



Member of **STEMMER**[®]
IMAGING



MANUAL DE PRÁTICAS SHERLOCK PROFESSIONAL

VERSÃO 2.1

INFAIMON PORTUGAL
Departamento técnico

infaimon.pt@infaimon.com

ÍNDICE

Introdução.....	4
Ajudas do Sherlock	4
Sherlock Professional – Versão Demo	6
Conhecer o Sherlock.....	7
1. Barra de atalhos e menu principal.....	7
2. Adicionar e remover janelas de visualização	8
3. Opções avançadas de configuração	8
Exercício 1 – Análise de <i>Blob</i> e pré-processadores.....	10
1. Criar uma nova investigação	10
2. Carregar uma imagem.....	10
3. Definir a região de interesse.....	11
4. Aplicar pré-processadores.....	13
5. Aplicar o algoritmo	15
6. Carregar um lote de imagens	17
Exercício 2 – Leitura de códigos e alinhamento	20
1. Iniciação da investigação	20
2. Carregar imagens	20
3. Aplicação da ferramenta de leitura de códigos 2D.....	20
4. Criar um alinhamento.....	22
5. Aplicar o alinhamento	26
Exercício 3 – Medição de distâncias e calibração do sistema.....	27
1. Iniciação da investigação	27
2. Carregar imagens	27
3. Criar um alinhamento inicial.....	27
4. Aplicação de ferramenta para medição das aberturas da peça	30
5. Aplicar uma calibração	33
Exercício 4 – Cor e comunicação	36
1. Iniciação da investigação	36
2. Carregar uma imagem.....	36
3. Aplicação da ferramenta <i>Color Map</i>	37
4. Segmentação da imagem	38
5. Contagem dos comprimidos verdes	40
6. Apresentação dos resultados	41
7. Comunicação <i>TCP/IP</i>	43

Exercício 5 – Ligação de uma câmara	46
1. Instalação do <i>Sentinel</i>	46
2. Licenças	46
3. Configuração dos parâmetros da câmara no <i>Sapera CamExpert</i>	47
4. Adicionar a câmara ao <i>Sapera Acquisition Wizard</i>	48
5. Adicionar a câmara ao Sherlock	49
ANEXO A	51
Instalação do Sherlock	51

Introdução

O objetivo deste manual é ajudar o formando a familiarizar-se com o *software* Sherlock 7, por forma a conseguir desenvolver um programa completo.

O Sherlock 7 é um *software* gráfico de visão no qual se pode desenvolver, testar e fazer o *debug* de aplicação de visão industrial.

Uma aplicação de visão industrial típica é composta por quatro passos:

1. Aquisição de imagens
2. Extração de dados das imagens
3. Análise dos dados
4. Envio dos dados ou tomada de decisão baseada na análise dos mesmos (escrita de resultados num ficheiro, comandar um mecanismo de rejeição, escrita em registos de PLC's, etc.)

Cada um destes quatro passos é conseguido através da seleção de elementos de programação a partir da interface gráfica do Sherlock, adicionando-os ao programa e configurando as suas opções de execução. Desta forma, não é necessária a escrita de código para criar uma aplicação de visão no Sherlock.

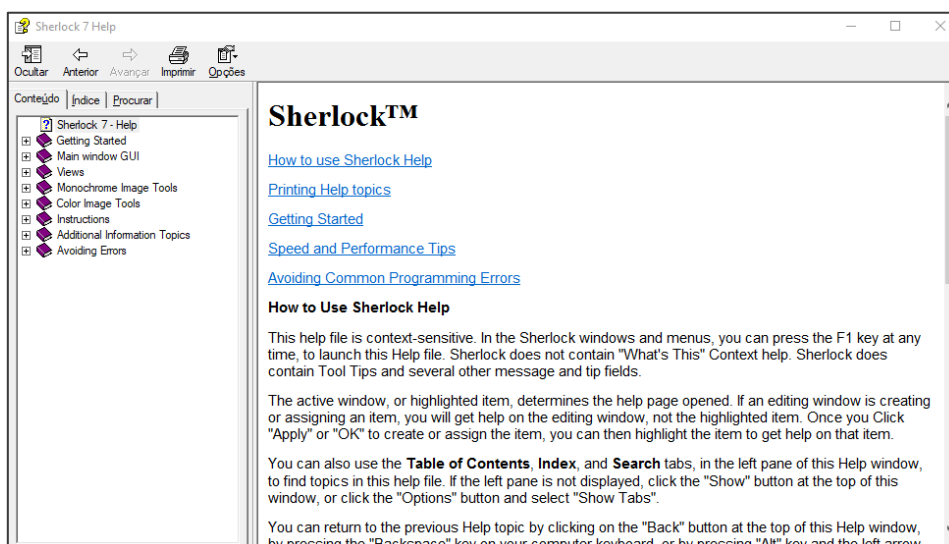
Uma aplicação completa desenvolvida em Sherlock é designada de investigação e é gravada num ficheiro com extensão ".ivs".

Por forma a executar os exercícios propostos, necessitará de instalar o *software* **Sherlock Professional 7.2.6.4** ou superior no computador.

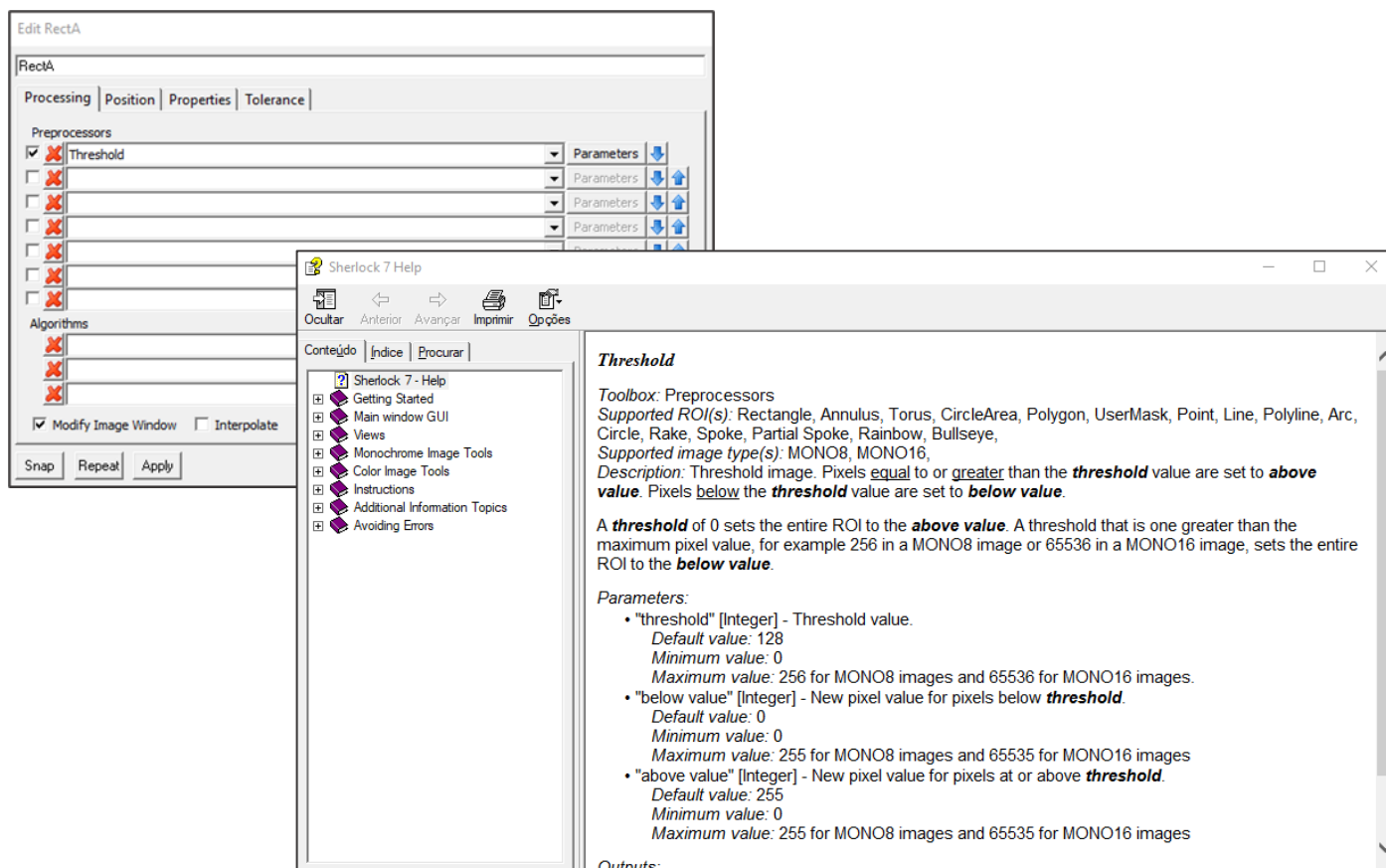
Este manual irá incidir apenas nas funcionalidades principais e fundamentais para trabalhar com o *software*.

Ajudas do Sherlock

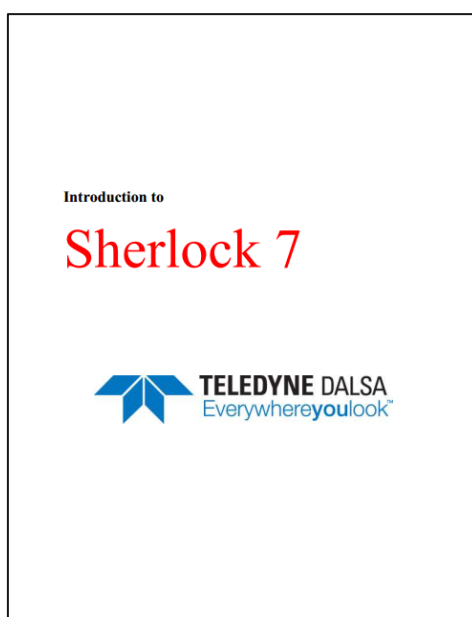
Para informações de programação ou parametrização não mencionadas neste manual, ou para informações mais detalhadas, poderá ser utilizada a ajuda *online* do Sherlock. Para aceder à Ajuda, basta aceder pelo menu principal a **Help -> Help Topics**, ou carregando na tecla **F1**.



Caso se pretenda obter ajuda com um pré-processador ou algoritmo basta utilizar o atalho de ajuda **F1** para a janela de ajuda do Sherlock abrir com informação desse pré- processador ou algoritmo.

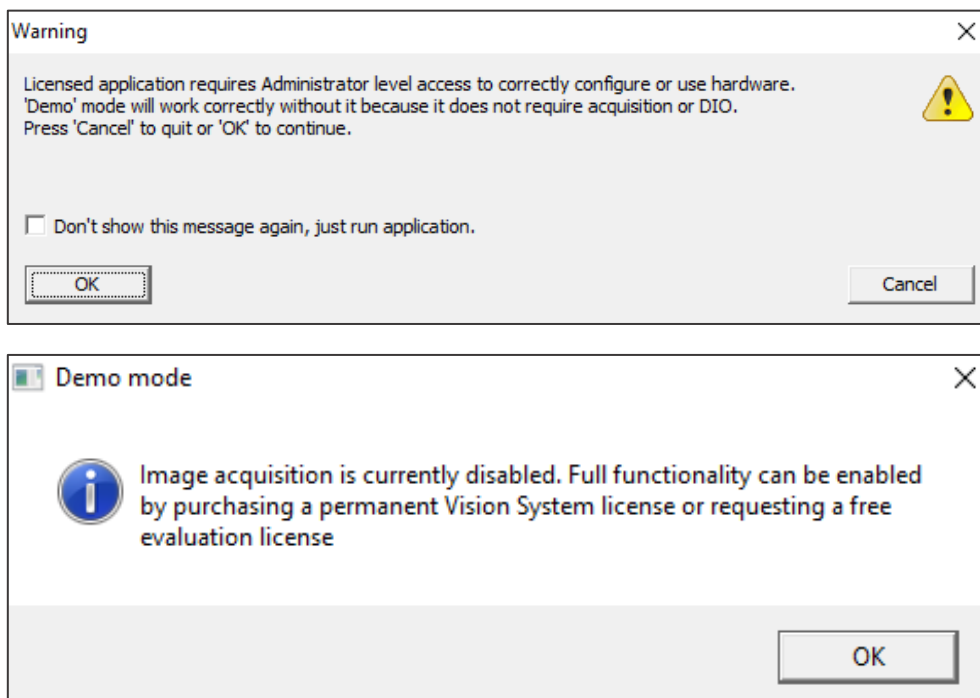


Para além da ajuda online, e da informação contida neste manual, a descrição detalhada de todo os menus e configurações pode ser encontrada no documento **Sherlock7 Training Manual.pdf** que acompanha este manual de práticas.



