

Universidade de Brasília – Faculdade de Tecnologia

364622 - Visão Computacional

Professor José Maurício S. T. Motta

1º Trabalho de Avaliação

José Gabriel Hermes Cavalcanti – 18/0154761

Brasília, 18 de Setembro de 2018

Introdução

Este trabalho consiste de atividades de aplicação e consolidação de conhecimentos ensinados durante o segundo semestre letivo de 2018 na disciplina Visão Computacional, consistindo de conceitos de propriedades de imagens digitais como de armazenamento e discretização espacial, ruídos em imagens seus tipos e de extração ou presença características de imagem, tal como a presença de bordas ou cantos e seus significados.

Está dividido em duas partes, a primeira (página 1) se tratando de relatório relativo a experimentos práticos envolvendo a construção e execução de rotinas em linguagem de scripts para cálculo numérico MATLAB, enquanto a segunda parte (página 12) contém respostas à perguntas propostas no próprio roteiro deste trabalho.

Parte prática

Item a - As imagens foram adquiridas pela câmera traseira de um smartphone Xiaomi Mi 5s, composta por um sensor Sony IMX378 e abertura f/2.0. Foi montado em um tripé com garra de afixação de celulares, de modo a evitar variações de posicionamento, e cujas fotos foram tiradas por meio de disparador remoto bluetooth, de forma a não gerar vibrações na montagem pelo disparo manual. A iluminação da cena foi realizada por meio de duas lâmpadas LED no ambiente.

A partir da coleção de imagens obtidas, foi calculada a imagem média, seguindo a seguinte fórmula:

$$\bar{E} = \frac{1}{n} \sum_{k=1}^n E_k$$

Em posse dos valores de média, para todos os pixels da imagem, calculamos o desvio padrão. Para cada valor de pixel da imagem, temos:

$$\sigma(i, j) = \frac{1}{n-1} \sum_{k=1}^n \sqrt{\left(\bar{E}(i, j) - E_k(i, j)\right)^2}$$



