



Universidade Federal de Pernambuco
Centro de Informática
Pós-Graduação Ciência da Computação

Data: 18/05/2021
Curso: Processamento de Imagens

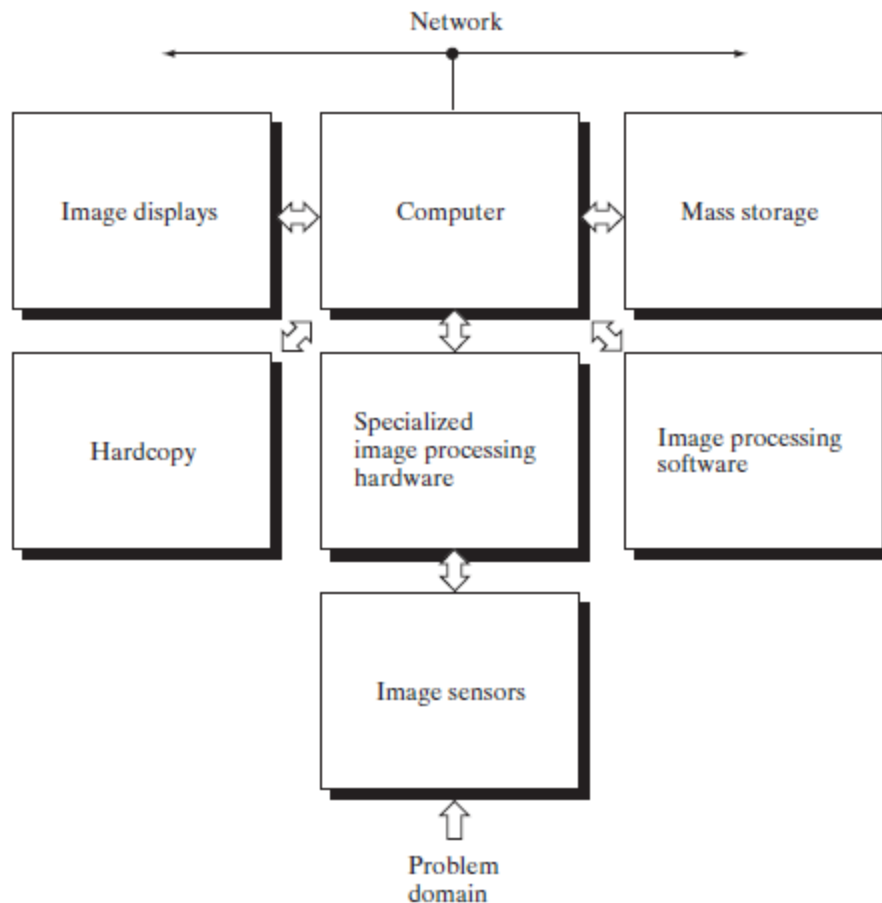
Estudante: Diógenes Wallis de França Silva

1. Fundamentos da Imagem:

a). Descreva os componentes de um sistema de processamento de imagens explicando a funcionalidade de cada um deles e dando exemplos práticos de problemas e soluções tratados em cada uma das etapas. Dê um exemplo prático de um sistema deixando claro qual é a funcionalidade desempenhada por cada uma delas.

R.:

A figura abaixo mostra os componentes básicos de um sistema de “general-purpose” para processamento digital de imagem.



Seguem os detalhes acerca de cada um dos componentes:

Image sensors: Basicamente dois elementos são necessários para obter imagens digitais. O primeiro trata-se de um dispositivo sensível à energia radiada pelo objeto que desejamos obter a imagem. Já o segundo, chamado de digitalizador, é responsável por fazer a conversão dos dados obtidos pelo sensor em dados digitais. Por exemplo, numa câmera digital os sensores geram uma saída elétrica proporcional a intensidade da luz, em que o digitalizador converte essa saída para uma informação digital.

