Universidade de Aveiro
Dep. de Engenharia Mecânica

Sistemas de Visão e Percepção Industrial

Trabalho Prático nº 1 - Março/Abril 2023
Processamento básico de imagens

1 Objetivo

Desenvolvimento de um programa em Matlab para fazer operações de processamento básico de imagens usando transformações geométricas, regiões de pixels como máscaras, filtros e deteção de "arestas". Os resultados a obter consistirão essencialmente na contagem de pixels obedecendo a determinadas regras que o enunciado descreve. No trabalho a entregar os resultados serão pedidos com parâmetros e imagens diferentes dos indicados neste enunciado.

2 Descrição

2.1 Etapas gerais do trabalho

O aluno desenvolve o seu trabalho de uma forma genérica, deixando-o preparado para facilmente alterar parâmetros dos diversos exercícios, conforme indicado adiante no enunciado.

É fornecido aos alunos o comando `SVPI_processTP1_2023.p` que, com os argumentos adequados, servirá para obter os parâmetros e imagens atribuídos a cada aluno, e depois servirá também para executar o código do aluno e gerar os ficheiros de resultados que serão submetidos no E-learning como explicado adiante.

Para testar com parâmetros diferentes dos do enunciado, o aluno recorre a este mesmo ficheiro/-comando que pode ser executado com dois argumentos nulos para gerar parâmetros aleatórios, ou com dois argumentos não nulos para gerar parâmetros fixos. Para o trabalho a entregar, a cada aluno serão depois fornecidos dois argumentos específicos para efeitos de avaliação (p1 e p2). O comando `SVPI_processTP1_2023.p` começa sempre por gerar as imagens a usar nos exercícios, e gera imagens sempre diferentes, independentemente dos argumentos. Para a entrega do trabalho, o aluno invoca o comando com os argumentos que lhe serão indicados no E-learning. Dessa invocação resulta uma listagem de parâmetros para as 8 questões (ver adiante), e que o aluno deve adaptar no seu código para obter as respostas pedidas.

Na entrega do trabalho, o aluno gerará os resultados para entrega e avaliação. Para gerar os resultados no formato a entregar, o aluno deve invocar o mesmo comando `SVPI_processTP1_2023.p` mas agora com os três argumentos: os argumentos p1 e p2 a fornecer, e o nome do ficheiro Matlab (.m) do seu próprio código. Por exemplo, se o ficheiro do aluno se chamar `svpi2023_TP1_123456.m`, então deve-se-á invocar o seguinte comando (com p1 e p2 com valores concretos):

```
SPVI_processTP1_2023( p1, p2, 'svpi2023_TP1_123456.m')
```

Este comando cria na pasta corrente novas imagens, e executa automaticamente o código do aluno que deverá utilizar essas mesmas imagens, de acordo com as instruções dadas neste enunciado. Da execução do comando resultam dois ficheiros com os seguintes nomes genéricos:

- `res_svpi2023_tp1_xxxxxx.txt`
- `cod_svpi2023_tp1_xxxxxx.vsz`

e que serão descritos adiante.

Quando o aluno entender que o trabalho está completo, fará a submissão destes dois ficheiros nos moldes descritos adiante, e o trabalho considera-se entregue. Caso contrário, pode repetir o procedimento as vezes que entender.

2.2 Resumo do procedimento com um exemplo concreto:

1. O aluno com o número 123456 cria o seu código Matlab (em formato function): `svpi2023_TP1_123456.m`
2. O aluno obtém do E-learning, acedendo ao questionário "Argumentos para o TP1 – 2023", a disponibilizar oportunamente, os argumentos designados por p1 e p2 (por exemplo os valores 40 e 35); nesse "questionário" on-line, o aluno deve responder à "questão" com o seu próprio número mecanográfico;
3. Com os argumentos p1 e p2 (40 e 35 ainda como exemplo) o aluno gera os parâmetros do seu trabalho, executando o comando: `SVPI_processTP1_2023(40,35)` na linha de comando do Matlab.
4. Como resultado da execução anterior, são escritos na consola do Matlab os parâmetros a usar para calcular as respostas às questões. O aluno deve usar essas indicações no seu programa `svpi2023_TP1_123456.m`
5. Para gerar os ficheiros de resposta para entregar, o aluno invoca o mesmo comando fornecido, mas agora com três argumentos: os dois argumentos p1 e p2 obtidos no E-learning e o nome do ficheiro do aluno, apresentado o comando um formato similar a:
   `SVPI_processTP1_2023(40,35,'svpi2023_TP1_123456.m')`.
   Note-se que este comando assegura parâmetros constantes para argumentos constantes p1 e p2, mas gerará imagens diferentes a cada execução.
6. O aluno faz upload no e-learning do ficheiro de resultados: `res_svpi2023_tp1_123456.txt`
7. O aluno faz upload no e-learning do ficheiro de controlo: `cod_svpi2023_tp1_123456.vsz`