Figura 1 - Imagem média

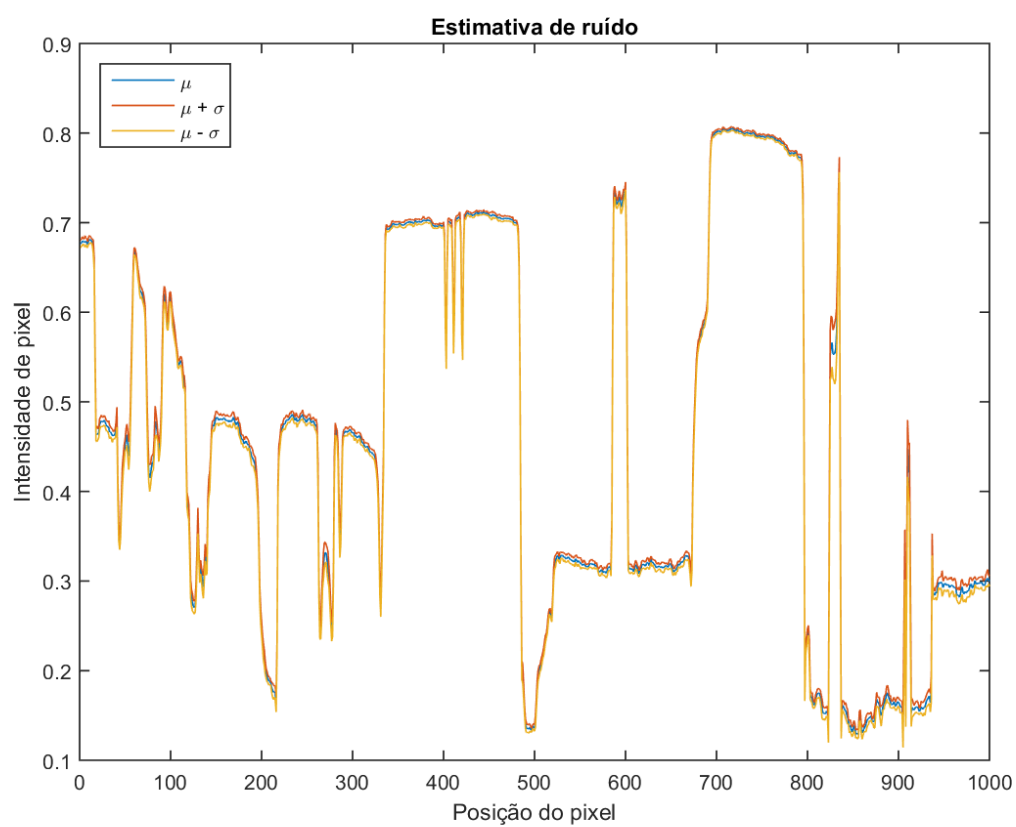


Figura 2 - Estimativa de ruído na linha 350 do conjunto de imagens

A partir da matriz de desvios padrões calculada, é obtemos o valor do desvio padrão médio da imagem. Expressando este valor como porcentagem, temos que a variação produzida por ruído segue a saída por console do programa, dada por:

Valor médio do Desvio Padrão: 0.601253%

Podemos observar visualmente a variação de níveis de intensidade devido ao ruído por meio da Figura 2. Nesta, são exibidos os valores de intensidade da linha horizontal 350 da imagem média (de um total de 750), acompanhados dos valores de intensidade somados e subtraídos ao desvio padrão dos pixels da das imagens do conjunto, para a mesma linha. Podemos então observar que de fato a variação dos valores é pequena, atestando o baixo nível de ruído das imagens adquiridas pela câmera.

Para obter a autocovariância da imagem, temos que, para matrizes indexadas a partir de 1:

$$c = \frac{1}{N^2}, N_{i'} = N - i', N_{j'} = N - j$$

$$C_{EE}(i', j') = c. \sum_{i=1}^{N'_i} \sum_{j=1}^{N'_j} (E(i, j) - \overline{E(l, j)})(E(i + i', j + j') - \overline{E(l + l', j + j')})$$

Por aspectos de desempenho computacional, dado que a expressão acima determina que cada elemento da matriz C_{EE} é dado pela soma dos elementos da multiplicação termo a termo de submatrizes de tamanho igual de uma matriz de diferenciação, temos que a expressão acima pode equivalentemente descrita por:

$$D = E - \bar{E}; \quad C_{EE}(i', j') = c. \sum_{i=1}^{N'_i} \sum_{j=1}^{N'_j} D(i, j)D(i + i', j + j')$$

E então, equivalentemente, por:

$$C_{EE}(i', j') = c. \sum D(1:N_{i'}, 1:N_{j'})D(1 + i':N_{i'} + i', 1 + j':N_{j'} + j')$$

Mesmo com estas otimizações, o cálculo desta matriz tomou 743.399646 segundos, medidos utilizando as funções tic e toc do MATLAB.

Uma representação visual da matriz de autocovariância pode ser vista na Figura 3.