Specialized image processing hardware: Trata-se basicamente do digitalizador mencionado anteriormente, mais um hardware capaz de efetuar outras operações, como uma ULA (unidade lógica aritmética), que executa operações lógicas e aritméticas em paralelo em imagens inteiras. Um exemplo de uso da ULA é no processo de *image averaging* (trata-se de somar as intensidades dos pixels de um conjunto de N imagens e depois tirar a média dividindo por N) tão rápido quanto as imagens são digitalizadas, para fins de reduzir o ruído.

Computer: Em um sistema de processamento de imagem, qualquer computador de *general-purpose* pode ser utilizado, podendo ser desde um PC até um supercomputador.

Quando há aplicações dedicadas, podem existir computadores customizados que são usados para obter um certo nível de performance.

Software: No contexto de processamento de imagens, o software consiste em módulos especializados que executam tarefas específicas. Um *package* bem feito também inclui a possibilidade do usuário escrever códigos que façam uso dos módulos especializados.

Mass storage: A capacidade de armazenamento é fundamental em aplicações de processamento de imagem. Por exemplo, uma imagem de tamanho 1024 x 1024 pixels, sendo cada pixel com uma intensidade de 8-bits (0 a 255), requer ao menos 1 megabyte de armazenamento, caso a imagem não seja comprimida. Quando trata-se de milhares de imagens, o armazenamento pode ser desafiador. O armazenamento digital pode ser dividido em 3 categorias principais: *short term storage*, *online storage* e *archival storage*.

Image displays: Atualmente os principais displays de imagem são os monitores de TV coloridos. Os monitores funcionam através das informações recebidas pela placa de *image display* do computador.

Hardcopy: Dispositivos de *hardcopy* usados para gravação de imagem incluem: impressoras a laser, câmeras de filme, dispositivos sensíveis ao calor, e até discos como CD-ROM. Vale mencionar que o filme providencia a maior resolução dentre as opções mencionadas.

Networking: Atualmente é praticamente uma função padrão para os computadores em uso. Por conta da grande quantidade de dados em aplicações de processamento de imagem, uma importante consideração a ser feita na transmissão das imagens é a largura de banda. Felizmente, esta situação está mudando, por conta de tecnologias como a fibra óptica.

Como exemplo, pode-se citar um **celular** (que seria o equivalente ao computador mencionado anteriormente) como o iPhone 12 PRO. Este celular possui uma câmera, que por sua vez contém **sensores** que vão gerar as saídas elétricas que serão digitalizadas por um digitalizador. Essas informações serão **armazenadas** na memória do próprio celular ou de algum **cartão de memória** (*hardcopy*) externo. A imagem capturada ainda pode sofrer algum tipo de edição em algum **software** do próprio celular ou de algum aplicativo. Do ponto de vista de mais baixo nível, também vale mencionar o **hardware** presente capaz de fazer operações lógicas e aritméticas na imagem. Por último, a imagem pode ser visualizada no **display** do celular. As imagens também podem ser enviadas via **internet** através de uma mensagem no WhatsApp, por exemplo.

b) Considere duas imagens de 8 bits cujos níveis de cinza cobrem todo o intervalo de 0 a 255:

i. Discuta o efeito limitador de subtrair repetidamente a imagem (2) da imagem (1). Suponha que o resultado também seja representado em 8 bits.