Sherlock Professional – Versão Demo

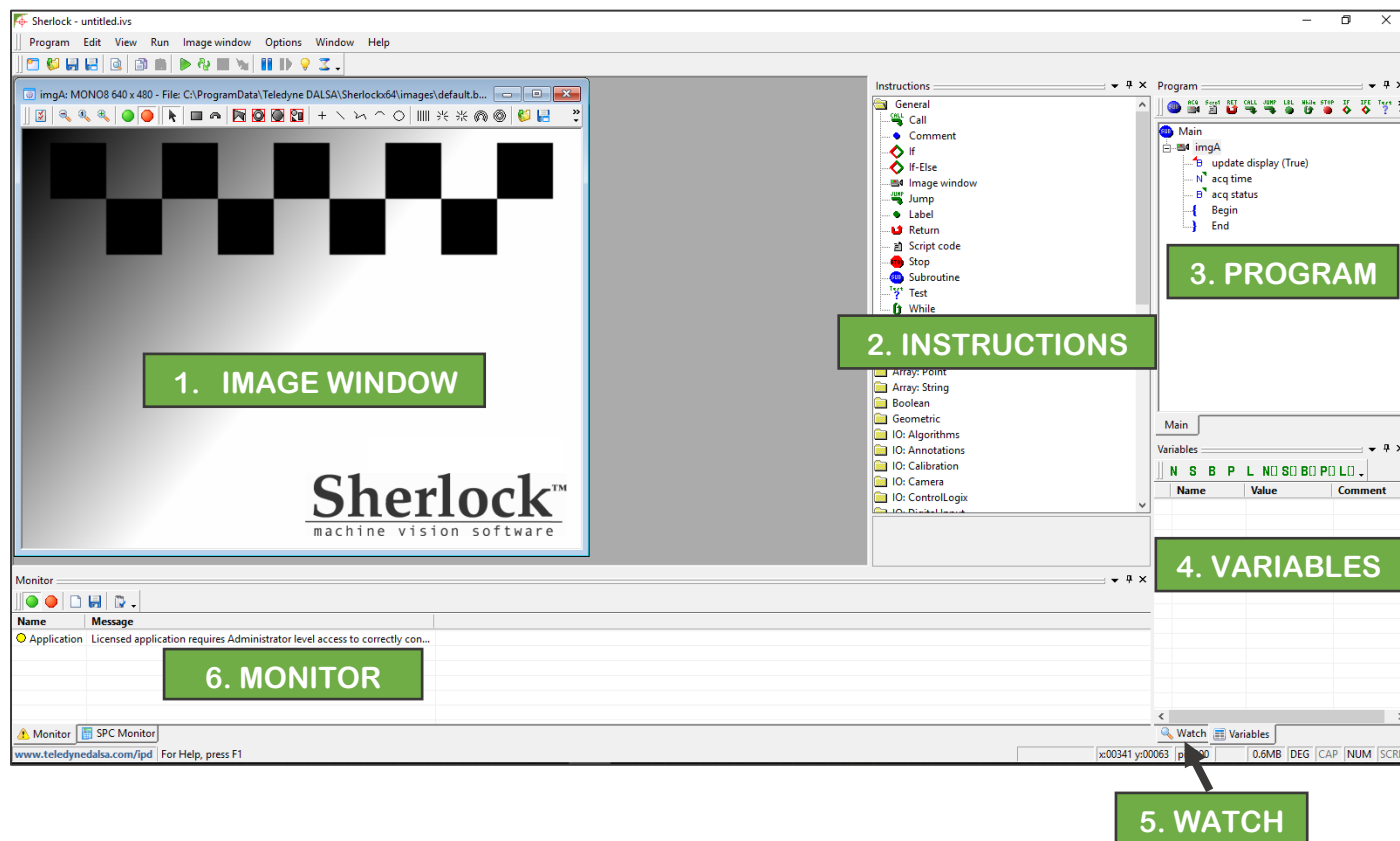
Após a instalação, basta executar a aplicação **Sherlock** ou **Sherlockx64** (versão de 64bits). Caso não exista uma licença válida, duas mensagens aparecerão a informar das limitações da versão de demonstração.



De uma forma resumida, a versão de demonstração do Sherlock possui todas as funcionalidades da versão profissional disponíveis, incluindo a parte das comunicações. A única exceção é que não permite a aquisição de imagem direta de uma câmara pois não é possível configurar hardware.

Conhecer o Sherlock

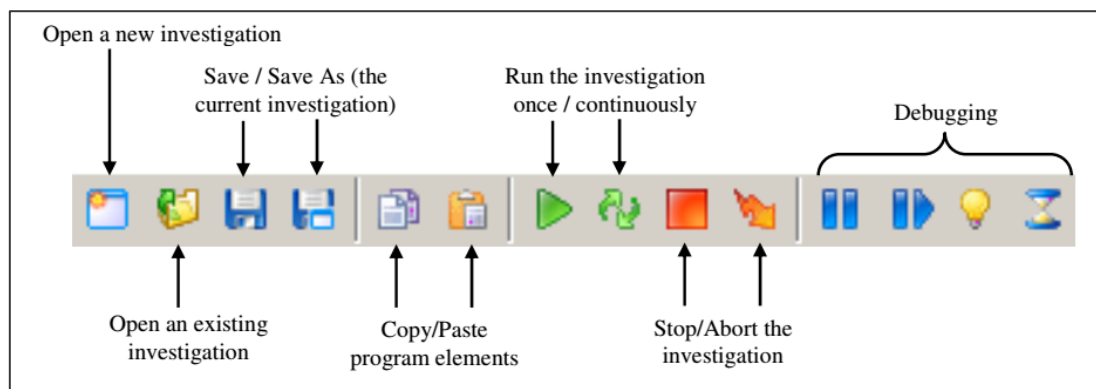
Uma investigação do Sherlock consiste em várias janelas que podem controladas pelo utilizador. Para a resolução dos exercícios deste manual é recomendado que a interface apresente as seguintes janelas.



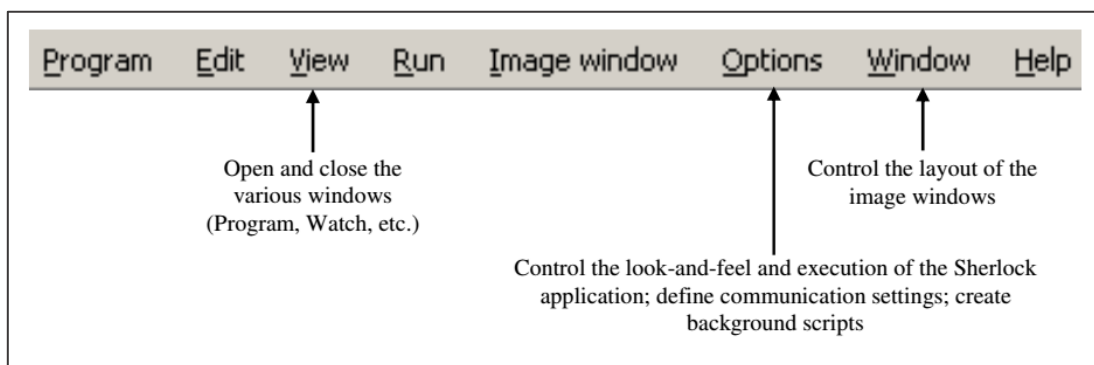
1. A janela de imagem contém a imagem e as ferramentas que extraem informação da mesma.
2. As instruções são os elementos de programação do Sherlock.
3. O programa é a sequência de instruções necessárias para extrair informação da imagem e tomar decisões.
4. As variáveis são registos de informação que o utilizador pode criar.
5. Nesta janela é possível visualizar *outputs* das ferramentas aplicadas.
6. No monitor são apresentados os erros e alertas bem como o tempo de execução da inspeção.

1. Barra de atalhos e menu principal

É na barra de atalhos que se pode aceder rapidamente a atalhos como por exemplo, a criação de nova investigação, guardar uma investigação, correr o programa, etc.

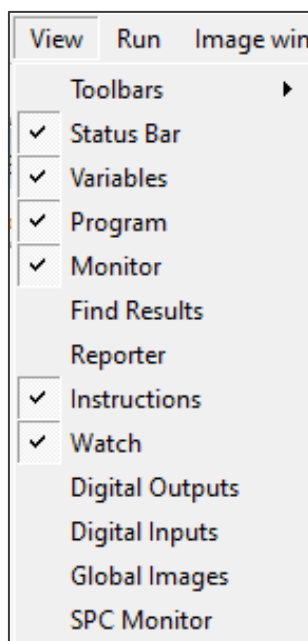


No menu principal estão disponíveis todos os comandos do Sherlock.



2. Adicionar e remover janelas de visualização

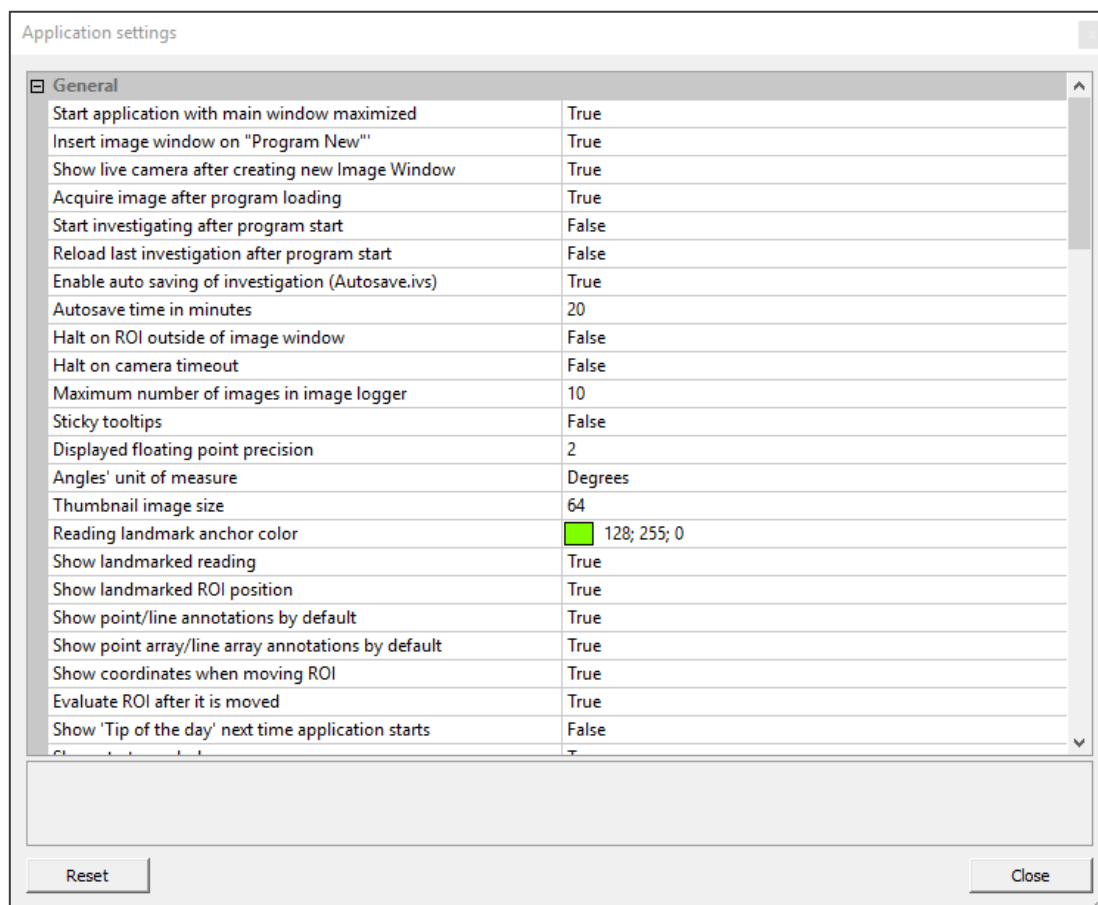
Para alterar a janelas que são visualizadas no ambiente de trabalho do Sherlock deve-se aceder ao menu *View*. As opções seleccionadas permitem a visualização das janelas necessárias á resolução dos exercícios deste manual.



3. Opções avançadas de configuração

O acesso a algumas opções avançadas de configuração do Sherlock é feito através do menu principal, carregando em **Options -> Application**.

As opções que tipicamente são alteradas são, por exemplo, o *AutoSave* para guardar automaticamente a solução, a mudança de unidades de ângulo de radianos para graus ou vice-versa, o *Log* estendido para guardar as imagens capturadas, o *AutoStart* para o arranque automático da aplicação, etc...



Exercício 1 – Análise de *Blob* e pré-processadores

O objetivo deste exercício é a deteção de defeitos em rede. Por defeito entende-se dois ou mais furos unidos ou furos com dimensões não conformes.

De seguida é demonstrada, passo a passo, a construção de um programa para deteção dos defeitos em rede plástica com recurso à análise de *blobs* e aplicação de pré-processadores.

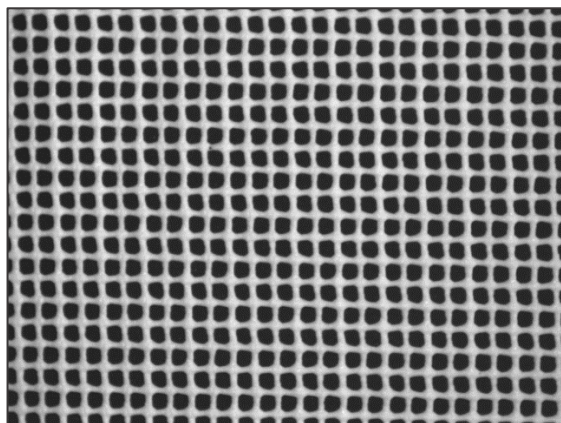


Figura 1 – Rede sem defeitos.

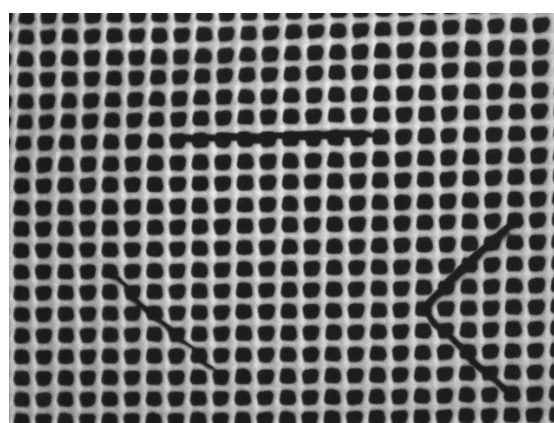
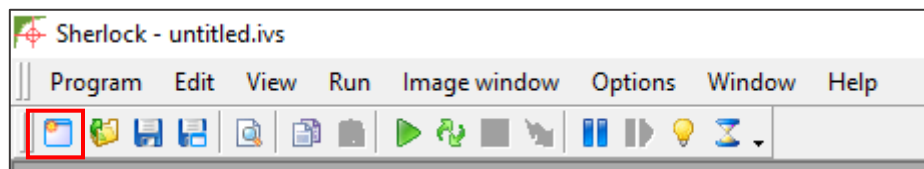


Figura 2 – Rede com defeitos.