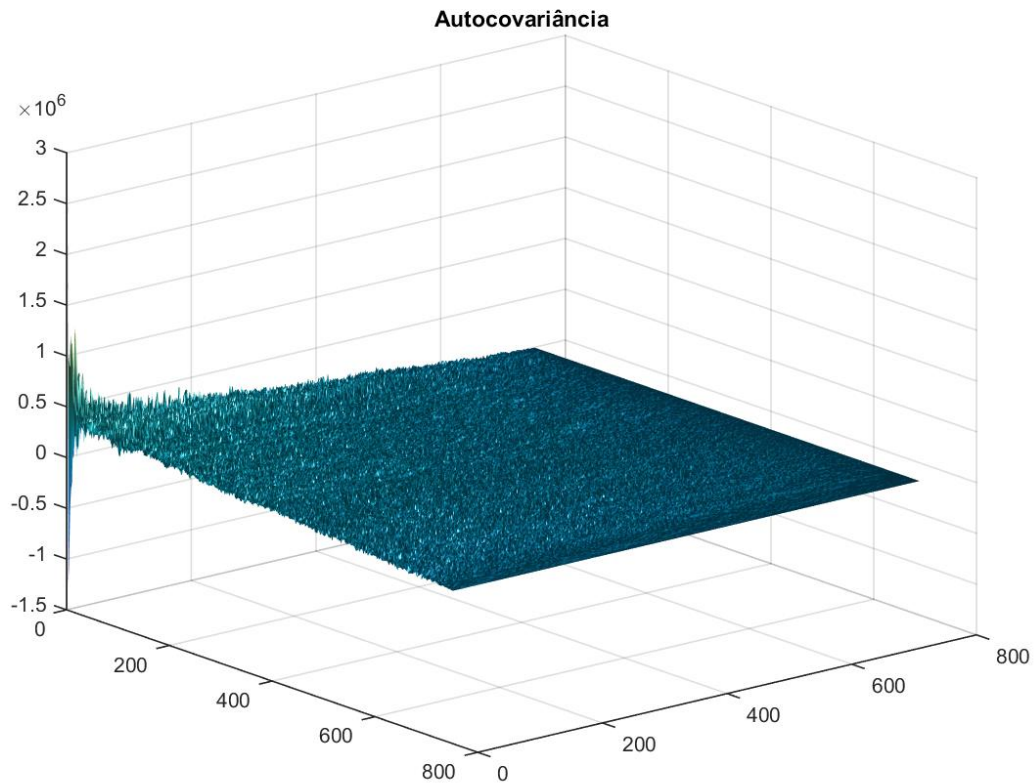


Figura 3 - Representação Visual da Matriz de Autocovariância

Item b - Ao considerar o ruído em imagens digitais como um termo aditivo aplicado sobre uma imagem não ruidosa, foram construídas rotinas para a geração de imagens com ruído gaussiano, e do tipo sal-e-pimenta. No primeiro caso, o ruído é expresso como valores de intensidades aleatórias, que seguem a distribuição normal, que são adicionados aos valores de intensidade originais da imagem. No caso do ruído sal-e-pimenta, este é construído por meio de uma função probabilidade de saturar os valores de intensidade dos pixels, para um limitante superior ou inferior de intensidade, gerando um efeito similar à dispersão de sal e pimenta-preta sobre a imagem original.

Imagens degradadas com o ruído artificial podem ser vistas nas figuras 4a, 4b e 4c.



Figura 4 – (A) Imagem Original, (B) Imagem com ruído gaussiano, (C) Imagem com ruído sal-e-pimenta.

Também foram construídos filtros passa baixas, que em geral atenuam os efeitos dos ruídos para tarefa posterior de processamento de imagem. Foram implementados filtros do tipo gaussiano e de média. O primeiro realiza uma média ponderada sobre os valores de intensidade dos pixels de uma imagem, considerando mais fortemente o elemento central. O filtro de média, em contraste realiza uma média simples. Foram aplicados os filtros para as mesmas imagens ruidosas apresentadas anteriormente, a 4ª e 4c, resultando nas figuras 5a, 5b e 5c para dois tamanhos de filtros gaussianos para ruído gaussiano, 6a, 6b e 6c para filtros de média e ruído gaussiano, 7a, 7b e 7c para filtros gaussianos e ruído sal e pimenta, e 8a, 8b e 8c para filtros de média e ruído sal e pimenta.