3 Descrição de comandos e ficheiros

3.1 Formato do ficheiro com o trabalho do aluno (.m file)

O trabalho de cada aluno deve consistir num único ficheiro Matlab (.m) e será obrigatoriamente do tipo *function*. Assim, se um aluno pretender usar funções auxiliares, elas devem estar inseridas neste mesmo ficheiro. Por exemplo, se um aluno desenvolveu as funções `f1()` e `f2()` chamadas pela sua função principal, então o código do seu programa em Matlab (.m) ficaria algo do género:

```
function [nnnnnn,a1,a2,a3,a4,a5,a6,a7,a8]=svpi2023_TP1_nnnnnn()
%nnnnnn -- o num mec do aluno
nnnnn=123456;
a1=f1(); a2=f2(); %% etc.
end

function X=f1()
X=10; % ex.
end

function Y=f2()
Y=10; % ex.
end
```

Portanto, o trabalho será um único ficheiro que inclui as eventuais funções auxiliares dentro do corpo desse mesmo ficheiro. A primeira função do ficheiro coincide, obrigatoriamente, com o nome do ficheiro (à parte a extensão .m) e não aceita argumentos. Essa função principal deve devolver as respostas aos exercícios e pela ordem com que aparecem (a1 é a resposta da questão 1, a2 da questão 2, e assim sucessivamente); deve ainda devolver o número mecanográfico do estudante que será o primeiro valor da lista de retorno: `[nnnnnn, a1, a2, a3, a4, a5, a6, a7, a8]`

Nota: a ausência do número mecanográfico, ou um número mecanográfico inválido, pode inviabilizar a avaliação.

3.2 O comando SVPI_processTP1_2023.p com dois argumentos (p1 e p2)

Este comando serve para gerar parâmetros e imagens específicas para o aluno, de acordo com a sequência do enunciado. Quando executado com dois argumentos, este comando escreve na linha de comando do Matlab os parâmetros das questões num formato similar ao seguinte:

```
Q1----------------------------------------------------------------
108    46   192    89   227

Q2----------------------------------------------------------------
5     7     1

Q3----------------------------------------------------------------
0     1     0
1     1     1
0     1     0

Q4----------------------------------------------------------------
100   200    100    30

Q5----------------------------------------------------------------
1    65   105   98    173     2   89   191   174    119

Q6----------------------------------------------------------------
5     3     5     7

Q7----------------------------------------------------------------
1     1     1     0     1
1     0     1     0     1
1     0     1     1     1

Q8----------------------------------------------------------------
460   10
```

Estes parâmetros são os que se devem substituir no enunciado, e pela ordem em que aparecem dentro de cada questão, nos locais onde o texto está em caixas. Por exemplo, para a questão 6 enumerada adiante, e conforme os parâmetros acima, deve-se ler o seguinte no enunciado:

> (...) a menos de 5 pixels dos bordos [...] forçar uma região quadrada de pixels brancos com 3 pixels de lado .[...] filtro de mediana de 5×7 (5 linhas por 7 colunas) (...)

O aluno deve ajustar o seu código para inserir os parâmetros requeridos. Nas questões Q3 e Q7 o parâmetro é único e é uma matriz (padrão). Nos restantes casos, os parâmetros estão todos em linha. Os parâmetros dados aos alunos pelo comando `SVPI_processTP1_2023.p`, quando chamado com dois argumentos, podem vir em ordem aleatória das questões, i.e., as questões poderão aparecer no ecrã por ordem diferente da ordem natural. Por exemplo, a sequência anterior poderia ter vindo na seguinte ordem:

```
Q4----------------------------------------------------------------
100   200    100    30

Q6----------------------------------------------------------------
5     3     5     7

Q1----------------------------------------------------------------
108    46   192    89   227
```

```
Q2-------------------------------------------------------------
5     7     1

Q8-------------------------------------------------------------
460   10

Q3-------------------------------------------------------------
0     1     0
1     1     1
0     1     0

Q7-------------------------------------------------------------
1     1     1     0     1
1     0     1     0     1
1     0     1     1     1
Q5-------------------------------------------------------------
1    65   105   98    173     2   89   191   174    119
```

Se for invocado com agumentos (0,0) o comando gerará sempre parâmetros aleatórios.

3.3 Comando `SVPI_processTP1_2023.p` com três argumentos (p1, p2 e ficheiro)

Para gerar os resultados a entregar, o comando deve ser executado com três argumentos. Assim, ele gera as imagens, executa o código do aluno e gera os ficheiros a entregar. Entre outras funções, este comando faz o seguinte:

1. verifica se o ficheiro indicado como terceiro argumento existe na pasta corrente.
2. verifica se o número de argumentos está correto (são 3: p1, p2 e nome do ficheiro do aluno)
3. gerará a(s) imagem(s) necessárias (as 5 imagens são geradas e apagam anteriores do mesmo nome)
4. executa o ficheiro do estudante (se ocorrer um erro aborta a execução)
5. verifica se os parâmetros de retorno estão no número pedido (são 9 = 1 + 8)
6. escreve em ficheiro (e na consola) uma linha com os resultados e outros valores de controlo (*.txt)
7. gera um ficheiro encriptado com o código do programa (com extensão *.vsz)

Se em alguma destas etapas surgir um erro (falta de permissões de escrita, falta de espaço em disco, path com caracteres não permitidos, ou outros) o comando termina sem concluir os resultados pedidos, e o erro deverá ser corrigido pelo aluno. Após execução com sucesso, dois ficheiros terão sido gerados. Esses ficheiros serão usados para avaliação e serão carregados no E-learning nas entradas adequadas.