1. Criar uma nova investigação

Para dar início a uma nova investigação deve ser criado um novo programa carregando em *New*.

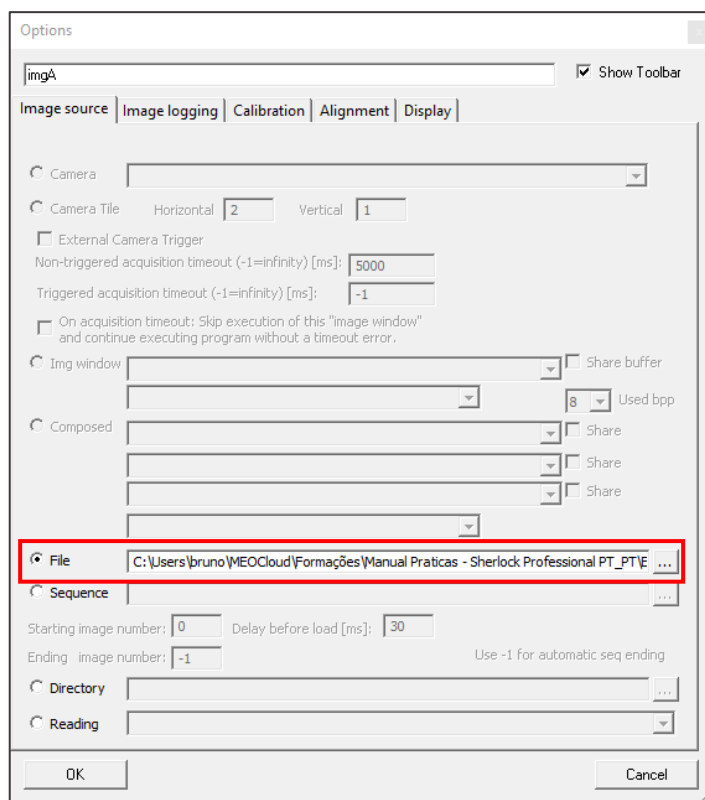
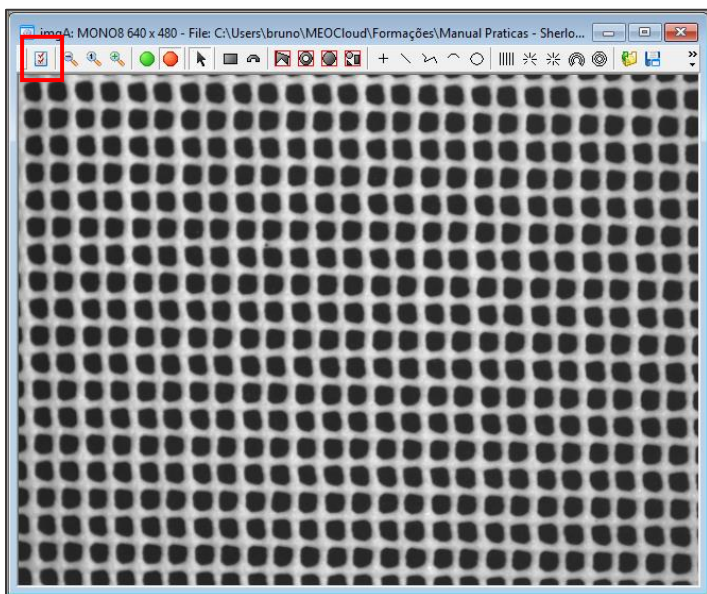


É também recomendado que a investigação seja guardada com frequência para evitar a perda do trabalho desenvolvido. É inclusive que seja ativada a função de *AutoSave* como descrito anteriormente neste manual.

2. Carregar uma imagem

Depois de criada e guardada a solução é necessário carregar uma imagem de referência na janela de imagem para se iniciar o programa. Para isso, basta aceder às opções da janela e, no separador de fontes de imagem, selecionar opção de ficheiro.

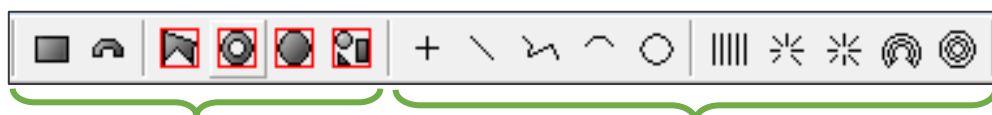
Deve de seguida ser fornecida a localização do ficheiro de imagem que se pretende carregar, neste caso irá ser carregada a imagem "*Imagem_0.bmp*" da pasta "*Ex1_Defeitos_Rede*" fornecida junto com este manual.



3. Definir a região de interesse

A região de interesse, também designada de ROI, é a delimitação de uma parte da imagem que irá ser alvo de processamento. Dentro do Sherlock existem diversos tipos de ROI's que podem ser divididas em dois tipos: ROI's de área, onde toda a região interior é processada, ou ROI's de linha, onde apenas os pixels das linhas da região são processados. Estas regiões podem ser formas geométricas simples ou formas complexas, desenhadas pelo utilizador.

No Sherlock existem as seguintes regiões de interesse:



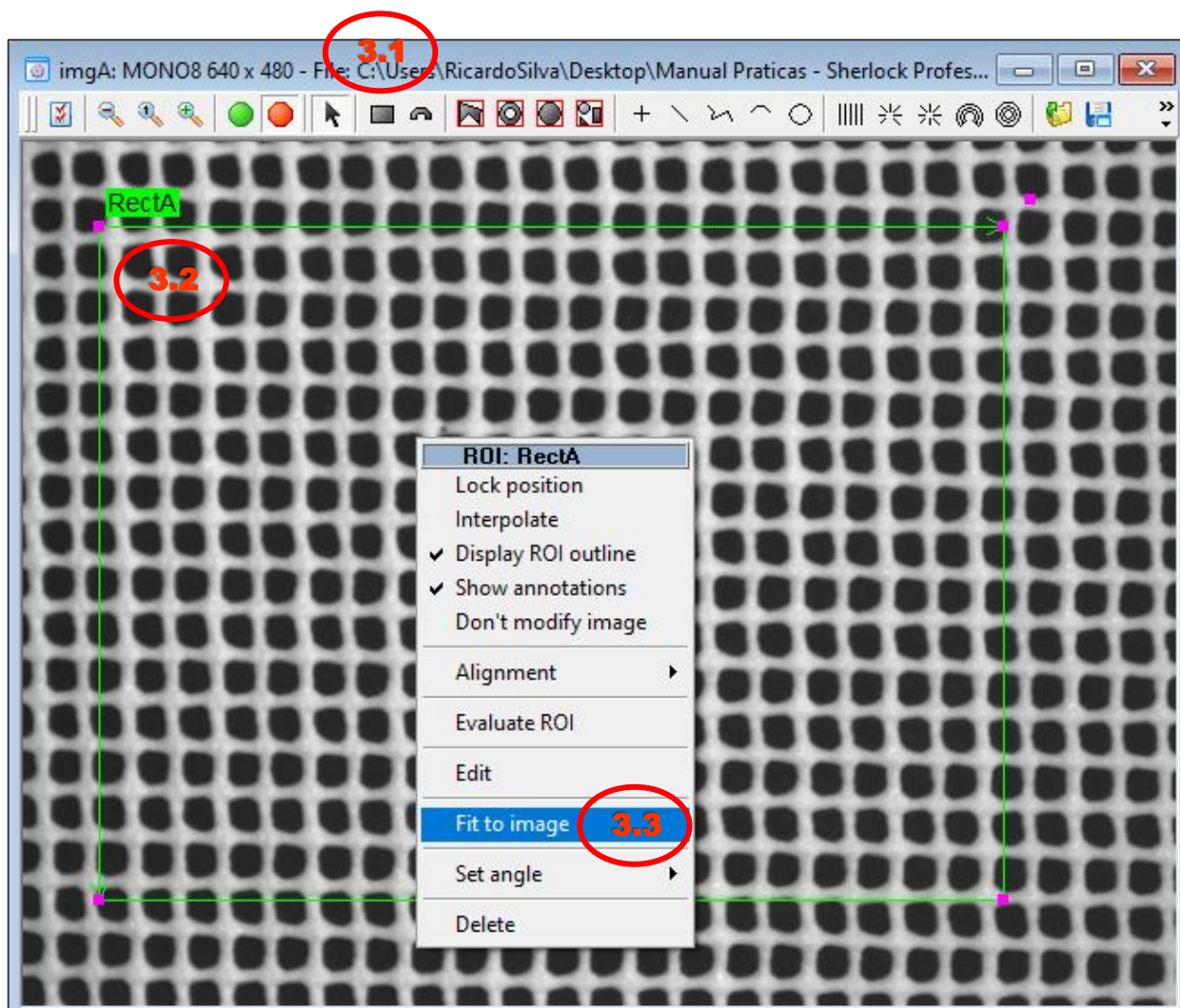
ROI's de área

ROI's de linha

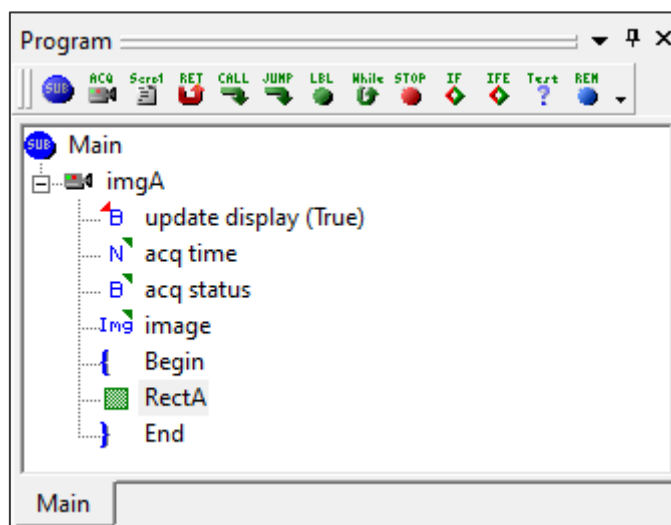
Para este exercício vai-se definir uma região retangular à volta da área que se pretende processar ou extrair informação, que neste caso será em toda a imagem.

Para a criação e ajuste da região de interesse devem ser seguidos os seguintes passos:

- 3.1. Selecionar a região do tipo retângulo.
- 3.2. Desenhar a região em cima da imagem carregando em cima da mesma.
- 3.3. De forma a garantir que toda a imagem é selecionada pela região, e que não se colocam os limites da mesma para fora da imagem, carregar com o lado direito do rato em cima da região desenhada e confirmar a opção *Fit to image* para ajustar automaticamente a região a toda a imagem.



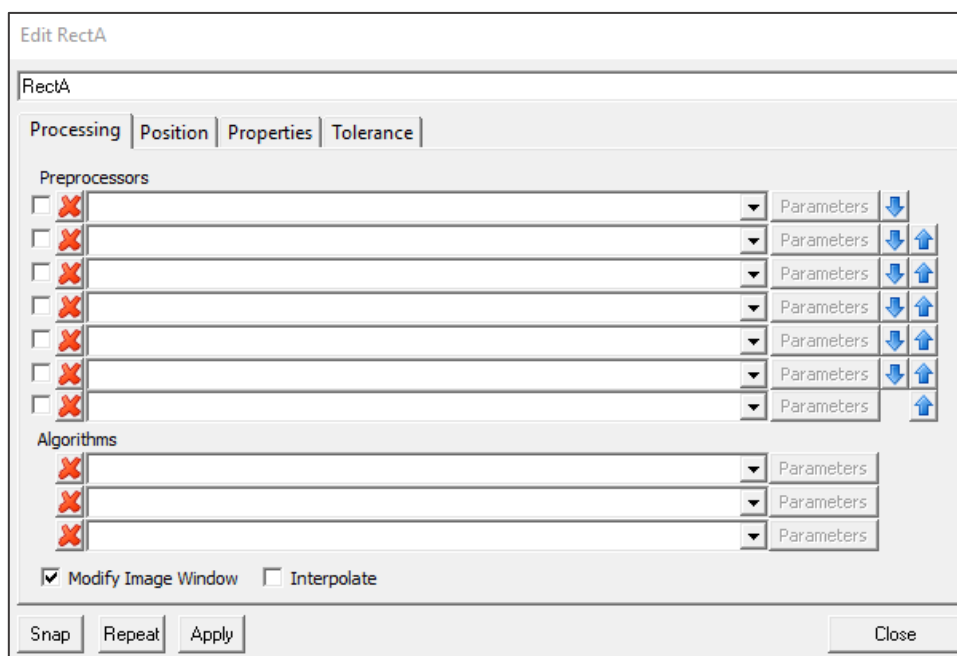
Pode-se observar na janela de programa, que à sequência pré-existente, foi adicionada, dentro da janela de imagem, uma linha correspondente a ROI criada.



4. Aplicar pré-processadores

O passo seguinte do desenvolvimento desta solução é aplicar alguns pré-processadores à região desenhada por forma a otimizar a imagem antes de aplicar algum tipo de algoritmo de processamento.