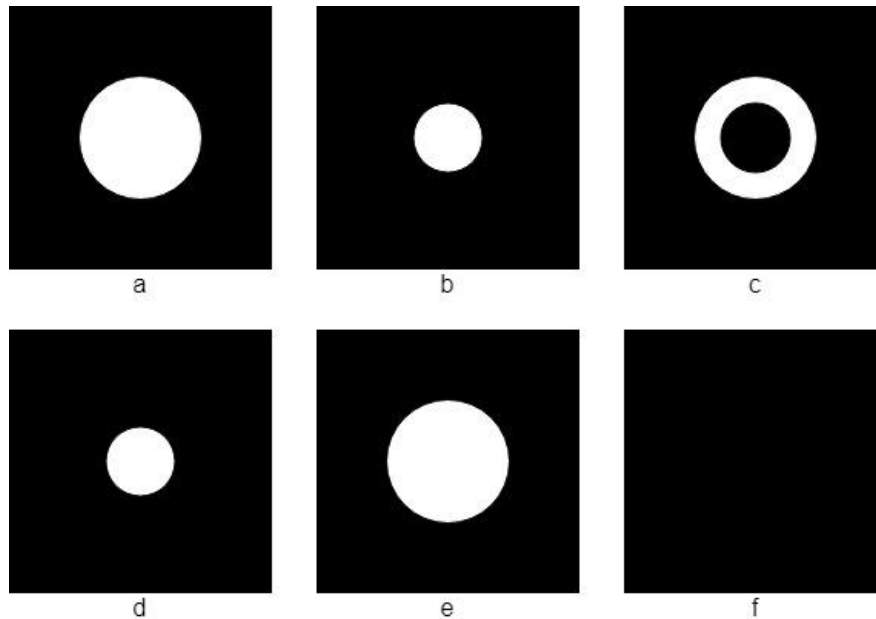
ii. Inverter a ordem das imagens levaria a um resultado diferente?

R.:

i. Sendo uma imagem constituída por $M \times N$ pixels, M sendo o número de linhas e N o número de colunas, a cada pixel é atribuído um número de 0 a 255. Vamos considerar imagens de mesmo tamanho apenas para exemplificar. Fixando a imagem (1) e realizando uma subtração pela imagem (2), isto é imagem (1) - imagem (2), temos o seguinte resultado: os pixels de valor 0 na imagem (2) não vão alterar na imagem (1) os pixels nas mesmas posições e os pixels da imagem (2) de valor maior que 0 vão reduzir os valores dos pixels de mesma posição na imagem (1). Por exemplo, se o pixel da imagem (2) na posição (10,10) tiver intensidade 0 e o pixel (10,10) na imagem (1) tiver intensidade 100, após a subtração o valor em (10,10) da imagem (1) permanece 100. Porém, se na imagem (2) o pixel (10,10) fosse 40, após a subtração, o pixel (10,10) na imagem (1) seria $100 - 40$, isto é, sua intensidade seria 60. Repetindo essa subtração o que vai acontecer é que onde os pixels na imagem (2) tem valor positivo, os pixels da mesma posição na imagem (1) vão ser reduzidos e podem até se tornar nulos. Enquanto que onde os pixels de (2) foram nulos, em nada vai mudar os pixels equivalentes em (1). Logo, a imagem resultante da subtração é a imagem (1) com os valores dos pixels reduzidos onde na imagem (2) existem pixels de intensidade maior que zero.

ii. Sim, inverter altera o resultado. Se houver essa inversão, a imagem resultante da subtração é a imagem (2) com os valores dos pixels reduzidos onde na imagem (1) existem pixels de intensidade maior que zero.

A imagem abaixo ilustra o que foi dito. Note que há a subtração da imagem (a) pela imagem (b), resultando em (c). Houve também o inverso: imagem (d) - imagem (e), resultando na imagem (f).



c) Como é obtido o zoom digital? Explique tanto para magnificação como redução da imagem.

R.:

O zoom é realizado através do processo de interpolação. A interpolação é o processo de utilizar dados conhecidos para estimar valores em posições desconhecidas, aqui vamos tratar da interpolação pelos vizinhos mais próximos. Vamos supor que temos uma imagem de tamanho 100x100 e queremos aumentá-la para 200 x 200. Primeiro, vamos dobrar o número de colunas, neste caso é necessário manter o mesmo espaçamento entre os pixels da imagem original. Neste caso vão aparecer pixels sem valor, cada um desses pixels vai receber o valor do pixel adjacente. O mesmo se aplica para as linhas, dessa forma obtém-se o zoom digital para magnificação. Além da técnica dos vizinhos mais próximos, há também a interpolação bilinear.