3.4 Ficheiro `res_svpi2023_tp1_xxxxxx.txt`

Ficheiro legível gerado pelo comando `SVPI_processTP1_2023.p` e que inclui os resultados do trabalho e outros valores de controlo e verificação de autenticidade. NÃO editar de forma nenhuma este ficheiro antes de o entregar. A alteração de qualquer dos seus valores ou dimensão da linha torná-lo-á inválido e não será aceite na avaliação. Formato da linha de resultados gerado pela execução de `SPVI_processTP1_2023.p`:

xxxxxx	a1	a2	a3	a4	a5	a6	a7	a8	CTRL1	CTRL2	P1	P2	P3	C1	C2	C3	...	Ci	N
Num. mec	Resultados pedidos sobre o trabalho e gerados pelo programa do aluno								Mensagens de controlo		Argumentos de execução			Coeficientes auxiliares de verificação					

Os primeiros 9 campos, **Num. mec** e **Resultados pedidos**, são os que o programa do aluno gera como retorno da sua função principal; os restantes parâmetros são gerados automaticamente como elementos de controlo e verificação. Esta linha de resultados tem o mesmo formato do que é escrito

no ficheiro `res_svpi2023_tp1_xxxxxx.txt`. Se os resultados devolvidos pelo programa do aluno não obedecerem ao formato pedido, e em especial o número mecanográfico como primeiro campo, não será possível fazer avaliação. **NB**. O ficheiro terá várias linhas como resultado de várias execuções com imagens diferentes e que também ficam disponibilizadas na pasta de execução.

3.5 Ficheiro `cod_svpi2023_tp1_xxxxxx.vsz`

Este ficheiro é também gerado pelo comando `SVPI_processTP1_2023.p`, quando invocado com três argumentos, e que contém o código do programa do aluno (o que foi usado para gerar os resultados). Está encriptado e portanto não é legível! Serve apenas para efeitos de entrega no E-learning. A não entrega deste ficheiro ou a entrega de um ficheiro diferente implica a anulação do trabalho.

Nota: os dois ficheiros referidos estão intimamente relacionados entre si e têm de ser os dois entregues para o trabalho poder ser avaliado.

4 Submissão e recomendações

Os dois ficheiros gerados (`res_svpi2023_tp1_xxxxxx.txt` e `cod_svpi2023_tp1_xxxxxx.vsz`) serão submetidos pelo E-learning nas entradas criadas para o efeito. Os comandos fornecidos estão preparados para versões de Matlab R2022 e posteriores. Em caso de eventuais incompatibilidades, contactar urgentemente o docente. Nalgumas situações, antes de dar por terminado um exercício, é recomendável experimentar os algoritmos em imagens pequenas que permitam verificação manual dos resultados. Um algoritmo robusto funciona independentemente da dimensão da imagem. Na execução do trabalho para entrega recomenda-se o seguinte:

- Trabalhar numa pasta vazia onde só constem o programa do aluno (1 ficheiro) e o comando fornecido (`SVPI_processTP1_2023.p`)
- Evitar usar funções do Matlab que possam não estar nas toolboxes comuns usadas nas aulas.
- Ter algum cuidado no nome de funções que possam coincidir com outros nomes existentes.

5 Avaliação

A avaliação é baseada em dois parâmetros principais:

1. Conformidade dos resultados do trabalho de acordo com imagens de avaliação.
2. Funcionamento do código fornecido.

A conformidade dos resultados é feita com base na taxa de resultados corretos em comparação com os resultados reais resultantes de uma análise correta às imagens de avaliação. A eficiência e/ou robustez do código fornecido poderá eventualmente ser levada em conta na apreciação final do trabalho.