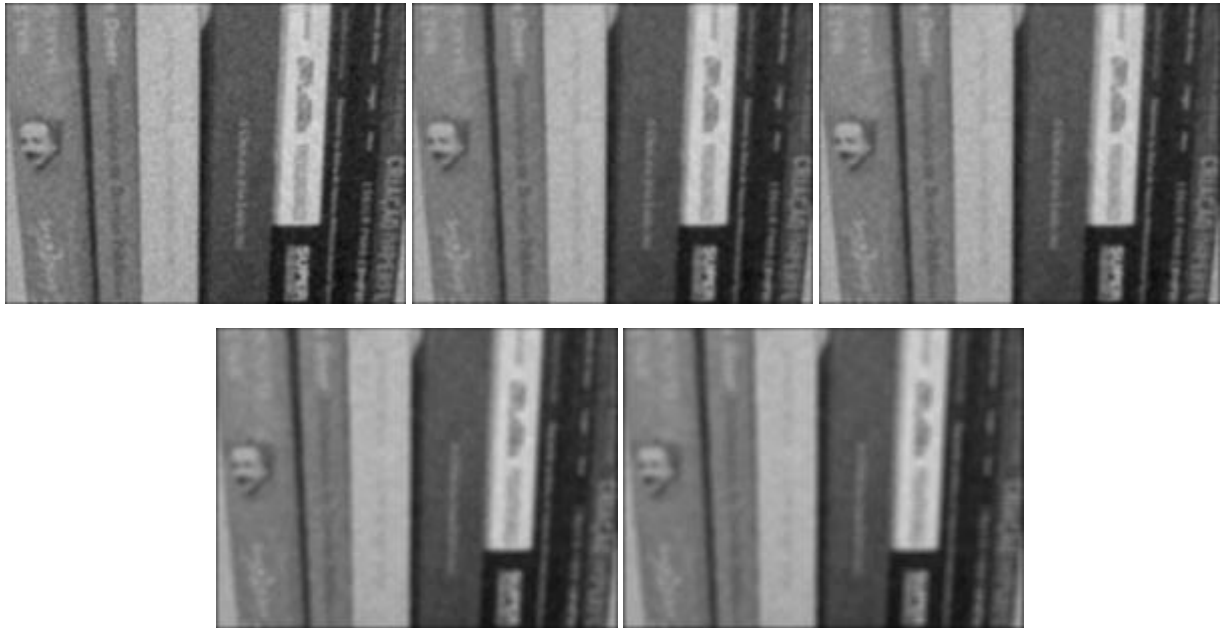
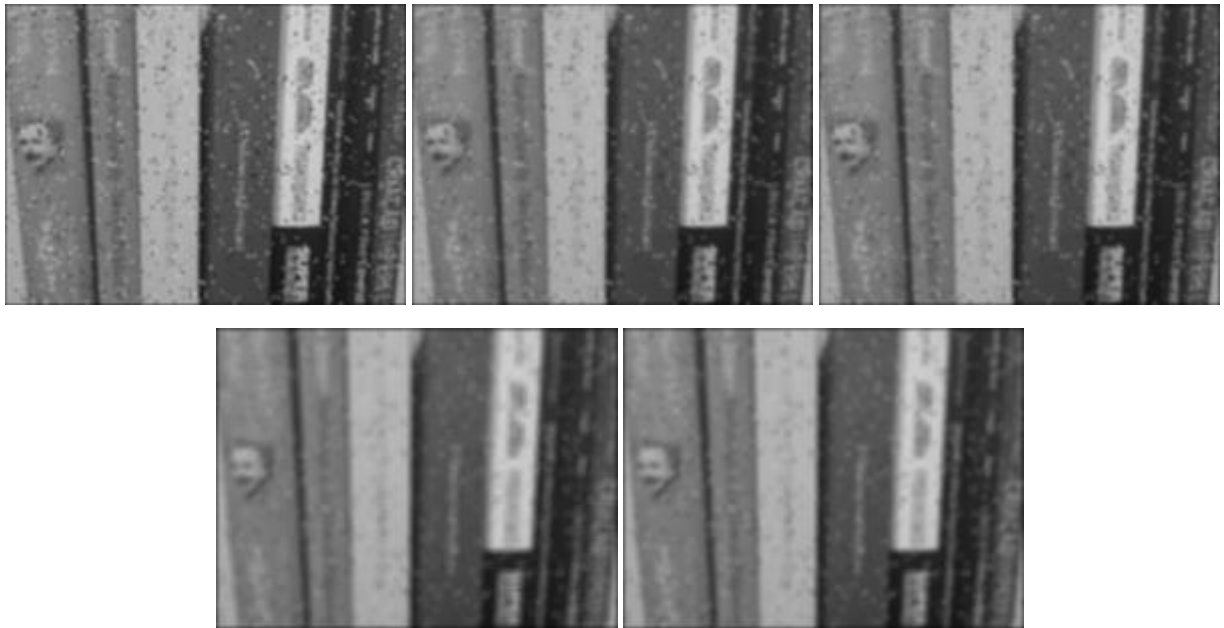