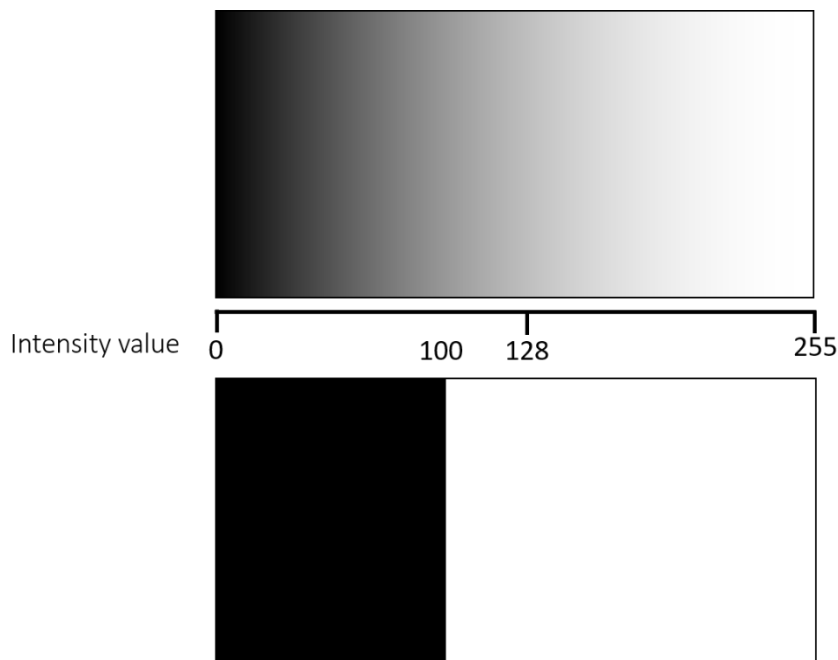
Clicando duas vezes com o botão esquerdo do rato no interior da ROI, ou na linha de programa correspondente, tem-se acesso ao menu de propriedades da região. Aqui é possível configurar e implementar pré-processadores e algoritmos, para além de configurar outras propriedades inerentes a região.



Para cada região o Sherlock permite aplicar até sete pré-processadores e três algoritmos. Os pré-processadores sobrepõem-se na sequência definida e a sua ordem pode ser alterada com as setas do lado direito de cada pré-processador selecionado.

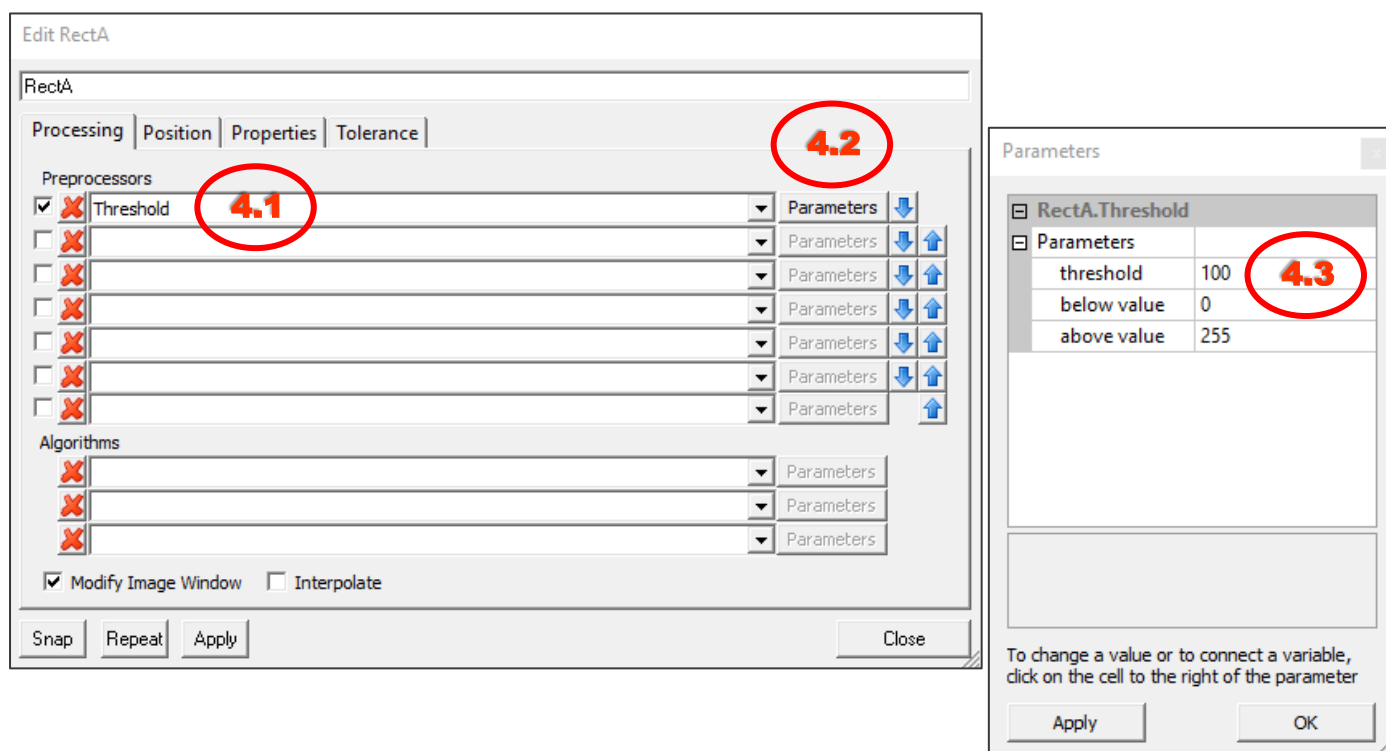
No caso desta aplicação utilizar-se-á, em primeiro lugar, o pré-processador de *Threshold* que irá binarizar a imagem, isto é, converter a imagem para dois níveis de cinzento apenas.

O valor de *threshold* define o limiar de binarização, isto é, numa imagem de 8 *bits* de profundidade, onde cada pixel contém valores de cinzento que podem variar de 0 a 255, todos os *bits* com valor abaixo do *threshold* passam a 0 e todos os que estão a cima passam para 255.



Para aplicar o pré-processador deve ser seguida a seguinte sequência:

- 4.1. Selecionar o pré- processador de *threshold*.
- 4.2. Clicar em *Parameters* para aceder aos parâmetros de configuração do pré-processador.
- 4.3. Ajustar o parâmetro do *threshold* para o valor 100.



Edit RectA

RectA

Processing | Position | Properties | Tolerance

Preprocessors

☒ Threshold **4.1** Parameters

Algorithms

☒ Modify Image Window ☐ Interpolate

Snap Repeat Apply Close

Parameters

RectA.Threshold

Parameters

threshold	100 4.3
below value	0
above value	255

To change a value or to connect a variable, click on the cell to the right of the parameter

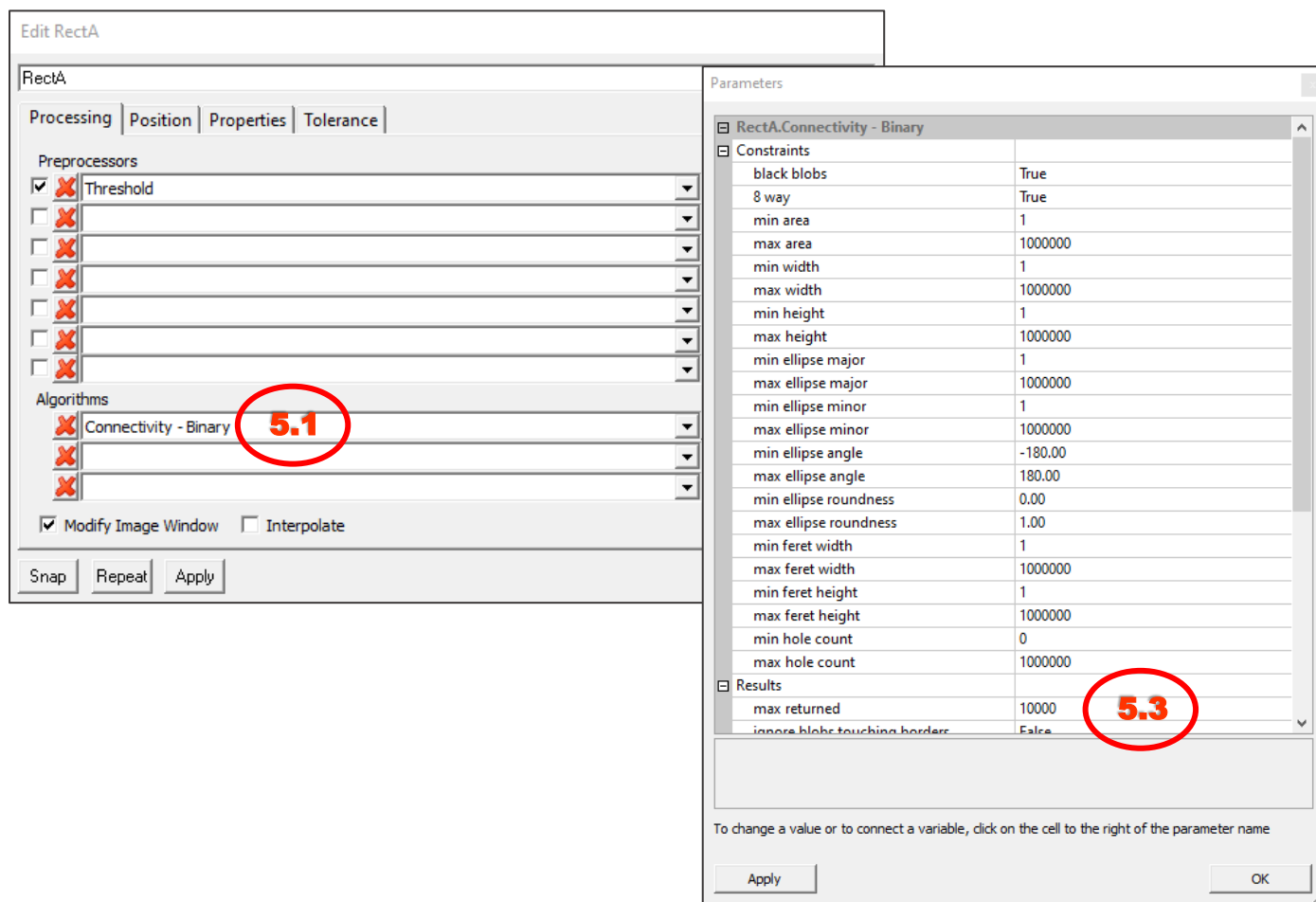
Apply OK

5. Aplicar o algoritmo

A aplicação de algoritmos funciona de forma semelhante à dos pré-processadores. Para a deteção dos defeitos na rede vai ser aplicado o algoritmo *Connectivity - Binary* que retorna todos os *blobs* dentro dos parâmetros configurados pelo utilizador.

Para aplicar o algoritmo deve ser seguida a seguinte sequência:

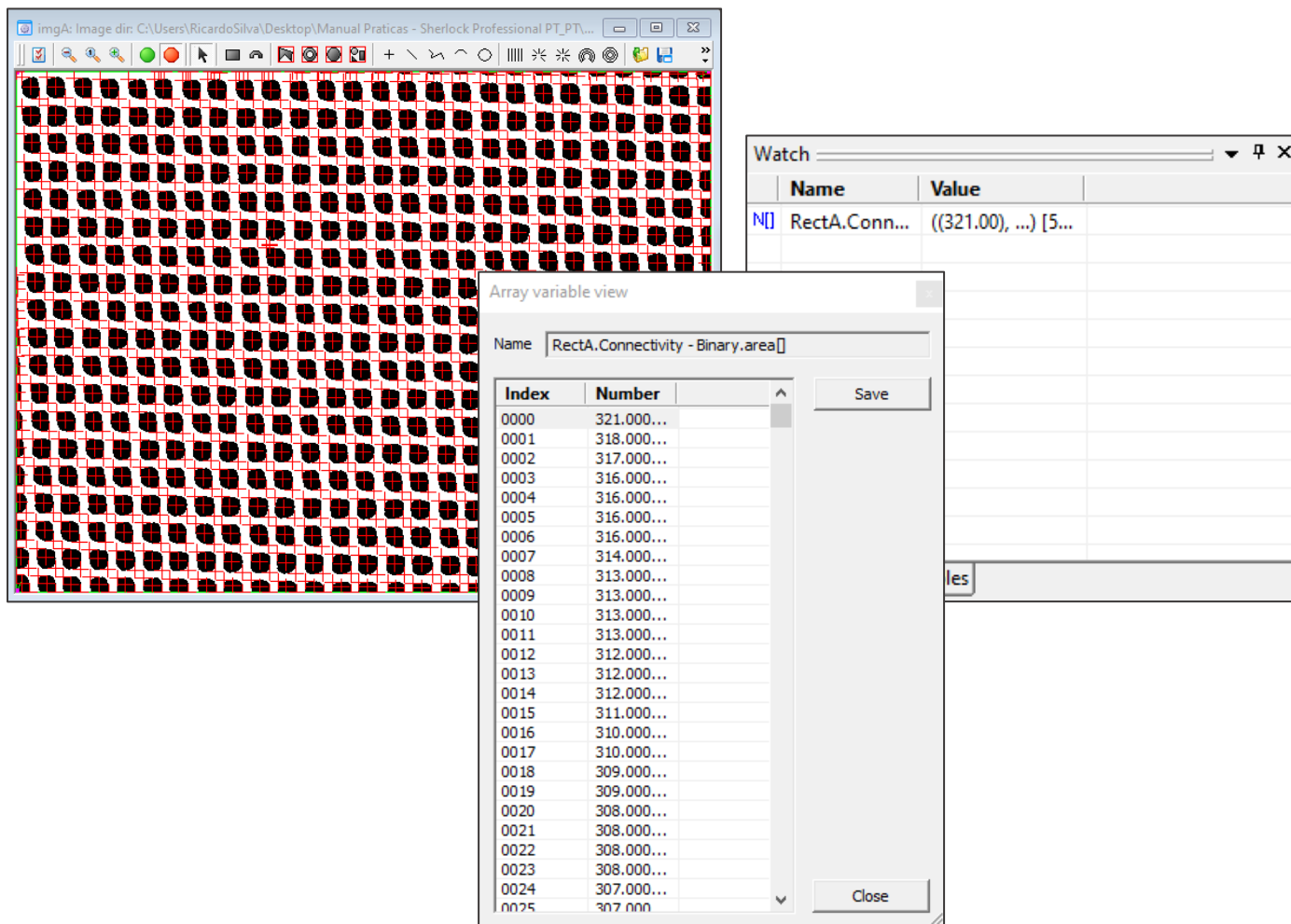
- 5.1. Selecionar o algoritmo *Connectivity - Binary*.
- 5.2. Clicar em *Parameters* para aceder aos parâmetros de configuração do algoritmo.
- 5.3. Ajustar o parâmetro *max returned* para o valor de 10000. Este valor define o número máximo de objetos retornados pela ferramenta.