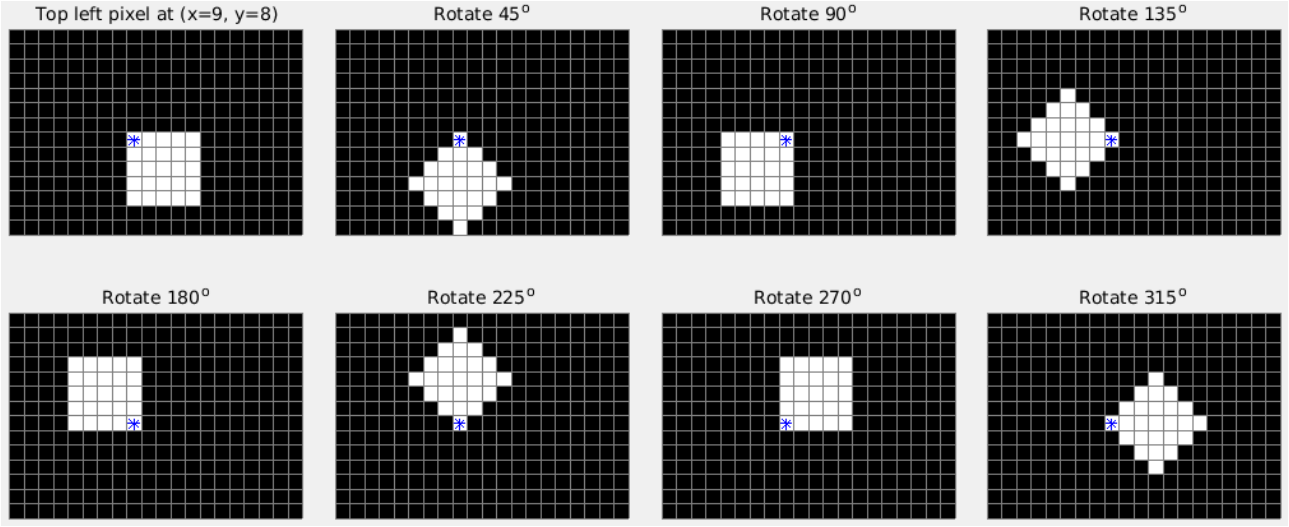
O código entregue pelos alunos será sujeito a um sistema de verificação de plágio (MOSS).

6 Enunciado genérico do trabalho

6.1 Observações preliminares

- Os exercícios a responder são os indicados adiante, mas as indicações dentro de uma caixa serão variáveis no trabalho final a entregar.
- Para todas as indicações, o pixel `(1,1)` é o do canto superior esquerdo da imagem.
- Quando se refere um pixel por um par ordenado `(px,py)` na imagem, o termo `px` representa a contagem horizontal (coluna) e o termo `py` representa a contagem vertical (linha).
- As imagens apresentadas no enunciado servem apenas para ilustrar o que é pedido no problema; as imagens a usar na avaliação serão diferentes.
- Se for necessário considerar pixels para além dos limites das imagens a analisar, o valor desses pixels será zero (preto).

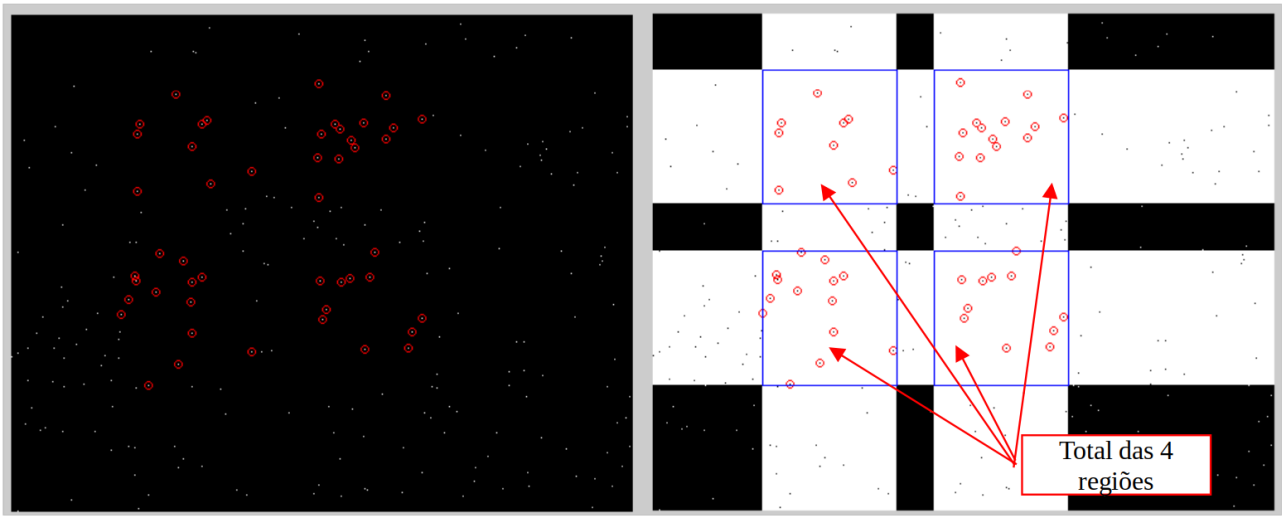
- Nas operações que envolvam rotação de um objeto na imagem, considera-se que ela ocorre em torno do centro do pixel superior esquerdo desse objeto na imagem como se ilustra:



