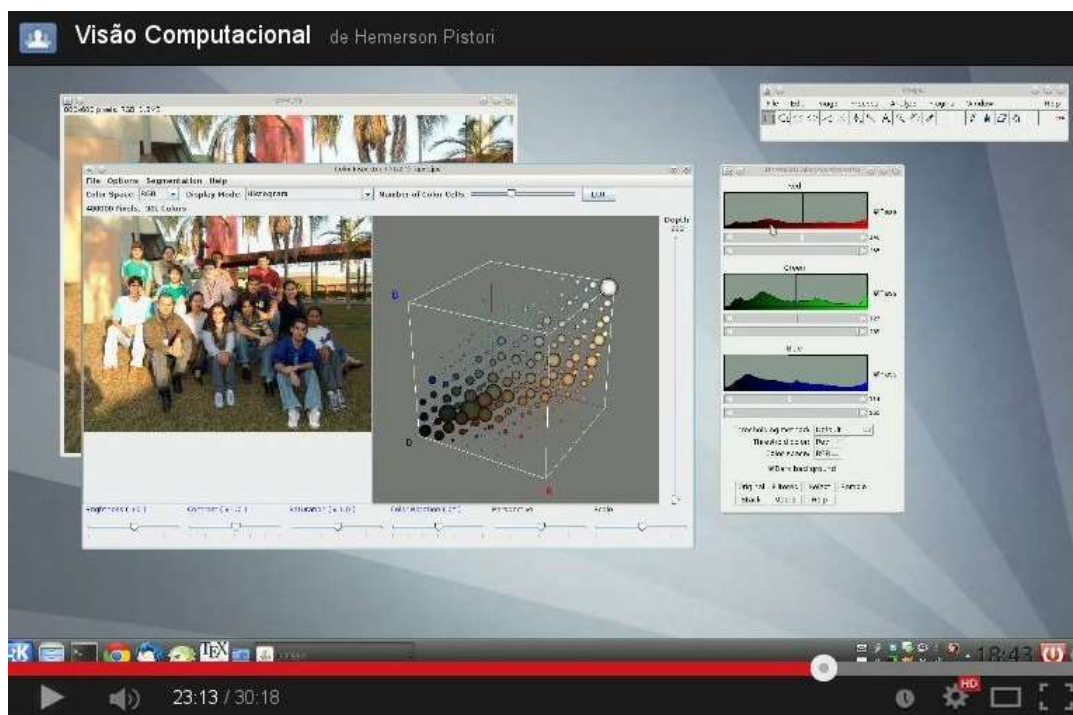


Figura 16: Vídeo explicando histogramas no ImageJ disponível em <http://youtu.be/jdiyqiHleXk>

### 3.2 Suavização

A operação ou transformação do tipo suavização também é chamada algumas vezes de alisamento pois em uma analogia com o processo de pintura de uma parede, corresponderia a lixarmos a superfície para retirar pequenas imperfeições e sujeiras deixando-a mais lisa ou suave para aplicação da tinta. As imperfeições e sujeiras corresponderiam aos ruídos que podemos encontrar em uma imagem digital e ao aplicar a suavização buscamos reduzir os ruídos. Existem diferentes técnicas de suavização de imagens e cada técnica pode também ser ajustada através de parâmetros que afetam o resultado final. De forma geral, a suavização tem como efeito um “borramento” da imagem, que acaba ficando parecida com uma imagem fora de foco e o usuário pode ter a tendência de achar que isto é ruim, pois passa a não conseguir identificar tão nitidamente os objetos. No entanto, para o sistema



computacional, isto não é necessariamente um problema, e geralmente, a imagem borrada ajuda a melhorar os resultados pois reduz o efeito dos ruídos. A Figura 17 mostra uma das telas de um vídeo que usa o ImageJ para ilustrar os efeitos de alguns tipos de suavização em uma imagem com ruídos.