Para cada algoritmo adicionado na ROI, é também adicionada uma linha ao programa na janela de programas dentro da respetiva ROI. Neste caso, clicando sobre o algoritmo *Connectivity - Binary*, tem-se acesso às saídas do algoritmo, que depois podem ser utilizadas para tomar uma decisão.

Outra forma de visualizar as saídas é na janela *Watch*. Para visualizar as saídas do algoritmo nesta janela basta arrastar a saída pretendida para dentro da janela. De seguida basta carregar duas vezes com o lado esquerdo do rato em cima da saída para se ter aos valores da mesma.

Neste caso, com a parametrização feita e com a maioria dos valores por defeito, obteve-se o seguinte resultado para a área dos objetos detetados:



Uma vez selecionada a saída de área pode-se observar o valor de área de cada objeto. O objetivo é verificar objetos fora das dimensões médias dos furos, o que em termos de imagem se traduzirá em objetos de área maior, isto porque, quando existe um corte, por exemplo, em termos de imagem há uma ligação entre os dois furos formando um único *blob* com maior área.

Tendo por base esta representação dos defeitos, e analisando a media de valores de área retornados com a imagem da rede OK, pode-se parametrizar o algoritmo para procurar apenas *blobs* quando existe defeito, alterando os limites de área mínima e máxima.

Nesse sentido, voltando aos parâmetros do algoritmo, devem ser ajustados o seguinte parâmetro:

- 5.4. Ajustar o parâmetro *min area* para o valor de 350. Este valor define o número mínimo da área dos objetos retornados pela ferramenta.

Parameters

RectA.Connectivity - Binary

Constraints

black blobs	True
8 way	True
min area	350
max area	1000000
min width	1
max width	1000000
min height	1
max height	1000000
min ellipse major	1
max ellipse major	1000000
min ellipse minor	1
max ellipse minor	1000000
min ellipse angle	-180.00
max ellipse angle	180.00
min ellipse roundness	0.00
max ellipse roundness	1.00
min feret width	1
max feret width	1000000
min feret height	1
max feret height	1000000
min hole count	0
max hole count	1000000

Results

max returned	10000
ignore blobs touching borders	False

To change a value or to connect a variable, click on the cell to the right of the parameter name

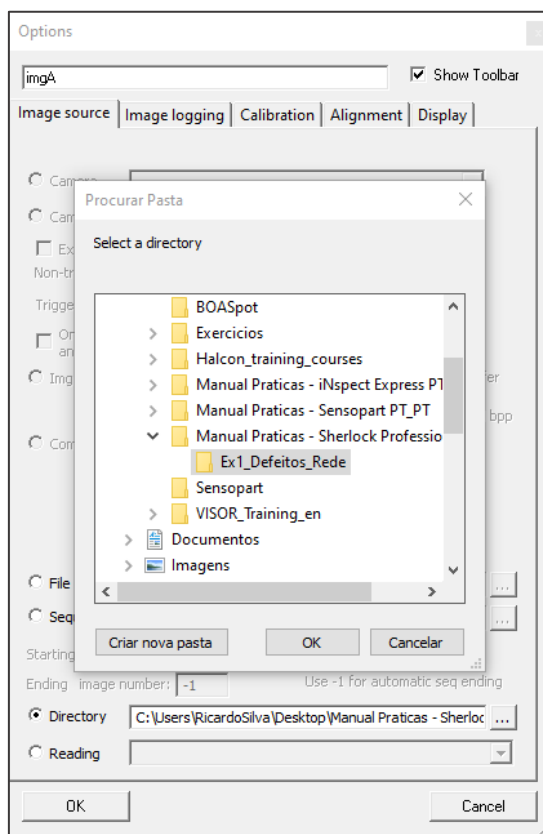
Apply

OK

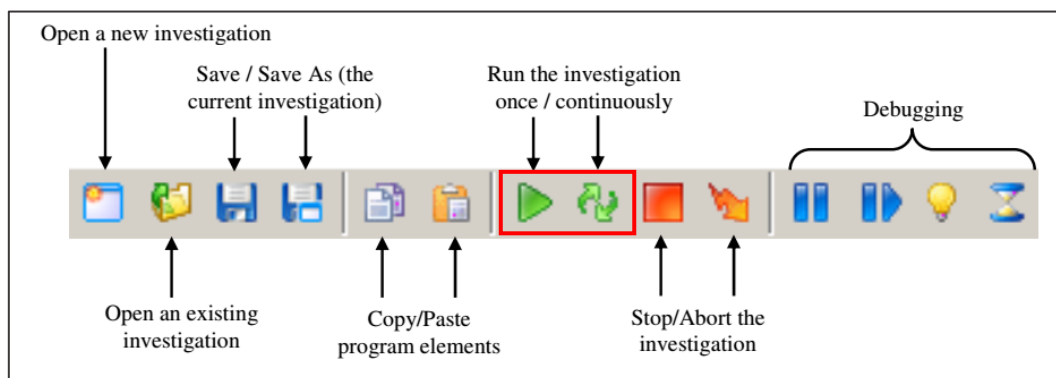
Para efeitos de visualização, é útil também alterar dois outros parâmetros do algoritmo: colocar as *bounding boxes* ativas e os *labels* ativos também.

6. Carregar um lote de imagens

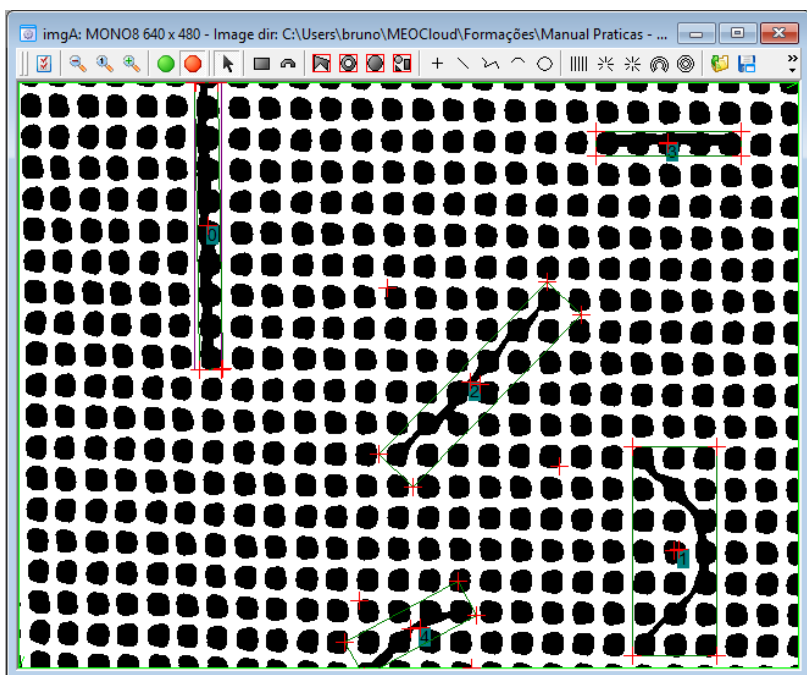
Para testar a funcionalidade do programa é necessário testá-lo em várias imagens. O Sherlock também permite carregar um lote de imagens e para fazê-lo basta aceder as opções da janela de imagem e selecionar a opção *Directory* em vez de *File*, definido de seguida o caminho para a pasta onde estão todas as imagens que se pretendem testar.



Depois de carregado o lote de imagens, estas devem ser inspecionadas para verificar que o programa está a funcionar corretamente e são detetados todos os defeitos. As imagens podem ser percorridas uma a uma, clicando em *Run Once*, ou de forma contínua, em *Run Continuously*.



O resultado que se deve obter deverá ser algo semelhante a isto:



O propósito de correr o lote de imagens (*OK* e *NOT OK*) é verificar a estabilidade do algoritmo com as variações que se podem encontrar, e assim poder ajustar os parâmetros que forem necessários.

Exercício 2 – Leitura de códigos e alinhamento

O objetivo deste exercício é a leitura do código 2D presente nas placas das imagens abaixo com a particularidade que os códigos variam a sua posição e orientação com grande amplitude entre imagens.

De seguida é demonstrada, passo a passo, a construção de um programa para a leitura dos códigos com recurso à ferramenta *Barcode QR* e aplicação de alinhamento para ajustar a região de interesse à posição variável da placa.

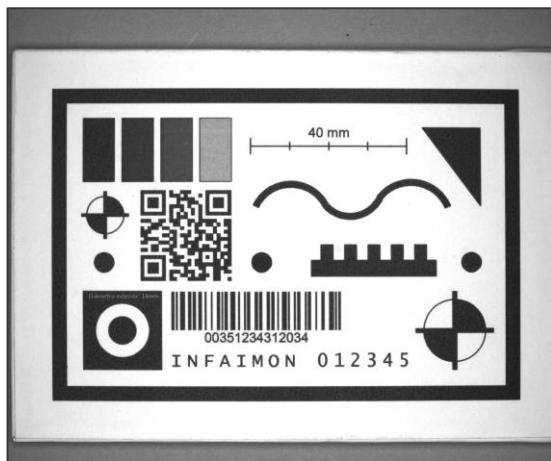


Figura 3 – Placa em posição normal.



Figura 4 – Placa em posição invertida.

1. Iniciação da investigação

Para dar início a uma nova investigação devem ser seguidos os passos descritos no Exercício 1.

2. Carregar imagens

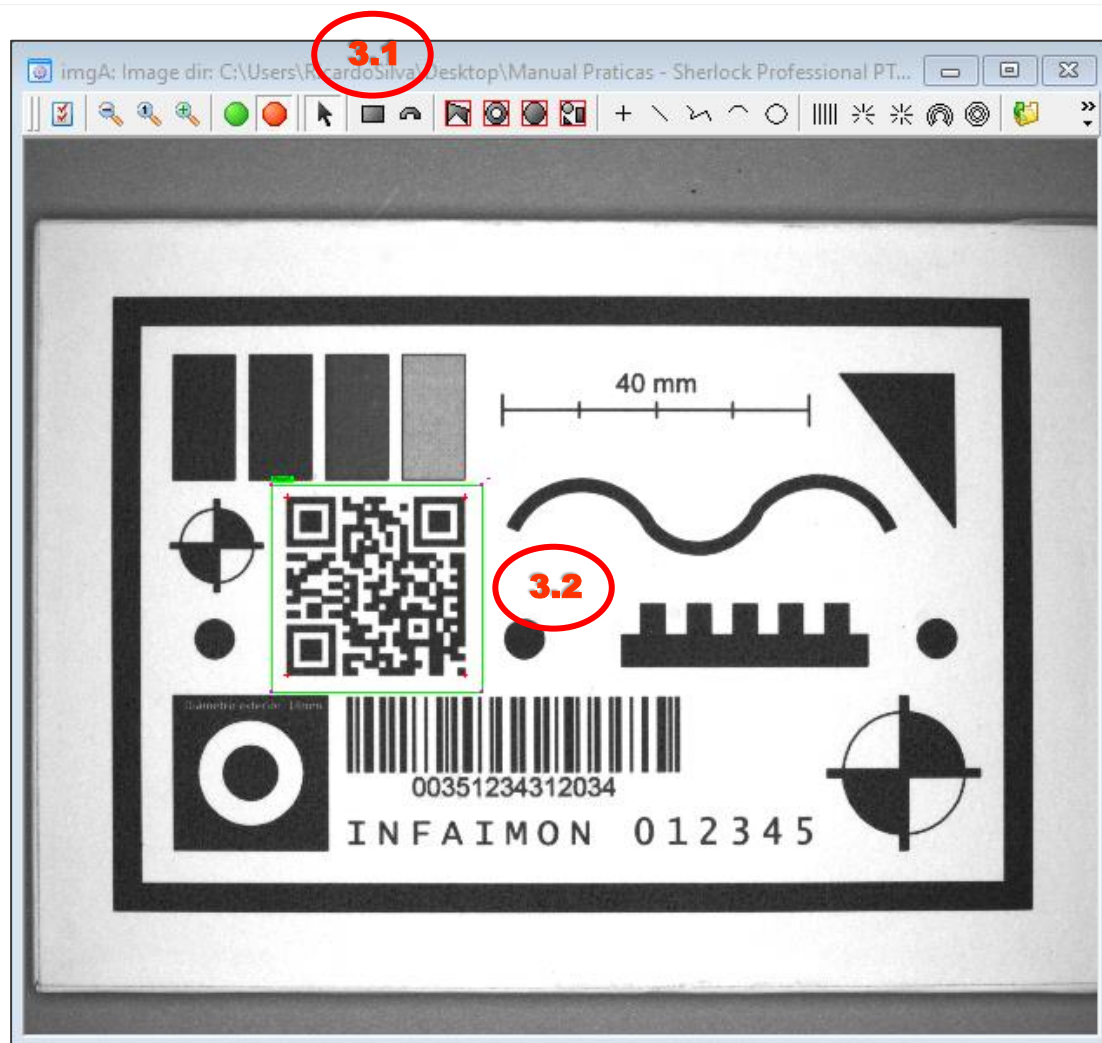
Depois de criada e guardada a solução é necessário carregar o lote das imagens a inspecionar selecionando a opção de aquisição *Directory*, cujos passos são também explicados no Exercício 1. Deve de seguida ser fornecida a localização dos ficheiros de imagem que se pretendem carregar, neste caso será pasta “Ex2_Codigos_Alinhamento” fornecida junto com este manual.