O processo de redução é parecido com o processo de zoom. Porém, ao invés de adicionar novos pixels e atribuir valores a estes, agora o processo consiste em deletar linhas e colunas. Para reduzir a imagem de 100 x 100 para 50 x 50, por exemplo, seria preciso deletar metade das linhas e das colunas.

2. Crie uma imagem de tamanho 5X5, com 3-bits de tons de cinza utilizando pelo menos uma vez cada um dos valores dos tons de cinza:

- a. Determine as frequências dos tons de cinza da imagem.**
- b. Desenhe o histograma destas frequências.**
- c. Aplique a equalização do histograma.**

- d. Desenhe o histograma resultante após a equalização.
e. Explique qual a utilidade da equalização do histograma.
f. O que ocorre se utilizamos a equalização do histograma na imagem diversas vezes?
Por quê?
g. Descreva outro método para o ajuste de tons de cinza em uma imagem.

R.:

A imagem criada é da seguinte forma:

5	1	2	7	1
1	2	5	0	7
3	1	6	7	2
4	6	0	3	2
6	3	4	5	1

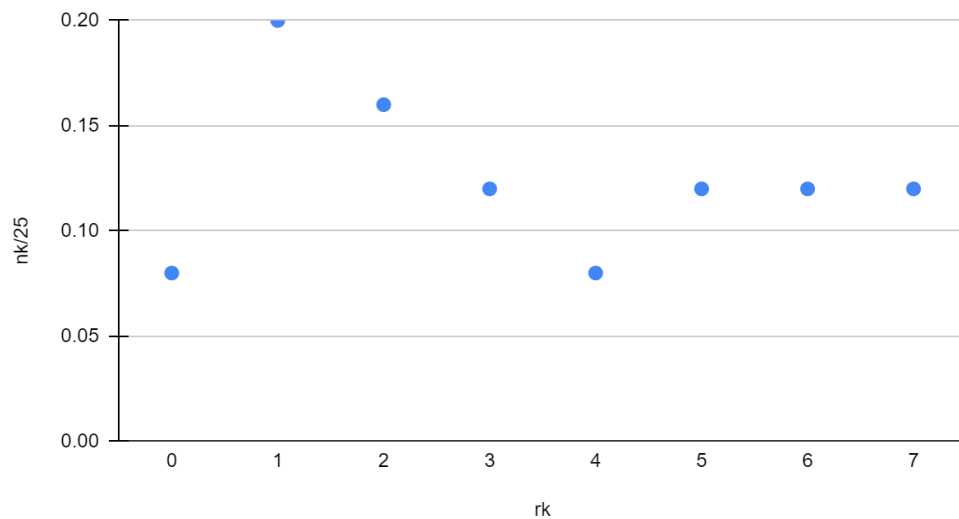
a.

Seja r_k o nível de cinza e n_k a quantidade de pixels com essa intensidade:

r_k	n_k	$n_k/25$
0	2	0.08
1	5	0.2
2	4	0.16
3	3	0.12
4	2	0.08
5	3	0.12
6	3	0.12
7	3	0.12

b.

