

Projeto 1 - Visão Computacional

Renê de Azevedo Delgado - rad@cin.ufpe.br
Centro de Informática, UFPE

10 de Outubro de 2017

1 Introdução

Este relatório é referente ao primeiro projeto da disciplina Visão Computacional do programa de pós-graduação do Centro de Informática - UFPE, período 2017.2. O projeto em questão consiste na resolução de uma lista com 4 questões de visão computacional, utilizando as técnicas aprendidas em sala de aula. O código implementado em Matlab R2012a utilizado na resolução dos problemas será distribuído junto com o relatório mas também pode ser encontrado em [Git].

2 Problemas

Nesta seção explicarei brevemente o problema abordado em cada questão e, em mais detalhes, qual foi a metodologia escolhida para resolução de cada uma delas. Durante a explicação dos algoritmos foi assumido que conceitos de processamento de imagem são conhecidos pelo leitor.

2.1 Encontrar Parafusos e Porcas em imagem

A primeira questão da lista consiste em usar uma imagem de referência para encontrar o número de cada objeto de interesse contido numa segunda imagem.

A imagem de referência dada consiste de dois objetos em tons de cinza escuros e bem separados um do outro por um plano de fundo branco. Uma limiarização global foi feita na imagem, de forma que os objetos de interesse fossem representados pelo preto e o plano de fundo pelo branco. Em seguida, a imagem foi complementada, assim, nossos objetos de interesse agora estão em branco. As propriedades dos componentes 8-conectados da imagem são captadas, ou seja, as propriedades dos objetos de interesse. Para descrever cada objeto, devido às características de forma entre eles serem tão diferentes, foi utilizado as características que relacionam o eixo maior e eixo menor de uma elipse de mesmo segundo momento do que a respectiva componente 8-conectado. Assim, para representar cada objeto foi escolhida a taxa selecionando o maior e o menor eixo da elipse correspondente, descrito na Equação 1.

$$descritor = \frac{maj_axis}{min_axis} \quad (1)$$

onde maj_axis e min_axis representam o eixo maior e o eixo menor, respectivamente e descritor será um escalar que representará o objeto.

O descritor criado na etapa passada deve servir para encontrar os objetos de interesse na imagem de teste. Na etapa de teste, do mesmo modo que anteriormente, a imagem foi limiarizada e depois complementada. Em seguida, cada componente 8-conectado foi encontrado e a taxa entre os eixos das elipses para cada um dos componentes é obtida. A taxa que descreve cada componente é comparada com os descritores obtidos da imagem de referência. Cada componente da imagem de testes será classificado como um determinado objeto como descrito na Equação 2.

$$\begin{cases} \text{se } 0.9 * par_{desc} \leq obj_{desc} \leq 1.1 * par_{desc}, \text{ então classe} = \text{Parafuso,} \\ \text{se } 0.9 * por_{desc} \leq obj_{desc} \leq 1.1 * por_{desc}, \text{ então classe} = \text{Porca.} \end{cases} \quad (2)$$

onde par_{desc} , por_{desc} e obj_{desc} correspondem ao descritor do parafuso, da porca e do objeto encontrado, respectivamente e classe será a classificação do objeto.

Usando o método descrito, o algoritmo foi capaz de classificar corretamente os objetos na imagem de teste.

2.2 Separar região de Interesse do plano desfocado em uma imagem

Nesta segunda questão da lista uma parte da imagem dada está desfocada. O objetivo da questão é que se consiga separar o objeto em primeiro plano da parte desfocada da imagem.

Para separar a parte em primeiro plano da imagem da parte desfocada, foi levado em consideração neste problema o domínio da frequência das imagens. Como o contraste da parte de interesse na imagem é diferente da parte desfocada, pode-se inferir que ambas as partes se encontram majoritariamente representadas em posições diferentes no espectro da frequência.

O método das Diferenças de Gaussianas (DoG) é muito utilizado em processamento de imagens para separação das partes da imagem que estão em uma determinada região de interesse no espectro da frequência. Uma Gaussiana é definida pela Equação 3, onde σ indicará o espalhamento da função. Este tipo de função tem como característica funcionar como um filtro passa-baixa no domínio da frequência de Fourier. Assim, ao declarar um filtro como sendo a diferença entre duas gaussianas com valores diferentes de σ , este filtro passa a admitir apenas as bandas de frequência que se encontram entre os limites destas gaussianas no domínio da frequência de Fourier.

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x)^2/2\sigma^2} \quad (3)$$

Depois de aplicar o DoG à imagem dada, tem-se uma imagem em tons de cinza na banda de frequência selecionada, que deve corresponder à parte de interesse da imagem, e preto em todo o resto da imagem. Em seguida, uma binarização é feita na imagem para separar bem a parte de interesse do resto. Para retirar pontos ruidosos tanto na parte de interesse, representada pelo branco, como no resto, representado pelo preto, uma dilatação seguida por erosão é aplicada à imagem. Assim, cria-se uma imagem onde o branco corresponde às posições de interesse na imagem original e o preto à todo o resto. Esta imagem será usada para separar os objetos em primeiro plano na imagem original.

Para separar o objeto de interesse na imagem original, uma multiplicação elemento-a-elemento é feita de cada canal de cor da imagem original com a imagem em preto e branco criada.