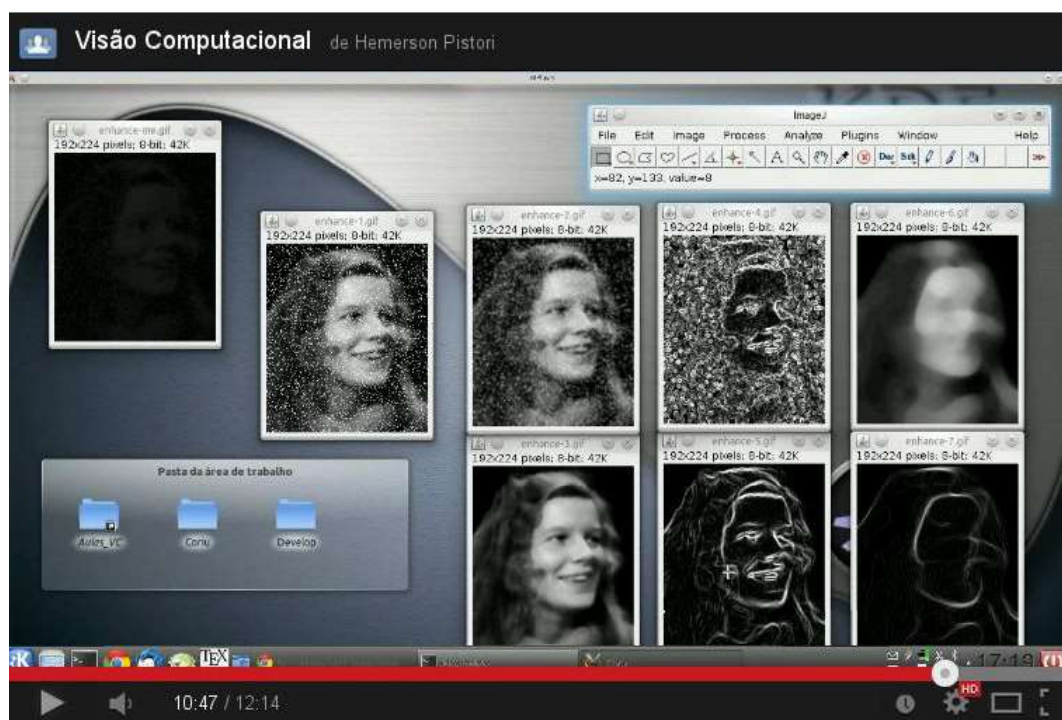


Figura 17: Utilizando o ImageJ para realizar algumas transformações na imagem, incluindo suavização e detecção de bordas. Vídeo disponível em <http://youtu.be/P1OjCHWeuew>

Entre os tipos mais comuns de técnicas de suavização estão as suavizações por média, por mediana, por média ponderada usando distribuição Gaussiana (ou simplesmente Gaussiana), por filtros passa-baixa com transformadas de Fourier e as suavizações inspiradas em modelos de difusão anisotrópica, que ao contrário das outras técnicas, aplica diferentes níveis de suavização em diferentes regiões da imagem buscando preservar melhor as regiões da imagem onde estão as bordas ou os limites dos objetos. Na Figura 18 temos o link para um vídeo que trata de uma ferramenta matemática chamada

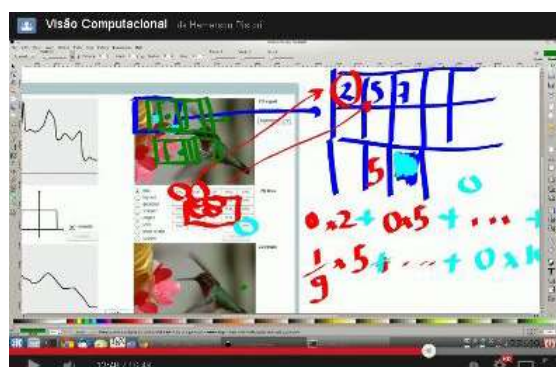


Figura 18: Vídeo sobre convolução: [http://youtu.be/WJqM\\_6HWbdI](http://youtu.be/WJqM_6HWbdI)

convolução que é muito importante para os leitores interessados em implementar programas de computadores para visão computacional. Temos também um vídeo (Figura 19) que mostra como usar o ImageJ para realizar suavização com transformadas de Fourier.