nk/25 vs. rk

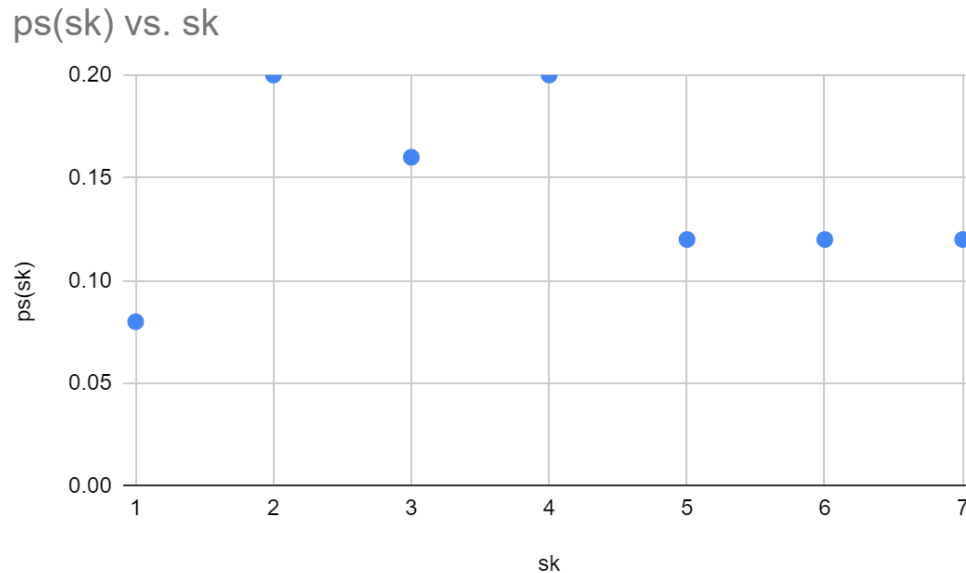


c.

Aplicando a equação 3.3-8 do livro, obtém-se s_k (o s'_k é a soma acumulativa que multiplicada pelo número de níveis de cinza vai resultar em s_k). Essa transformação de r_k em s_k é chamada de equalização do histograma. São obtidos valores fracionários que são arredondados para o inteiro mais próximo. A equalização consiste neste mapeamento de r_k em s_k . Caso os níveis em r_k sejam mapeados para um mesmo valor de s_k , os níveis de intensidade se somam. Isso acontece em $s_k = 4$, o número de pixels atribuídos são a soma das quantidades com $r_k = 3$ e $r_k = 4$.

rk	s'k	sk	ps(sk)
0	0.08	1	0.08
1	0.28	2	0.2
2	0.44	3	0.16
3	0.56	4	0.2
4	0.64	4	0.2
5	0.76	5	0.12
6	0.88	6	0.12
7	1	7	0.12

d.



e.

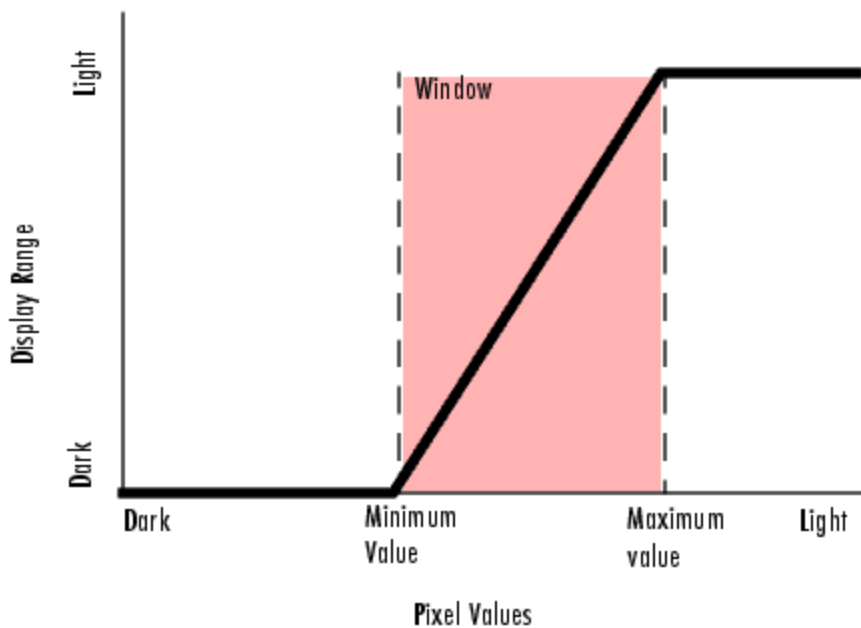
A equalização do histograma é um método para processar imagens com o intuito de ajustar o contraste de uma imagem através da modificação da distribuição de intensidade no histograma. O objetivo desta técnica é gerar uma tendência linear na função de distribuição acumulativa da imagem referente às probabilidades das intensidades dos pixels.

f.