O resultado final foi capaz de segmentar bem a flor que está em primeiro plano, embora partes do caule da flor tenham sido perdidos. Devido às operações morfológicas aplicadas, o resultado final ficou com bordas quadradas. O resultado final obtido é demonstrado em Figura 1.



Figura 1: Resultado final da segunda questão.

2.3 Separar Objeto de interesse em imagem

Nesta questão é pedido para que se implemente um método para segmentar um canguru do resto da imagem e em seguida testar o mesmo método em duas imagens com características semelhantes.

Após uma análise dos canais de cores da imagem do canguru na formatação HSV, foi percebido que boa parte do plano de fundo da imagem possui valores similares na matriz H (Matiz) e poderia ser segmentado através deste valor. Para isolar esse valor específico, foi selecionado o valor da

matiz que mais se repete na imagem. Já que a cor predominante na imagem corresponde à grama, esse valor de moda da matiz foi tido como referência para retirar o plano de fundo da imagem. Então todos os pixels cujos valores de matiz estivessem em um certo grau de proximidade da moda foram classificados como plano de fundo e em seguida retirados da imagem (atribuídos para a cor preta). Após uma transformação da imagem resultante de formato HSV para tons de cinza, foi aplicada operação morfológica de erosão seguido por dilatação para retirar ruídos restantes no plano de fundo e eliminar a ramificação que corresponde à terra próximo às patas superiores do canguru. Com isso parte da pata também é perdida no processo. Por fim, o componente 8-conectado de maior área em pixels é atribuído ao branco e o resto será preto, este componente escolhido corresponde ao canguru. Esta imagem criada será usada para separar o canguru na imagem original através da multiplicação elemento-a-elemento.

Na imagem do canguru o método funciona bem, vide Figura 2a. Apesar de parte do canguru ter sido eliminado e ainda restarem parte de terra na imagem. Como todas as imagens tem características semelhantes, um plano de fundo de cor mais ou menos constantes e o objeto de interesse junto e grande na imagem o método proposto foi capaz de funcionar relativamente bem para todas. Enquanto os cavalos foram bem segmentados perdendo apenas detalhes nas pernas e caudas, Figura 2b, na imagem da águia, Figura 2c, os galhos foram retirados da imagem final devido à operação morfológica para retirar as ramificações da terra na imagem do canguru, assim como parte de seu rosto foi deformado.



(a)



(b)



(c)

Figura 2: Resultados da aplicação do método para a terceira questão.

2.4 Número de Andares de Prédio

Na quarta da questão da lista deve-se implementar um algoritmo que receba a imagem de um edifício e retorne o seu número de andares.

O primeiro passo a ser feito na imagem original é isolar o prédio de todo o resto da imagem. Ao analisar a imagem na formatação de cor HSV, foi percebido que a saturação dos pixels correspondentes era bem menor do que da maioria do resto da imagem. Assim, uma segmentação na matriz de saturação foi feita mas ainda sobraram alguns elementos indesejados. O prédio possui uma matiz bem similar em todas suas partes enquanto os elementos indesejados restantes distoam destes valores. O prédio é separado do restante após eliminar da imagem (atribuir à posição valor

preto) todos os elementos com valores de saturação maiores que um threshold e um matiz diferente do valor encontrado no edifício, este valor de matiz será a moda do valor de matiz na imagem. Para eliminar ruídos e ramificações indesejadas, foi aplicada a operação morfológica de erosão seguida por uma dilatação. A Figura 3 representa a máscara da imagem nesta etapa, note que os andares estão bem separados por uma linha escura. Após isolar o prédio na imagem, as linhas que correspondem às divisões dos andares do prédio devem ser identificadas.

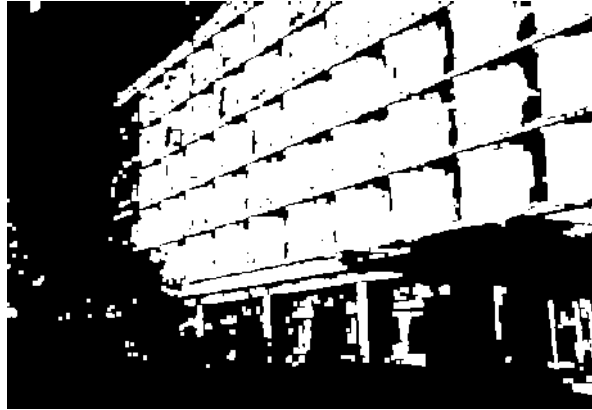


Figura 3: Resultado da segmentação do prédio.

Para detectar as divisões dos andares dos prédio, primeiramente é aplicado filtros de Sobel para detecção das bordas da imagem. Em seguida, é utilizada a transformada de Hough para detectar as linhas mais fortes em uma imagem. A transformada de Hough conta todas as retas que passam pelos pixels válidos da imagem e em seguida seleciona as retas que tem um número máximo de correspondências. Através dos cantos da máscara criada anteriormente as únicas retas relevantes o suficiente para serem detectadas pela transformada de Hough correspondem exatamente às divisões dos andares.

Na Figura 4 estão desenhadas sobre a imagen original as linhas detectadas pela transformada de Hough no passo final do algoritmo, cada linha corresponde a um andar na imagem.



Figura 4: Resultado final da quarta questão.

Referências

[Git] Implementação em matlab. https://github.com/R3NI3/visao_proj.