3. Aplicação da ferramenta de leitura de códigos 2D

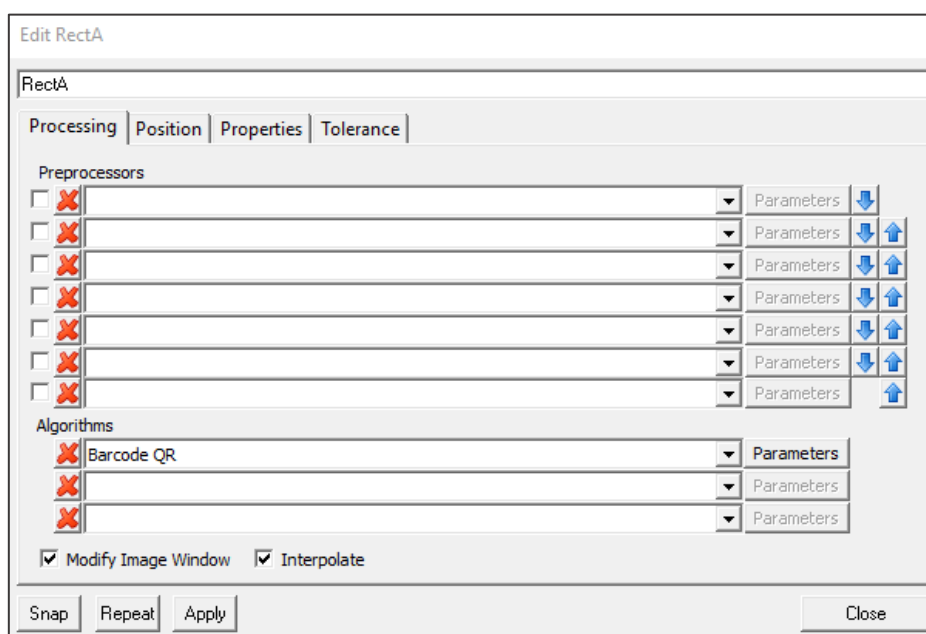
Antes de aplicar a ferramenta de leitura de códigos 2D é necessário definir uma região de interesse retangular à volta da área onde se encontra o código.

Para a criação e ajuste da região de interesse devem ser seguidos os seguintes passos:

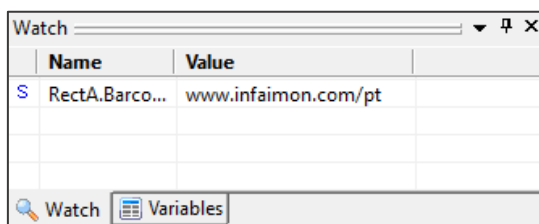
- 3.1. Selecionar a região do tipo retângulo.
- 3.2. Desenhar a região em cima da imagem carregando em cima da mesma.



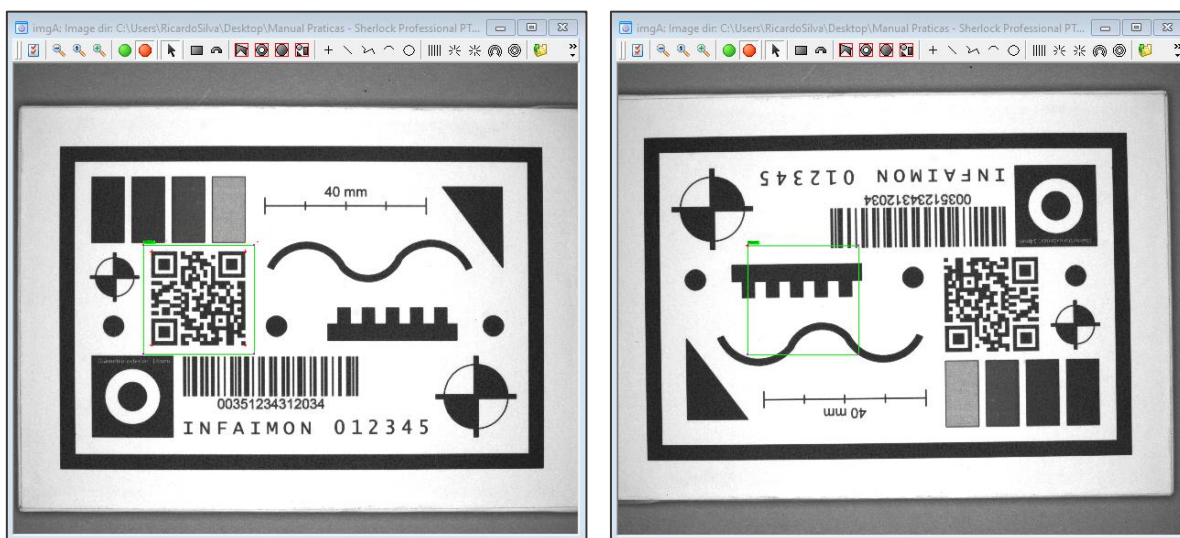
De seguida deve-se aceder à janela de configuração da ROI desenhada, clicando duas vezes com o botão esquerdo do rato, e seleccionar o algoritmo *Barcode QR*. Neste caso não é necessário parametrizar o algoritmo uma vez que as definições padrão deste algoritmo são adequadas para a leitura do código 2D.



Para visualizar a leitura do código 2D basta arrastar a saída *String* do algoritmo para a janela *Watch*.



Avançando na sequência de imagens verifica-se que não se conseguem ler todos os códigos pois em várias imagens os códigos estão fora da região de interesse desenhada.



4. Criar um alinhamento

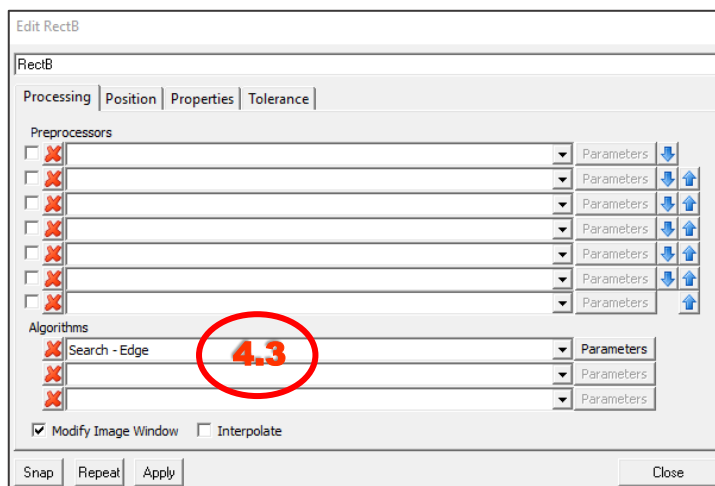
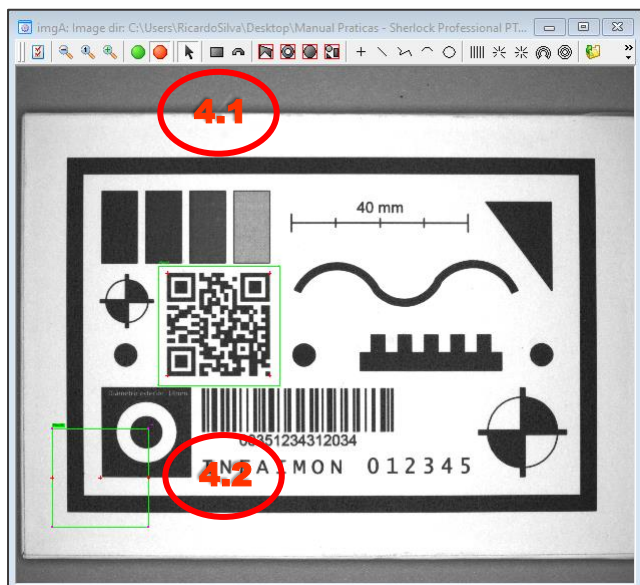
O problema anterior podia ser resolvido alargando a ROI a toda a imagem, no entanto, para algoritmos mais complexos e limitações de tempo de processamento, esta solução não era a ideal. Assim, a forma mais correta de resolver o problema é criar um alinhamento a partir do qual a ROI para leitura do código vai ser posicionada.

Para criar um alinhamento deve ser utilizada uma característica única da imagem e definir a posição da região de procura de código em relação a essa característica. Neste caso vai-se utilizar o canto inferior esquerdo da imagem e a uma ferramenta de *match* para localizar esta característica na imagem.

Uma nota importante é que as ferramentas devem ser posicionadas na mesma imagem que é feita a procura do alinhamento. Deve então ser percorrida a sequência de imagens até chegar á primeira antes de se iniciar a criação do alinhamento.

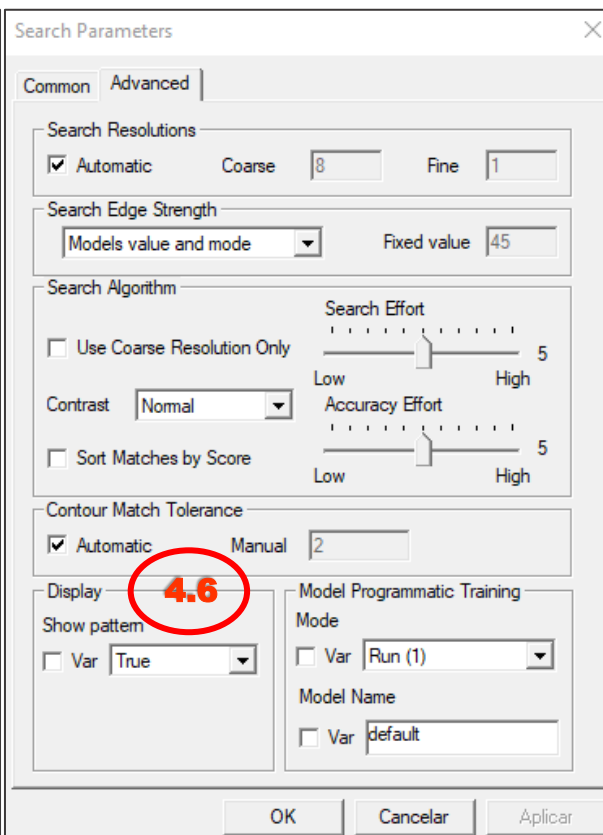
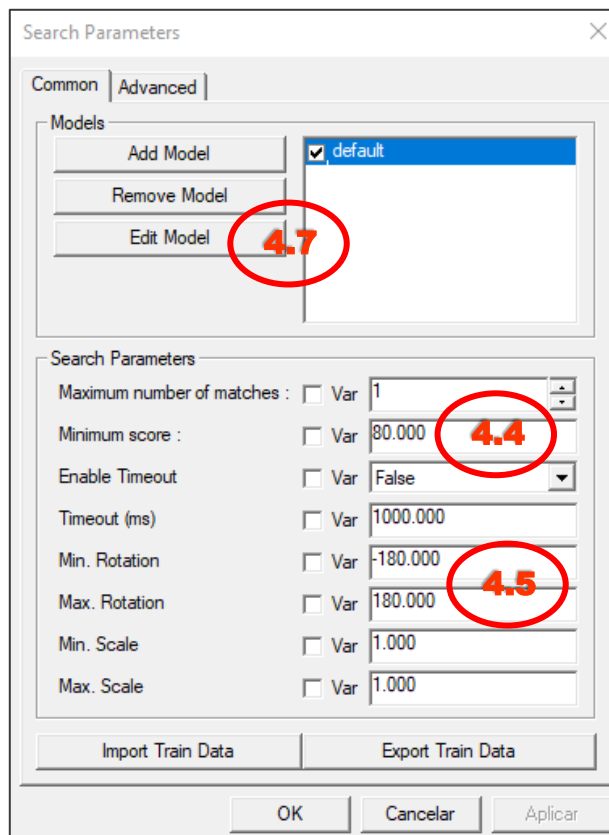
Para criação do alinhamento vai ser necessário então desenhar a região de interesse e aplicar o algoritmo, para tal devem ser seguidos os seguintes passos:

- 4.1. Selecionar a região do tipo retângulo.
- 4.2. Desenhar a região de forma a apanhar o canto inferior esquerdo da imagem.
- 4.3. Aplicar o algoritmo *Search – Edge* de procura de padrão por arestas.