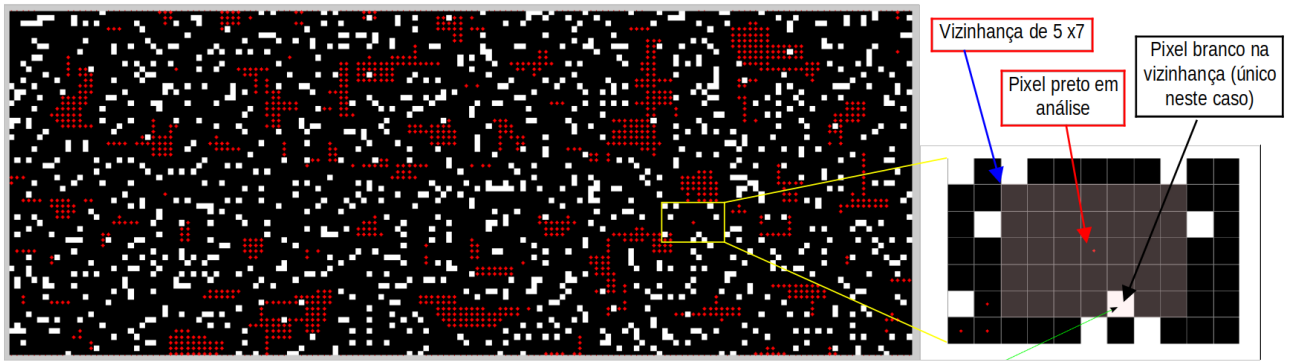
- Na geração de imagens que servem de máscaras, devem-se usar imagens do tipo `logical` em vez de `double` ou `uint8`. Por exemplo, para gerar uma máscara quadrada de 100 por 100 fazer um comando do género: `M=true(100)`. Desta forma, os resultados de `imwarp` sobre `M` não geram níveis de cinzento.
- As vizinhanças e os filtros dados serão sempre em número ímpar de linhas e de colunas.
- **NB**. Devido ao sistema de coordenadas do Matlab nas operações em imagem, os ângulos positivos referem-se a medição/rotação no sentido horário, e os ângulos negativos ao sentido anti-horário.

6.2 Questões a responder

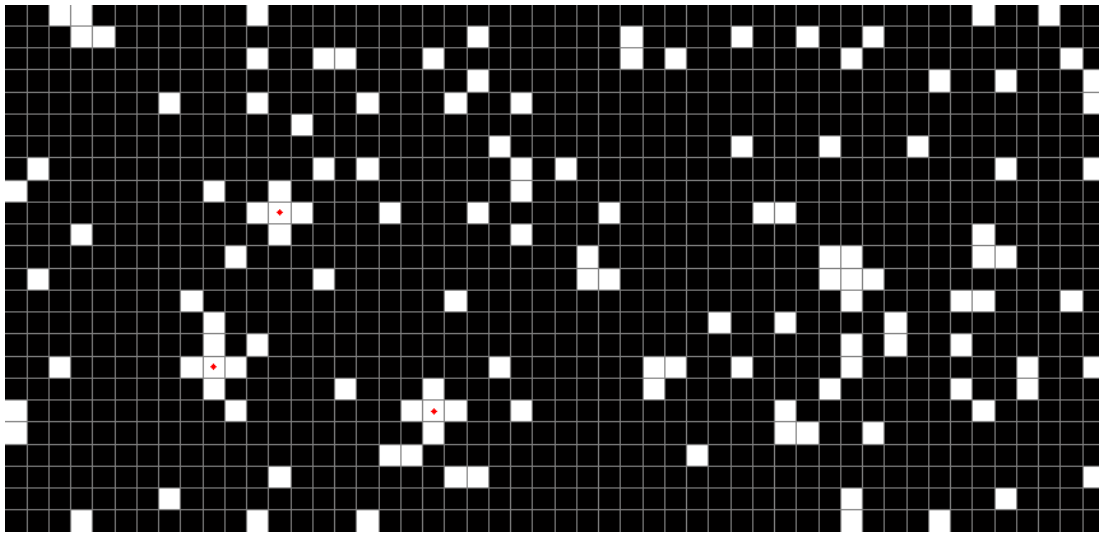
1- Na imagem `image_TP1_2023_1.png` calcular o número total de pixels brancos que se encontram nas 4 regiões dadas pelas interseções de duas faixas horizontais com duas faixas verticais. Todas as 4 faixas têm uma largura (a menor dimensão) de 108 pixels. As faixas horizontais iniciam-se respetivamente nas linhas 46 e 192, e as faixas verticais iniciam-se nas colunas 89 e 227. As faixas atravessam a imagem até aos seus bordos. NB. Pode acontecer que alguma das faixas ultrapasse os limites direito ou inferior da imagem: nesses casos, a região fora da imagem não será levada em conta. As faixas paralelas nunca se sobrepõem entre si.