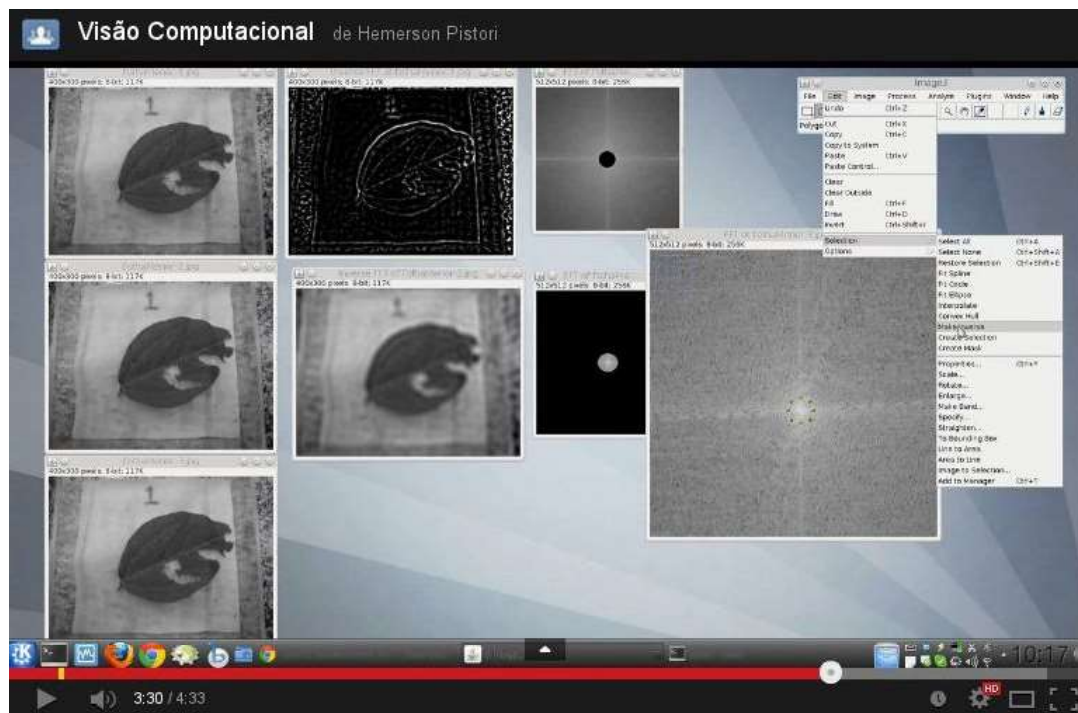


Figura 19: Suavização com transformadas de Fourier:  
<http://youtu.be/6a9tersgwgg>



*Figura 20: Imagem de rapaz com violão (esquerda) e o resultado da aplicação de um detector de borda (direita)*

### **3.3 Realce e Detecção de bordas**

O realce ou aguçamento é de certa forma uma operação oposta ou inversa à suavização. Ela aumenta a nitidez dos contornos dos objetos presentes na imagem e pode também aumentar o efeito de determinados ruídos, por isso, é comum que os filtros de realce sejam aplicados depois de filtros de suavização. No caso extremo, a operação de aguçamento manterá na imagem apenas as bordas ou contornos dos objetos e portanto, existe uma relação muito grande entre a operação de realce e a operação chamada de detecção de bordas, que tem como objetivo justamente gerar uma nova imagem apenas com os contornos dos objetos. Assim como na suavização, existem diversas técnicas para realizar o realce e a detecção de bordas. Entre as técnicas de detecção de bordas podemos citar os detectores de Sobel, de Roberts, de Prewitt e de Canny. A convolução também é uma ferramenta importante para implementar técnicas de aguçamento e detectores de borda e os vídeos das Figuras 17, 18, 19 tratam também desse tema. A Figura 20 mostra o resultado da aplicação de um filtro de detecção de bordas. Note que apenas observando a imagem da direita (imagem de bordas) é possível obter muita informação sobre a imagem original. Ou seja, em alguns casos, as cores ou mesmo as tonalidades de cinza não são necessárias para se resolver um problema e

podemos trabalhar apenas com as imagens das bordas ou contornos, o que pode representar, além de uma menor quantidade de informação a ser tratada, uma redução na quantidade de ruídos. É claro que existem outros casos em que cores e tonalidades são essenciais, como por exemplo, em um módulo para identificar sinais de trânsito em um sistema de visão computacional acoplados a um veículo que dirige sozinho. Como em quase todas as etapas de um sistema de visão computacional, as técnicas apropriadas terão que ser definidas caso a caso, muitas vezes através de experimentos comparativos com imagens reais das cenas ou objetos que serão analisados pelo sistema.

### 3.4 Morfologia Matemática

A morfologia matemática é uma abordagem para tratar problemas de visão computacional fundamentada na teoria dos conjuntos. Ela tem como uma de suas principais características permitir que as soluções para problemas como suavização, detecção de bordas e segmentação, entre outros, sejam representadas de forma concisa através de expressões algébricas onde operadores mais complexos vão sendo construídos através de operadores mais simples. Para compreender o funcionamento dos principais operadores da morfologia matemática recomendamos o vídeo referenciado na Figura 21 que mostra alguns exemplos do funcionamento dos operadores de dilatação, erosão, abertura e fechamento.

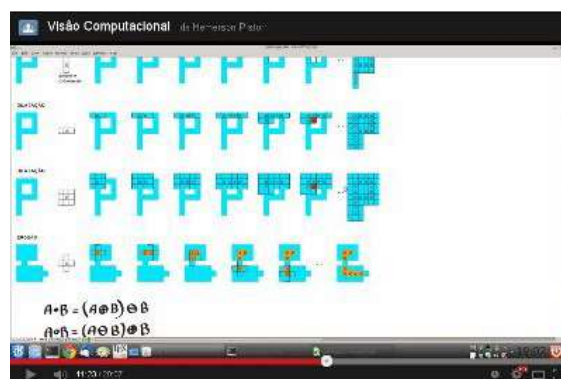


Figura 21: Vídeo sobre morfologia matemática disponível em <http://youtu.be/E8cqrkK4L> M