Figura 5 – Ruído Gaussiano, Filtros Gaussianos



Figuras 6 – Ruído Gaussiano, Filtros de Média



Figuras 7 – Ruído Sal-e-Pimenta, Filtros Gaussianos



Figuras 8 – Ruído Sal-e-Pimenta, Filtros Gaussianos

É possível observar que de modo geral todos os filtros resultaram em imagens borradas quando comparadas às imagens ruidosas originais da Figura 4. Porém, o comportamento geral difere muito de acordo com as combinações utilizadas. Primeiramente, observa-se pouca alteração entre as imagens (a), (b) e (c) (primeira linha de imagens) dos conjuntos de filtros gaussianos. Tais imagens utilizam o mesmo desvio padrão na rotina geradora dos filtros, e conforme explicado mais detalhadamente no item 3 das questões teóricas, estes filtros apresentarão pesos muito similares entre si para os valores centrais, de modo que o aumento do tamanho do filtro resultará em efeito análogo a um preenchimento de bordas do filtro com valores zero.

Em contraste, podemos perceber que as imagens se tornam progressivamente mais borradas na segunda linha, de imagens (d) e (e), que apresentam tamanho igual entre si (filtro quadrado de lado 5), mas desvios-padrão de valores 1.5 e 2.

É possível também, observar que os filtros de média em geral borram mais intensamente as imagens do que filtros gaussianos de mesma dimensão. Isso se deve ao fato de que o filtro de média pondera os pixels de sua vizinhança de forma igual, atenuando fortemente quaisquer detalhes presentes nas regiões

centrais da vizinhança. Isso contrasta com o filtro gaussiano, que dá maior peso justamente à região central, o que ajuda a manter detalhes locais.

Esta mesma característica, entretanto, que faz com que os filtros de média consigam atenuar mais facilmente o ruído sal e pimenta do que os filtros gaussianos. Enquanto na imagem com ruído gaussiano

Item c – O item começa por indicar que sejam geradas imagens com detecções de bordas por meio de uma implementação qualquer do algoritmo Canny. Fora escolhida a função *icanny*, do *Machine Vision Toolbox for MATLAB*, mas através da variação dos 3 parâmetros utilizados para a função, não foi possível observar mudanças subjetivamente, apesar de não estar descartada variação dos valores de intensidade da matriz de saída da função.

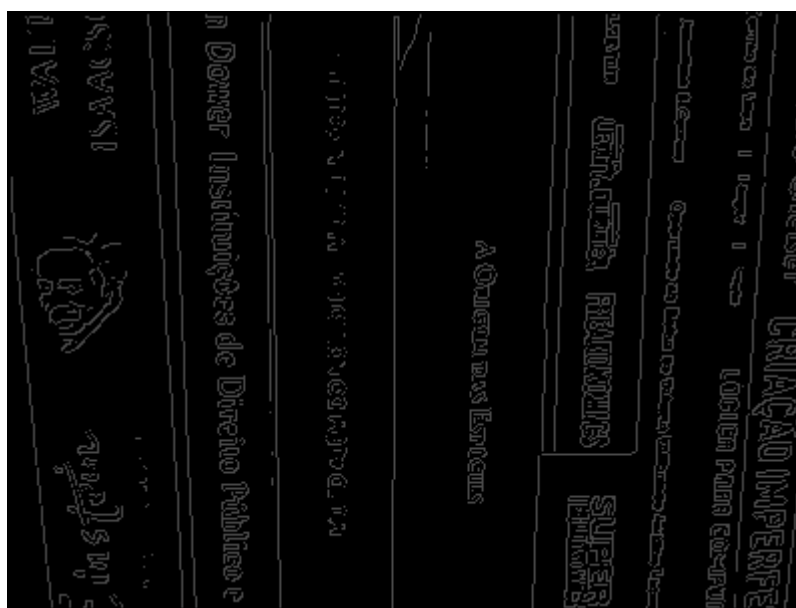


Figura 9 - Saída da função *icanny* com parâmetros padrão

Então, experimentamos com aspectos de escala das imagens, obtendo as figuras 10, para escalas 1.0, 0.75, 0.5 e 0.25. Para detectar as bordas nestas diferentes escalas, procedeu-se por redimensionar a imagem original, seguindo pela aplicação da função *icanny* com seus parâmetros em valor padrão, e finalmente, reajustou-se a escala das imagens para que retornassem às dimensões originais.