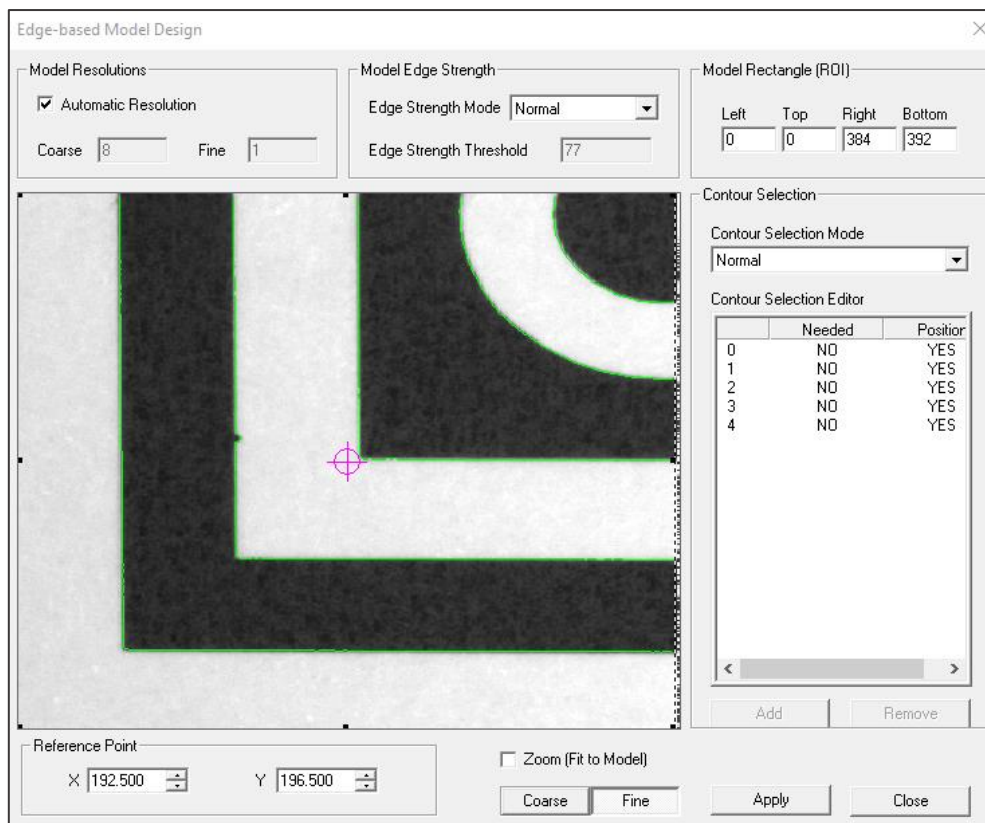


O próximo passo é configurar os parâmetros do algoritmo para encontrar o padrão em toda a imagem. Para tal importa configurar os seguintes parâmetros:

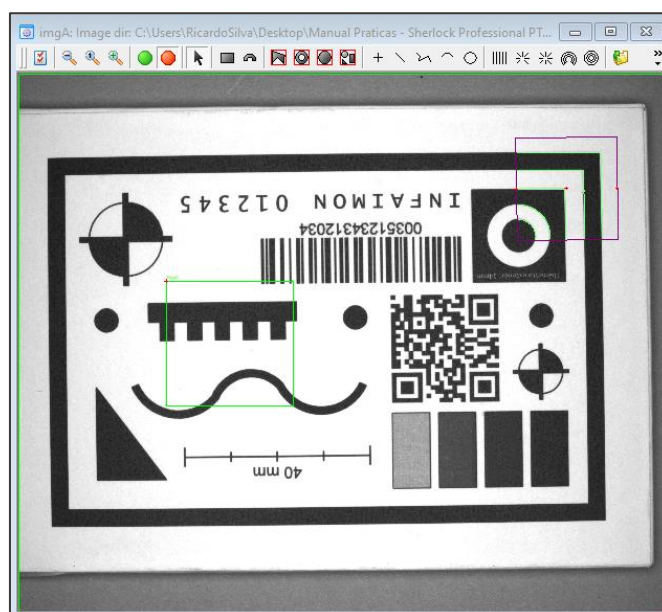
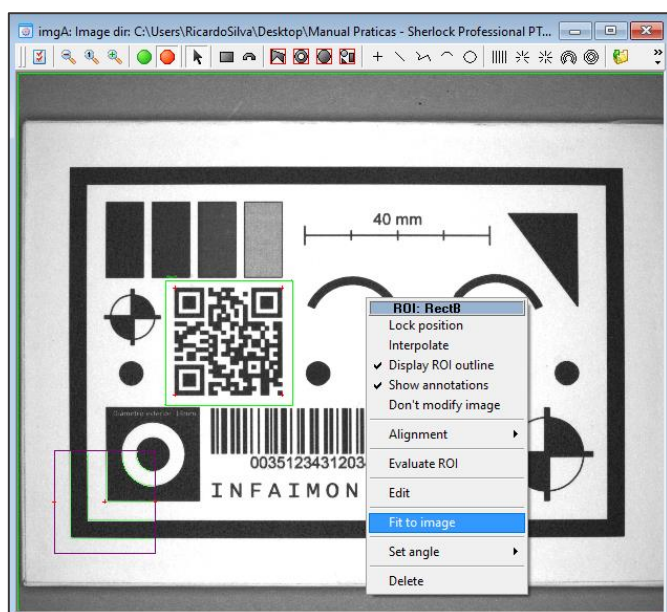
- 4.4. Aumentar o *score* para 80 para garantir que o padrão encontrado é o pretendido.
- 4.5. Definir os limites de rotação de -180° a $+180^\circ$ para que o padrão possa ser encontrado em qualquer orientação.
- 4.6. Ativar a visualização do padrão encontrado para uma interpretação mais visual dos resultados.
- 4.7. Aceder à edição do modelo.



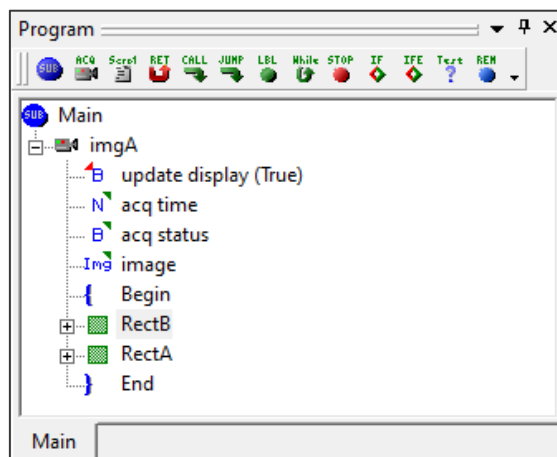
Uma vez na janela de edição do modelo podem ser adicionadas ou retiradas arestas às identificadas a verde na janela. Neste caso, como se tem contraste suficiente no padrão, as arestas estão bem definidas e não é necessário nem adicionar nem remover nenhuma aresta ao padrão a procurar.



Agora, resta ajustar a região de procura a toda a imagem, como explicado no Exercício 1, e verificar que o padrão (representado pelo quadrado roxo), é encontrado em todas as imagens carregadas.

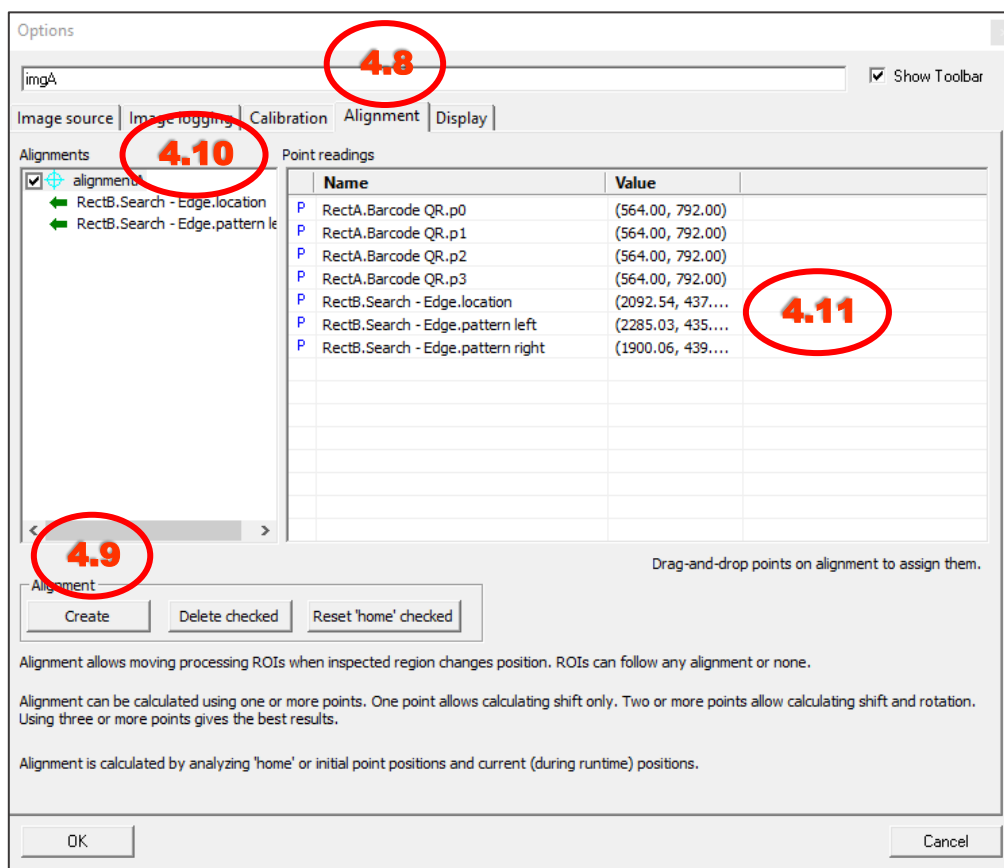


Antes de criar o alinhamento é necessário reorganizar a sequência do programa pois a execução do mesmo é sequencial. Assim, a ROI *RectB*, que vai fornecer os pontos para alinhamento, tem de ser executada primeiro do que a ROI *RectA*, que vai ser alinhada. Para realizar esta alteração basta arrastar o item *RectB* para cima do *RectA* na janela de programa de forma a que a sequência fique idêntica à da imagem a baixo.



Por fim, resta a criação do alinhamento. Para criar o alinhamento é necessário abrir as opções da janela de imagem e seguir os seguintes passos:

- 4.8. Aceder ao separador de alinhamento.
- 4.9. Criar um novo alinhamento.
- 4.10. Selecionar o alinhamento criado para adicionar os pontos de alinhamento.
- 4.11. Arrastar dois dos pontos criados pela ferramenta *Search – Edge* para dentro do alinhamento de forma a criar uma referência de posição e orientação.



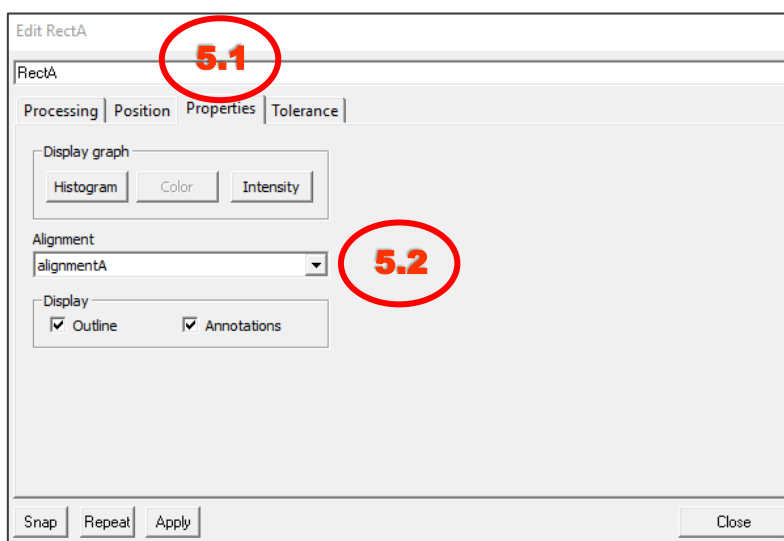
5. Aplicar o alinhamento

No Sherlock o alinhamento é aplicado às regiões de interesse, ou seja, é necessário configurar cada região para que a mesma seja orientada por o alinhamento pretendido.

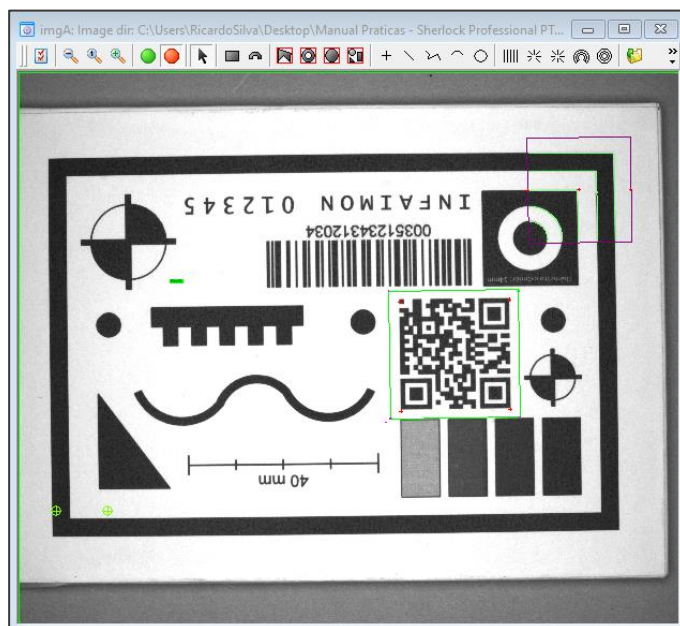
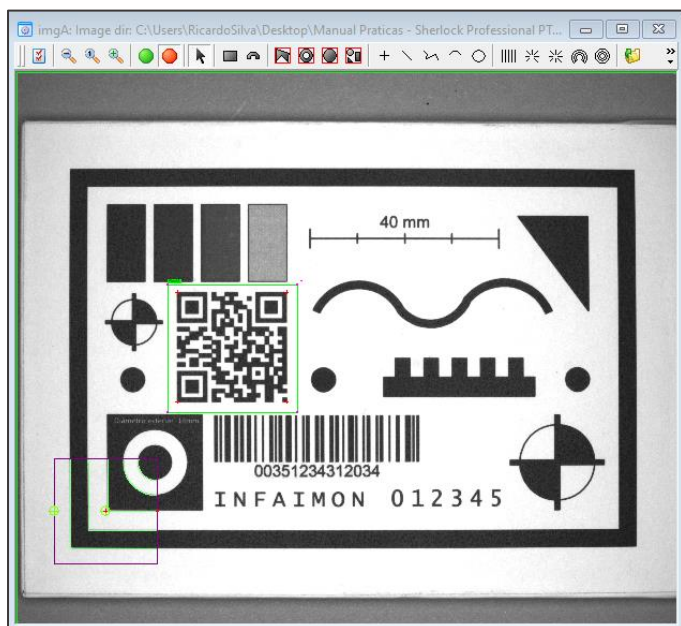
A região de leitura de códigos 2D, *RectA*, tem então que ser configurada para que siga o alinhamento criado no ponto anterior. Para essa configuração deve-se aceder à janela da edição da região clicando duas vezes com o lado esquerdo do rato sobre a mesma.