Não há mudanças, isto é, aplicar o processo de equalização a um histograma já equalizado não gera modificação. Isto se dá pelo fato da equalização ser *idempotent*, mesmo aplicando a equalização várias vezes o resultado é sempre o mesmo. Do ponto de vista matemático, o mapeamento de r_k em s_k vai gerar sempre o mesmo resultado.

g.

Uma das técnicas utilizadas para o ajuste de tons de cinza é o *contrast stretching*. Trata-se de um processo que expande o intervalo de valores de níveis de intensidade em uma imagem, no caso os níveis de intensidade da imagem serão expandidos para o intervalo total disponível pelo dispositivo de display. A imagem abaixo ilustra bem o que foi dito:



Note que o intervalo de Minimum Value a Maximum value é razoavelmente menor do que de Dark a Light. A figura 3.10 do livro também ilustra bem o resultado do *contrast stretching*.

3) Transformações de Intensidade

- a. Qual seria o efeito geral no histograma de uma imagem de zerar todos os planos de bits de baixa ordem?
- b. Qual seria o efeito sobre o histograma se zerássemos os planos de bits de alta ordem?

R.:

a.

Cada plano de bit é uma imagem binária. Para uma imagem de 8 bits, por exemplo, para o 8º plano de bit, tem-se que os níveis de intensidade são mapeados para 1 apenas se o oitavo bit for 1, do contrário será mapeado para zero. O mesmo se aplica aos outros planos, para o 3º plano de bit, por exemplo, onde o terceiro bit estiver em nível alto será considerado 1 e o restante 0. Basicamente, para que haja os 256 níveis de cinza possíveis é preciso dos 8 planos de bit, logo, zerar os planos resulta em menos níveis possíveis de intensidade. Sendo assim, o efeito de zerar os planos de baixa ordem vai fazer com que alguns componentes do histograma sejam removidos, enquanto o número de pixels de outros componentes devem aumentar, já que o número de pixels não muda. Do ponto de vista visual, o impacto de remover os planos de bits de baixa ordem não é tão impactante quanto remover os de ordem alta.

b.

Zerar os planos de bits de alta ordem vai fazer com que haja uma perda significativa de detalhes na imagem. A figura 3.14 do livro mostra uma imagem e seus 8 planos de bits, nesta imagem é possível perceber que boa parte do conteúdo visual da imagem está nos planos de bits de ordem mais alta. Do ponto de vista do histograma, haverá redução no número de componentes, assim como no caso anterior. Porém, considerando o plano de 8 bits, por exemplo, a análise é mais simples, já que nenhum nível com o MSB em nível alto será possível na imagem. Consequentemente, todas as componentes de 128 a 255 serão removidas do histograma. Isto fará com que outros componentes aumentem de valor, já que o número de pixels ainda se mantém, conforme a letra a).

4. Em uma dada aplicação, um filtro de média é aplicado a imagens de entrada para reduzir o ruído, e um filtro laplaciano é aplicado para realçar pequenos detalhes. O resultado seria o mesmo se a ordem dessas operações fosse invertida? Demonstre a sua resposta.

R.:

Sim, pois trata-se de duas operações lineares.

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)$$

A expressão acima representa o processo de filtragem espacial linear para um dado filtro de coeficientes w . Esta operação é realizada no filtro de média.