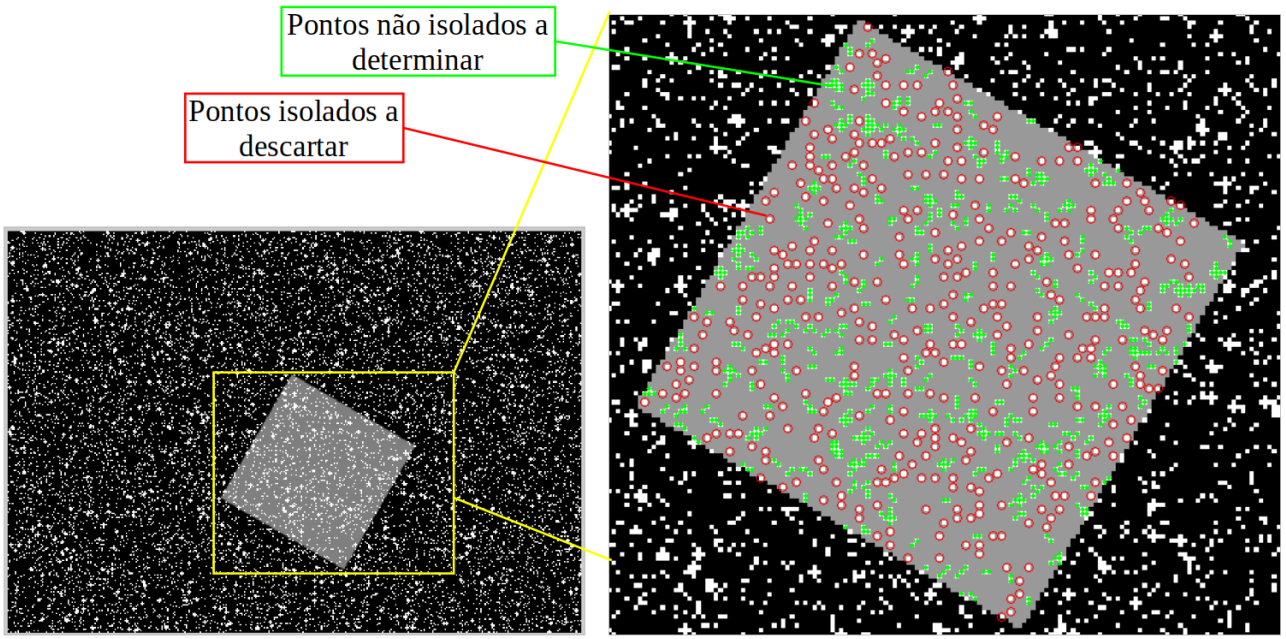
2- Na imagem `image_TP1_2023_1.png` determinar o número de *pixels* pretos que na sua vizinhança de 5 × 7 (5 linhas por 7 colunas) têm no máximo 1 pixel branco. Todos os *pixels* devem ser considerados, incluindo nos bordos da imagem. **NB**. Quando se diz "no máximo 1 *pixel* branco" isso implica naturalmente 1 ou 0 *pixels* brancos.



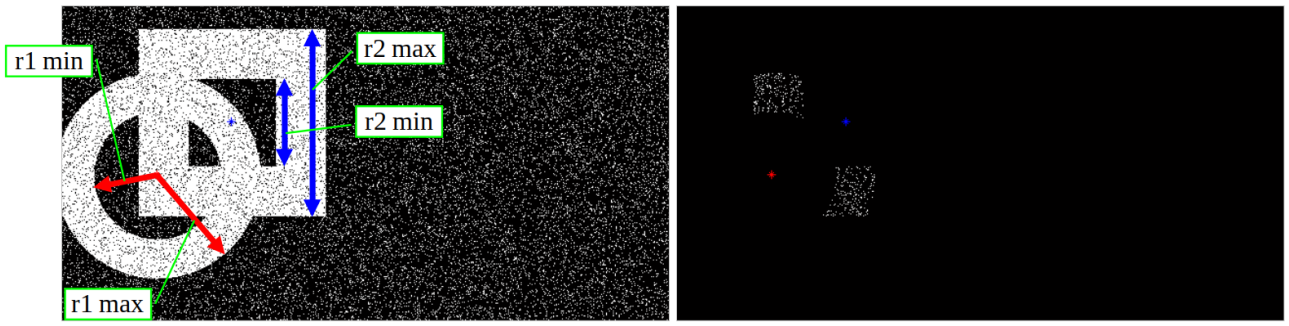
3- Na imagem `image_TP1_2023_2.png` determinar o número de *pixels* brancos que na sua vizinhança têm pixels brancos exatamente de acordo como indicado pela matriz F=[0 1 0; 1 1 1; 0 1 0] na notação matricial do Matlab. **NB**. O *pixel* em análise define o centro da vizinhança com as dimensões de F.



4- Na imagem `image_TP1_2015_2.png` considerar uma região quadrada de 100 *pixels* de lado, cujo vértice superior esquerdo está no *pixel* (200 , 100) e com uma rotação em torno desse *pixel* de 30° . Dos pixels da região, quantos são pixels não isolados no contexto de toda a imagem? **NB**. É possível que a região ultrapasse os limites da imagem.