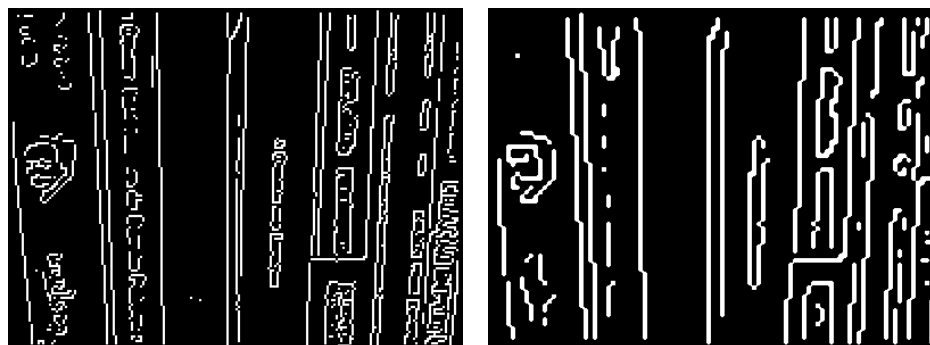


Figura 10 – Detecção de bordas Canny para escalas 1.0, 0.75, 0.5 e 0.25

É possível observar que com a diminuição da escala da imagem, as bordas de maior contraste, mais bem delineadas, como as que delimitam o espaço entre os livros, de modo geral permaneceram bem demarcadas em todas as escalas. Porém é notável que o título do terceiro livro, a contar da esquerda para a direita, apesar de fracamente delineado na escala original, já passa completamente despercebido a partir da escala de 0.5. Em suma, observa-se uma detecção de alta qualidade, resistente à mudanças de escala, mas não infalível.

Por último, em se tratando de comparar diferentes algoritmos de detecção de borda (entre Canny, Robert e Sobel), Canny se mostrou o com a melhor detecção de borda, apresentando definição clara dos contornos da imagem, geralmente sem muitas interrupções de continuidade ou falhas. Em seguida, há o detector Sobel, que apresenta bordas fortes porém mais descontínuas do que as apresentadas pelo algoritmo Canny. Por último, comparativamente, há o algoritmo Robert, que quase não apresenta continuidade, e cujo limiar teve que ser selecionado consideravelmente pequeno para que detectasse borda. O resultado pode ser observado na figura 11.



Figura 11 – Bordas obtidas com detectores de Canny, Robert e Sobel respectivamente (binarizadas).

Item d - A imagem sintética fora construída utilizando uma função geradora de matrizes com números zeros e a função de geração de matrizes de número um. Os valores dos pixels centrais da imagem de zeros foram substituídos pelos valores da imagem de uns, obtendo a figura 12.