Já para o Laplaciano é realizado o cálculo abaixo:

$$g(x, y) = f(x, y) + c[\nabla^2 f(x, y)]$$

O laplaciano acima pode ser implementado utilizando também uma máscara de filtragem, de modo que é definido um filtro e seus coeficientes para em seguida realizar a filtragem espacial linear, assim como no caso do filtro de média.

Consequentemente, não faz diferença inverter a ordem das posições. Note que se for aplicado um filtro A e depois um filtro B, não faz diferença se a ordem for invertida. Após as operações, independentemente de ordem, os pixels da imagem são multiplicados pelos coeficientes dos dois filtros.

Abaixo está a demonstração matemática. Os índices nos somatórios foram suprimidos para facilitar a escrita, mas os considere como na definição acima de filtragem espacial linear.

$$g(x, y) = \sum \sum w(n, t) f(x+n, y+t) \quad (1) \quad (\text{Filtragem espacial linear})$$

$$g(x, y) = f(x, y) + c [\nabla^2 f(x, y)] \quad (\text{Laplaciano})$$

Seu $\nabla^2 f(x, y)$ implementado por uma máscara como:

0	1	0
1	-4	1
0	1	0

Então o Laplaciano pode ser calculado como:

$$g(x, y) = f(x, y) + c \sum \sum w(n, t) f(x+n, y+t) \quad (2)$$

Aplicando o filtro de média, temos $f(x, y)$ a imagem de entrada:

$$g(x, y) = \sum \sum w(n, t) f(x+n, y+t) \quad (3)$$

Agora consideramos $g(x, y)$ a entrada para calcular o Laplaciano:

$$h(x, y) = g(x, y) + c \sum \sum w(n, t) g(x+n, y+t) \quad (4)$$

Substituindo $g(x, y)$ por (3):

$$h(x, y) = \sum \sum w(n, t) f(x+n, y+t) + c \sum \sum w(n, t) \left(\sum \sum w(m, s) f(x+m, y+s) \right) \quad (5)$$

Vamos inserir, primeiro a Duplicação:

$$g(x, y) = f(x, y) + c \sum \sum w(n, t) f(x+n, y+t) \quad (6)$$

Agora o filtro de média:

$$h(x, y) = \sum \sum w(n, t) g(x+n, y+t)$$

Substituímos $g(x, y)$ por (6):

$$h(x, y) = \sum \sum w(n, t) \left(f(x+n, y+t) + c \sum \sum w(n, t) f(x+2n, y+2t) \right)$$

$$h(x, y) = \sum \sum w(n, t) f(x+n, y+t) + c \sum \sum w(n, t) \left(\sum \sum w(n, t) f(x+2n, y+2t) \right)$$

Note que a expressão acima é igual a (5) que é o resultado do filtro de média e da Duplicação em sequência. Portanto, a ordem dos processos não altera o resultado.

5. Uma câmera de TV do tipo CCD é utilizada para realizar um estudo de longo prazo observando a mesma área 24 horas por dia, durante 30 dias. Imagens digitais são capturadas e transmitidas a uma central a cada 5 minutos. A iluminação da cena varia entre a luz diurna natural e iluminação artificial. Em momento algum a cena fica sem iluminação, de forma que é sempre possível obter uma imagem. Como a variação da iluminação é tal que se mantém sempre na faixa linear de operação da câmera, decide-se não empregar nenhum mecanismo de compensação na própria câmera. Em vez disso, foi decidido utilizar técnicas de processamento de imagens para o pós-processamento, normalizando as imagens ao equivalente de uma iluminação constante. Proponha um método para fazer isso. Você pode utilizar qualquer método que quiser, mas explique claramente todas as considerações feitas para chegar a seu objetivo.