5- Na imagem `image_TP1_2023_3.png` quantos pixels brancos são comuns (portanto, contidos simultaneamente) a uma coroa de tipo 1 com os parâmetros 65 ≤ r1 ≤ 105 e (x01, y01)=(98,173) e a uma coroa de tipo 2 com os parâmetros 89 ≤ r2 ≤ 191 e (x02, y02)=(174, 119)? **Definições**: Coroa tipo 1→circular; coroa tipo 2→quadrangular, coroa tipo 3→circular pelo exterior e quadrangular pelo interior; coroa tipo 4→quadrangular pelo exterior e circular pelo interior. **Nota**: as coroas poderão ultrapassar os limites da imagem. Nas coroas com limites quadrangulares os valores de r indicam os lados dos quadrados (ver a figura seguinte). Em todos os tipos de coroas os centros são os centros geométricos das figuras. Note-se que os limites de r1 e r2 são todos inclusivos na definição das coroas (≤)



6- Na imagem `image_TP1_2023_3.png` começar por colocar a zero todos os pixels brancos que estejam a menos de 5 pixels dos bordos da imagem, ou seja, os pixels das linhas e colunas {1,2,...,4} e das linhas e colunas {end,end-1,....,end-3}, e obter uma imagem T. Depois, centrado em cada um dos pontos isolados (pretos ou brancos) da imagem T, forçar uma região quadrada de pixels brancos com 3 pixels de lado, mas mantendo os restantes pontos da imagem T. De seguida, aplicar repetidamente um filtro de mediana sobre a imagem que vai resultando da aplicação anterior desse filtro de mediana. Quantas vezes se deve passar o filtro de mediana de 5×7 (5 linhas por 7 colunas) até a imagem estabilizar? (Ou seja, quando a aplicação do filtro de mediana não alterar mais a imagem sobre a qual é aplicada).

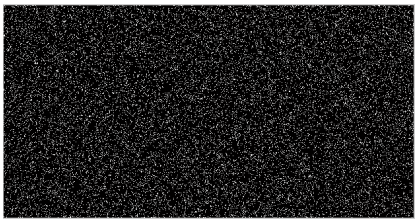


Imagem original

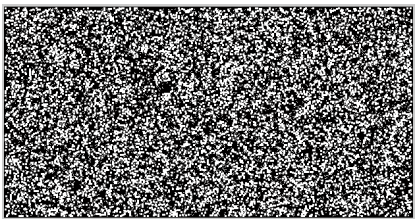


Imagem alterada onde se vai aplicar o filtro de mediana pela 1ª vez

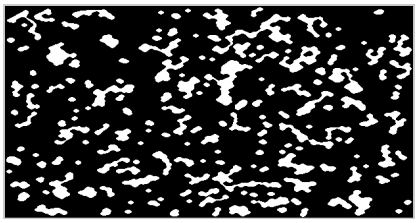


Imagem estabilizada após a aplicação sucessiva do filtro de mediana

Formalmente, esta operação da aplicação sucessiva do filtro de mediana pode-se traduzir assim:

- $f_1(x, y)$ é uma imagem binária (imagem inicial).
- $M_{[5x7]}\{f_i(x, y)\}$ é o resultado da aplicação de um filtro de mediana de 5×7 sobre $f_i(x, y)$.
- $f_{n+1}(x, y)$ é o resultado da aplicação de $M_{[5x7]}$ em $f_n(x, y)$.
  - Por exemplo, $f_2(x, y) = M_{[5x7]}\{f_1(x, y)\}$
- Iniciando o processo com $n = 1$, qual o valor mínimo de n para que seja verdadeiro o seguinte?
  - $f_{n+1}(x, y) = M_{[5x7]}\{fn(x, y)\} = f_n(x, y)$

Exemplos e considerações adicionais:

- Se $f_2(x, y) = M_{[5x7]}\{f_1(x, y)\} = f_1(x, y)$, então bastou aplicar a função uma só vez e $n = 1$.
- Se $f_1(x, y)$ consistir de um único pixel branco no centro então: $f_2(x, y) = M_{[5x7]}\{f1(x, y)\} \neq f_1(x, y)$

mas

- $f_3(x, y) = M_{[5x7]}\{f_2(x, y)\} = f_2(x, y)$ (e $f_2(x, y)$ será constituída só por pixels pretos), e a resposta será $n = 2$.

7- Na imagem `image_TP1_2023_4.png` determinar quantos são os pixels brancos que podem ser o centro de um padrão A de forma a que **nenhum** pixel (preto ou branco) do padrão fique na vizinhança N8 de qualquer outro pixel (preto ou branco) de outro padrão. Na notação matricial do Matlab a matriz A é: A=[1 1 1 0 1; 1 0 1 0 1 ; 1 0 1 1 1]. **Nota**: São permitidos padrões parciais nos bordos da imagem desde que não interfiram com outros padrões.