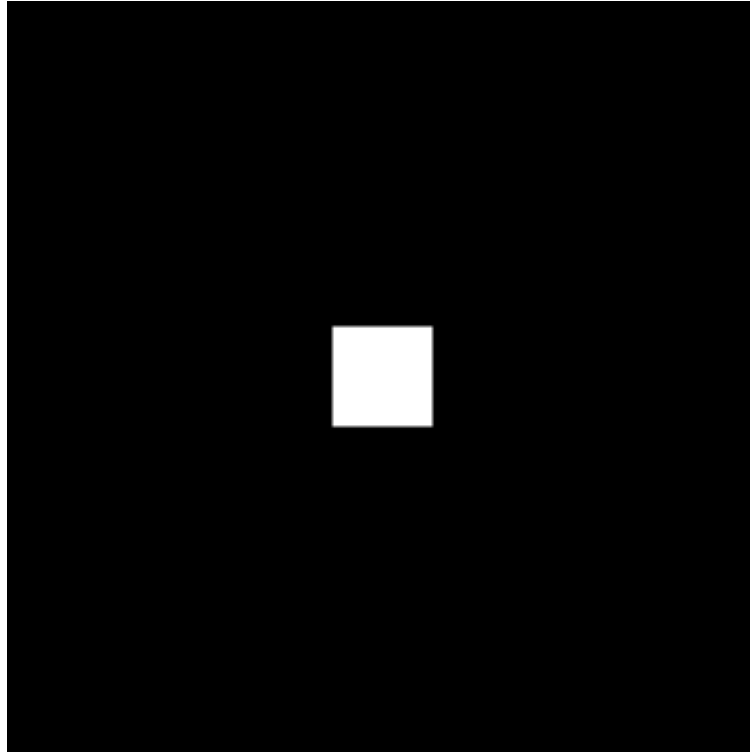
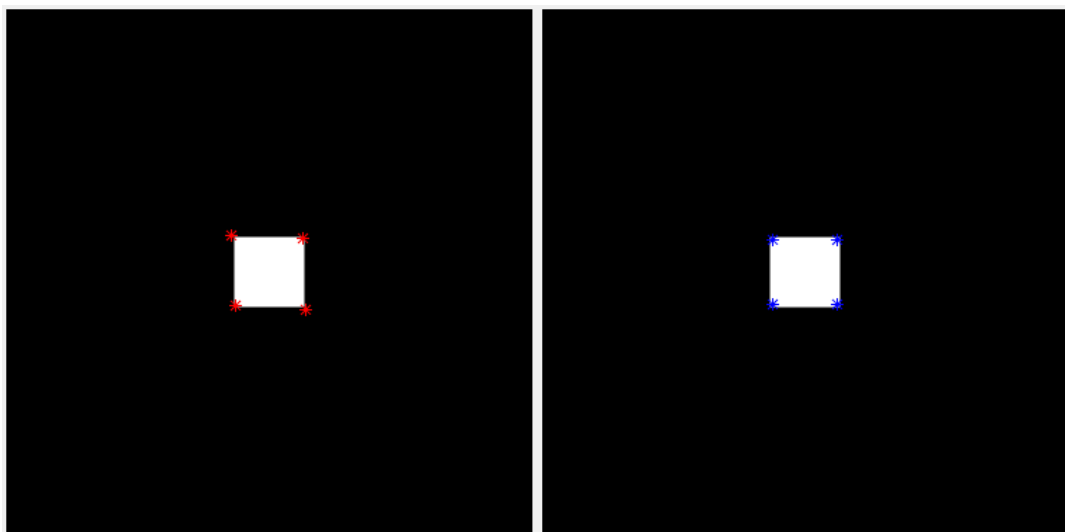


Figura 12 - Imagem sintética de quadrado branco sobre fundo preto

Utilizando esta imagem, foi possível comparar a execução da implementação do algoritmo Corners com a função *icorner* do *Machine Vision Toolbox for MATLAB*. Cada função obteve, como esperado, 4 pontos de canto, que então puderam ser visualizados por meio de comandos de visualização de imagem seguidos por de traço de gráficos (desenhando os pontos sobre a imagem). Os resultados podem ser observados na figura 13



Figuras 13 - Detecção de cantos utilizando algoritmo Corners (esquerda) e função *icorner* (direita)

Podemos observar que o resultado do algoritmo Corners apresentou pequena variação em torno do verdadeiro canto do quadrado branco, em contraste com o obtido com a função *icorner*.



Figura 14 - Detecção de cantos utilizando o algoritmo Corners



Figura 15 - Detecção de cantos utilizando a função *icorner*

É possível notar que essencialmente as duas funções selecionaram pontos distintos da imagem como cantos. Nota-se que há uma correta seleção de pontos pertencentes ao edifício para ambos os algoritmos, mas que a implementação do algoritmo Corners marca muitos pontos em nuvens como cantos, o que não ocorre com a saída da função *icorner*.