R.:

Trata-se de um ambiente que recebe luz natural e iluminação artificial, porém não se sabe se há ou não objetos ou pessoas se movendo na imagem. Algumas considerações são feitas: será desconsiderada a iluminação natural (para que não haja a necessidade de efetuar várias normalizações ao longo do dia) e nenhum objeto está se movendo na imagem (para que haja

um ambiente o mais “constante” possível). Também vale ressaltar que a iluminação opera na faixa linear da câmera, ou seja, não há problemas de saturação (quando a perda de informação por ultrapassar o limite superior da faixa de operação) e nem problemas em ultrapassar o limite inferior da faixa de operação.

Considerando um ambiente sem objetos se movendo e com apenas iluminação artificial, verifica-se o valor mínimo e máximo (v_{\min} e v_{\max}) das intensidades dos pixels da imagem. De posse desses valores, basta fazer uma simples transformação linear para realizar o mapeamento dos valores mínimo e máximo de uma dada imagem de entrada para v_{\min} e v_{\max} .

Seja uma transformação linear $g(x,y) = a \cdot f(x,y) + b$, faz-se necessário determinar **a** e **b**.

Para calcular **a**, basta fazer:

$$g(x,y) = v_{\min} \text{ quando } f(x,y) = f_{\min} \text{ e } g(x,y) = v_{\max} \text{ quando } f(x,y) = f_{\max}$$

$$v_{\min} = a \cdot f_{\min} + b$$

$$v_{\max} = a \cdot f_{\max} + b$$

Resolvendo o sistema de equações acima:

$$a = (v_{\min} - v_{\max}) / (f_{\min} - f_{\max})$$

$$b = (f_{\min} \cdot v_{\max} - v_{\min} \cdot f_{\max}) / (f_{\min} - f_{\max})$$

Uma vez que **a** e **b** foram determinados, seja uma imagem de entrada $f(x,y)$, basta aplicar a transformação linear acima e mapear os mínimos e máximos de $f(x,y)$ para v_{\min} e v_{\max} . Dessa forma, faz-se a normalização da iluminação.

6) Construa um filtro 3 x 3 para realizar a máscara de nitidez passando uma única vez por uma imagem. Considere que a imagem média seja obtida utilizando o filtro da figura abaixo.

$\frac{1}{9} \times$	a		
	1	1	1
	1	1	1
	1	1	1

R.:

Seja $g(x, y) = f(x, y) + k \cdot g_{\text{mask}}(x, y)$ em que $k=1$ implica que $g(x, y)$ é o resultado do unsharp masking, tem-se:

$$g(x, y) = f(x, y) + g_{\text{mask}}(x, y), \text{ em que}$$

$$g_{\text{mask}}(x, y) = f(x, y) - \bar{f}(x, y), \text{ sendo } \bar{f}(x, y) \text{ a imagem média}$$

Substituindo $g_{\text{mask}}(x, y)$ em $g(x, y)$:

$$g(x, y) = f(x, y) + f(x, y) - \bar{f}(x, y)$$

$$g(x, y) = 2f(x, y) - \bar{f}(x, y)$$

$$g(x, y) = 2f(x, y) - \sum_{n=-a}^a \sum_{t=-b}^b w(n, t) f(x+n, y+t)$$

Pela expressão acima, percebe-se que quando $n=0$ e $t=0$, este elemento da soma (centro do filtro) será acrescido de $2f(x, y)$, enquanto os outros termos ficam inalterados. Sendo assim, para o elemento central:

$$\begin{aligned} 2f(x, y) - w(0, 0) \overset{\rightarrow \frac{1}{9}}{f}(x, y) &= 2f(x, y) - \frac{f(x, y)}{9} \\ &= \frac{17 f(x, y)}{9} \end{aligned}$$

Sendo o filtro:

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Após a modificação por conta de $2f(x,y)$ e também do sinal negativo por conta do unsharp masking:

$$-\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & -17 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Note que o sinal negativo em -17 se dá porque após a multiplicação por $-\frac{1}{9}$ o resultado é positivo, conforme os cálculos que chegaram em $\frac{17}{9} f(x,y)$.

Note que basta aplicar o filtro acima para obter o processo de unsharp masking.