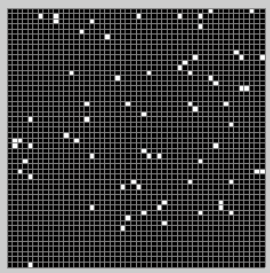
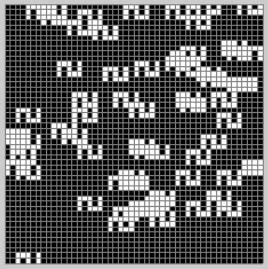
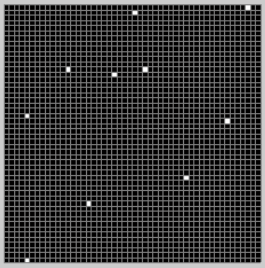


Imagem original com pixels aleatórios



Todas as possibilidades de repetição do padrão A incluindo os casos inválidos.



Pixels válidos (pedido no exercício)

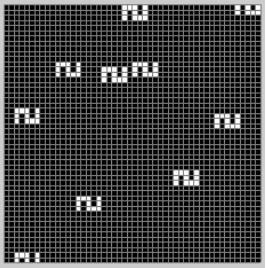
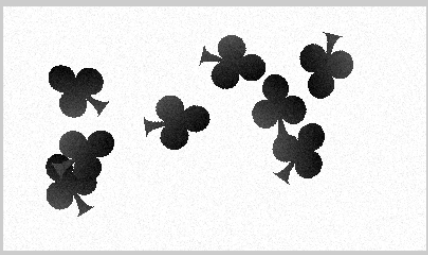


Ilustração do padrão da matriz A em torno de cada um dos pixels válidos

8- Na imagem `image_TP1_2023_5.png` estão presentes vários símbolos iguais entre si em diversas posições e orientações. Com base na deteção de "arestas", determinar quantos são os símbolos isolados, ou seja, os que não estão em contacto, ou sobrepostos, com outros símbolos. O critério para considerar um símbolo isolado neste problema é o número total de pixels da sua "aresta", e que será de 460 ± 10 pixels. **Sugestão**: Para que este critério tenha mais sucesso, a técnica da deteção de "arestas" deve dar origem a contornos fechados, especialmente para os símbolos isolados. Pode dar-se o caso de, numa dada imagem, não haver nenhum símbolo isolado. A resposta nesse caso é zero!



Exemplo de imagem fornecida com símbolos isolados e sobrepostos.

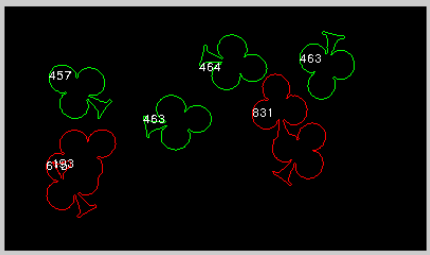


Ilustração das arestas detetadas e dos seus "comprimentos"

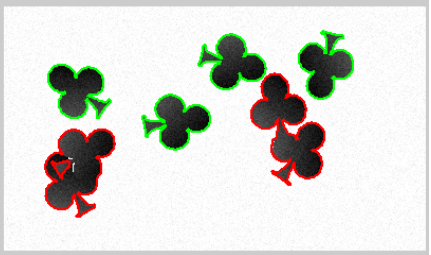
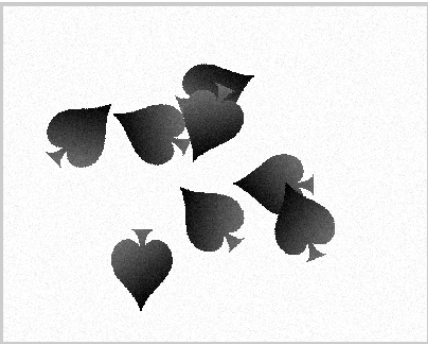
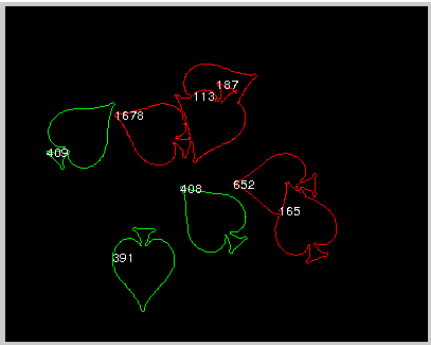
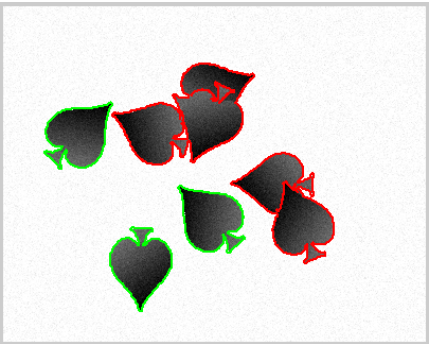


Ilustração da sobreposição da deteção de "arestas" com a imagem original







Outro exemplo com parâmetros diferentes.