Resposta às questões teóricas

1 - Explique quais efeitos você observaria em uma imagem caso a distância focal da lente seja alterada para mais ou para menos.

Considerando a formação de imagens nítidas, a distância focal se apresenta uma variável que expressa uma escolha exclusiva entre ângulo de visão e ampliação. Ou seja, quanto maior a distância focal, menor o ângulo de visão, mas maior a ampliação.

A distância focal também influencia a medida do número F, que indica o nível de intensidade de luz no plano focal e é definida pelo quociente entre a distância focal e a abertura. Se a abertura permanece constante, mas a distância focal aumenta, o número F também aumenta, o que significa uma diminuição da intensidade de luz no plano focal.

2 - Explique o conceito de ruído em imagens, como pode ser quantificado e como pode afetar o processamento das imagens em visão computacional

Se trata de variações indesejadas de intensidade nos valores de intensidade em imagens, que no caso da fotografia digital, em geral são produzidos pelo sistema de aquisição de imagens, pelos mais diversos motivos, variando desde ruído térmico no sensor a efeitos resultantes da manipulação numérica digital dos valores de intensidade.

É possível quantificar o ruído ao o modelar como um termo a ser adicionado a uma imagem sem ruído de modo a produzir uma imagem ruidosa. A imagem sem ruído pode então ser aproximada por meio do valor médio de intensidade de várias imagens da mesma cena. Em posse desta imagem média, seguindo a modelagem estabelecida, a diferença entre esta e uma imagem ruidosa é devido ao próprio ruído, e a partir das diferenças é possível obter métricas como a relação sinal-ruído.

Como a extração de características de imagem é realizada por meio de operações matemáticas realizada sobre os valores de intensidade, flutuações espúrias são capazes de impactar este procedimento, de forma indesejada. Por exemplo, em um processo de detecção de bordas em uma cena com superfície lisa, esta pode se apresentar como granulada na imagem ruidosa, e eventualmente resultar na detecção indevida de uma borda pelo algoritmo de extração de bordas.

3 - Explique porque a precisão de amostragem de um filtro Gaussiano 1-D com $\sigma = 0,6$ não pode ser melhorada utilizando mais de três amostras espaciais (largura espacial maior que 3 pixels).

Em se tratando da construção de um filtro gaussiano, este se dará por medidas de intensidade que seguem o formato delimitado seguindo o formato característico de sino de uma distribuição normal, tendo em sua posição central sua maior intensidade, que então será reduzida até o valor zero em posições progressivamente afastadas do centro, em medida dependente da variância escolhida: quanto maior a variância, menor a maior intensidade do filtro, mas seus valores se encontrarão mais espalhados espacialmente, entre a medida central e posições de valor zero.

Com a medida do desvio padrão de 0.6, um filtro de largura espacial 3, apresentará os valores (obtidos com a função `fspecial`, do MATLAB) dados por [0.1664, 0.6672, 0.1664]. Um filtro de largura espacial 5 e mesmo desvio padrão apresentará os valores [0.0026, 0.1655, 0.6638, 0.1655, 0.0026]

4 - Dadas as definições e classificações de bordas, discuta as diferenças entre bordas em imagens de intensidade e de profundidade. Quais as diferenças nos algoritmos para detectá-las e localizá-las?

Bordas em imagens de profundidade correspondem a descontinuidades espaciais, representando um aspecto geométrico de forma da cena que a imagem representa. Bordas em imagens de intensidade podem corresponder a aspectos geométricos relacionados à forma, mas não necessariamente. Podem expressar apenas características de coloração e textura de objetos da cena.

5 - Explique porque a segmentação H-K não pode ser aplicada a imagens de intensidade na expectativa de se encontrarem cenas de superfícies homogêneas.

O algoritmo utiliza fortemente a geometria diferencial, e imagens de intensidade não garantem medidas de derivadas de superfícies confiáveis à medida que enquanto o cálculo realizado em uma imagem de profundidade estará relacionado à forma da superfície, processo similar realizado sobre imagem de intensidade pode gerar resultados provenientes não da forma, mas da textura da superfície.