A aplicação do alinhamento é bastante simples, basta seguir estes dois pontos:

- 5.1. Aceder ao separador de propriedades da região.
- 5.2. Escolher o alinhamento criado e que vai ser utilizado para posicionar a região.



Antes de dar o exercício como concluído, deve-se verificar que a região de leitura de códigos segue o alinhamento criado e, conseqüentemente, lê corretamente o código 2D presente em todas as imagens.



Exercício 3 – Medição de distâncias e calibração do sistema

O objetivo deste exercício é a medição das aberturas laterais das peças plásticas das imagens a baixo de forma a garantir que estas não apresentam defeitos.

De seguida é demonstrada, passo a passo, a construção de um programa para a medição das aberturas laterais das peças com recurso à ferramenta *Outside Caliper* e aplicação de calibração do sistema para extração dos valores da abertura em unidades reais.

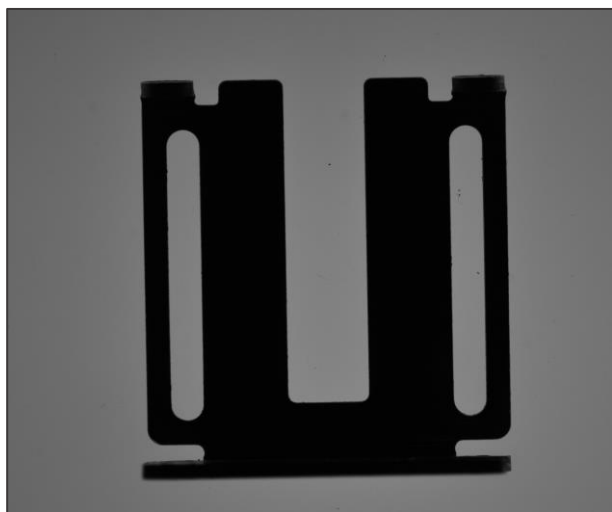


Figura 5 – Peça plástica sem defeito nas aberturas laterais.



Figura 6 – Peça plástica com defeito nas aberturas laterais.

1. Iniciação da investigação

Para dar início a uma nova investigação devem ser seguidos os passos descritos no Exercício 1.

2. Carregar imagens

Depois de criada e guardada a solução é necessário carregar o lote das imagens a inspecionar selecionando a opção de aquisição *Directory*, cujos passos são também explicados no Exercício 1. Deve de seguida ser fornecida a localização dos ficheiros de imagem que se pretendem carregar, neste caso será pasta “Ex3_Medicoes_Calibracao” fornecida junto com este manual.

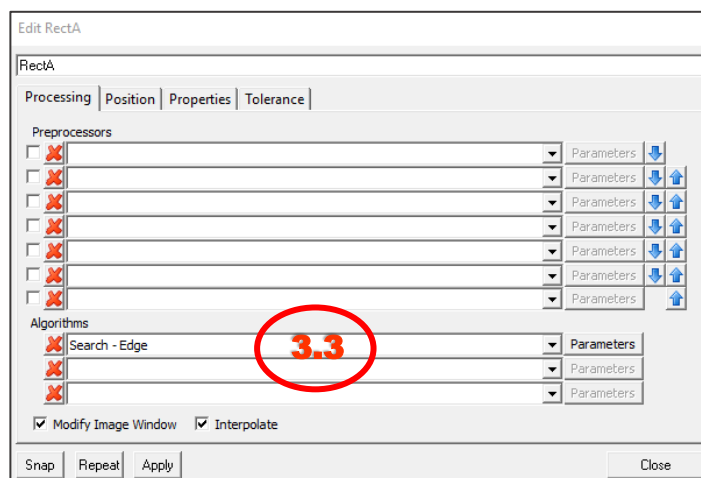
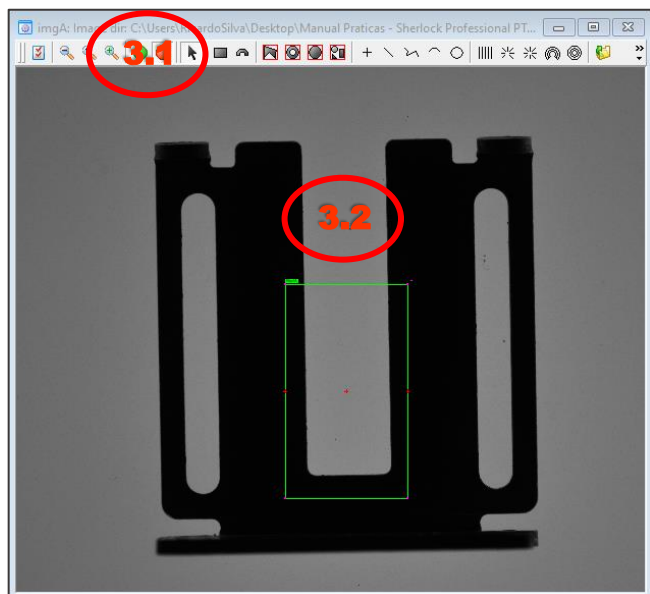
3. Criar um alinhamento inicial

Como existe variação da posição da característica que se pretende avaliar, à semelhança do Exercício 2, deve ser criado um alinhamento inicial para as ferramentas de medição.

Para criar um alinhamento deve ser utilizada uma característica única da imagem e neste caso vai-se utilizar a abertura central das peças e a uma ferramenta de *match* para localizar esta característica na imagem.

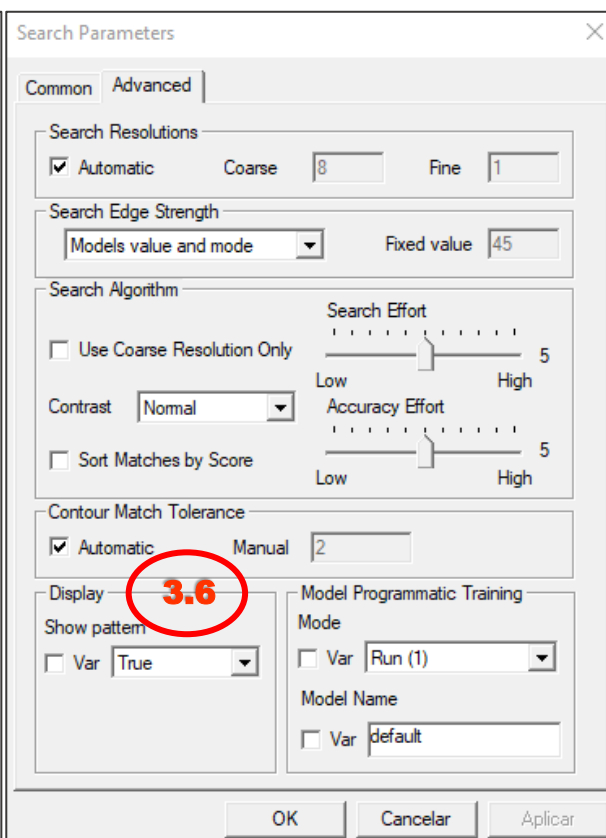
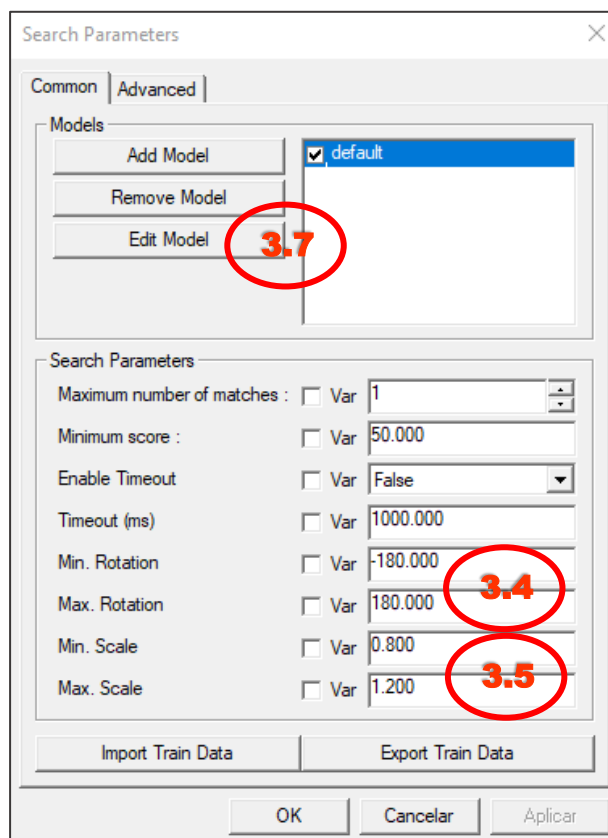
Para criação do alinhamento vai ser necessário então desenhar a região de interesse e aplicar o algoritmo, para tal devem ser seguidos os seguintes passos:

- 3.1. Selecionar a região do tipo retângulo.
- 3.2. Desenhar a região de forma a apanhar cerca de metade da abertura central da peça.
- 3.3. Aplicar o algoritmo *Search – Edge* de procura de padrão por arestas.

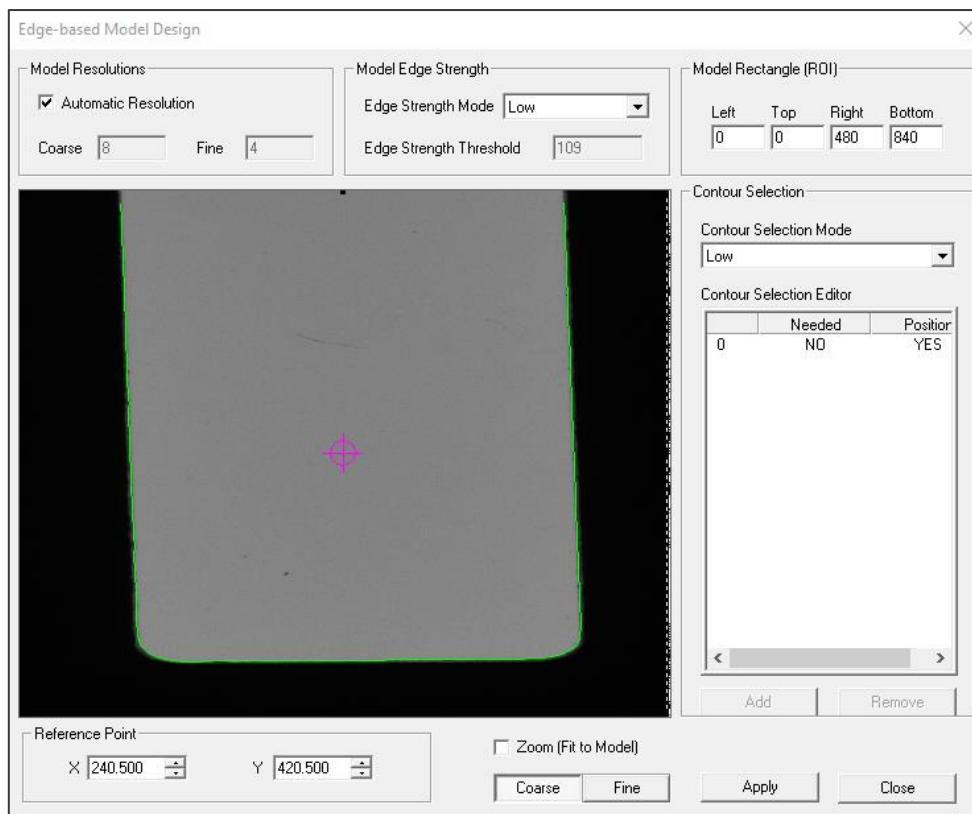


O próximo passo é configurar os parâmetros do algoritmo para encontrar o padrão em toda a imagem. Para tal importa configurar os seguintes parâmetros:

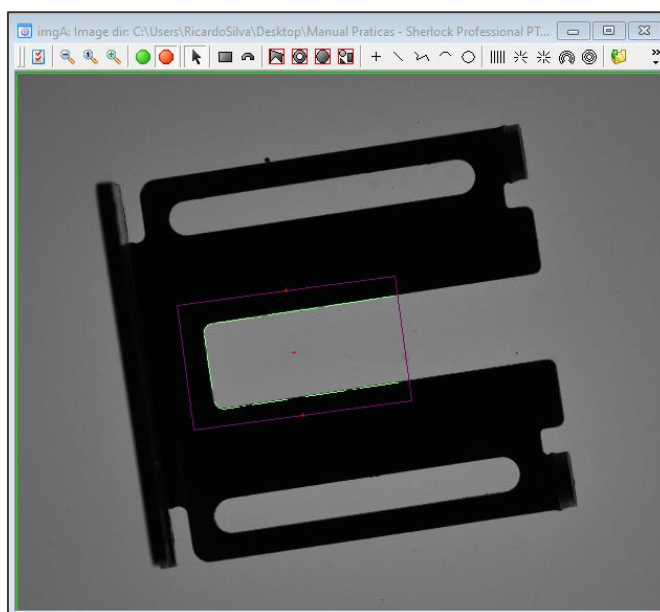
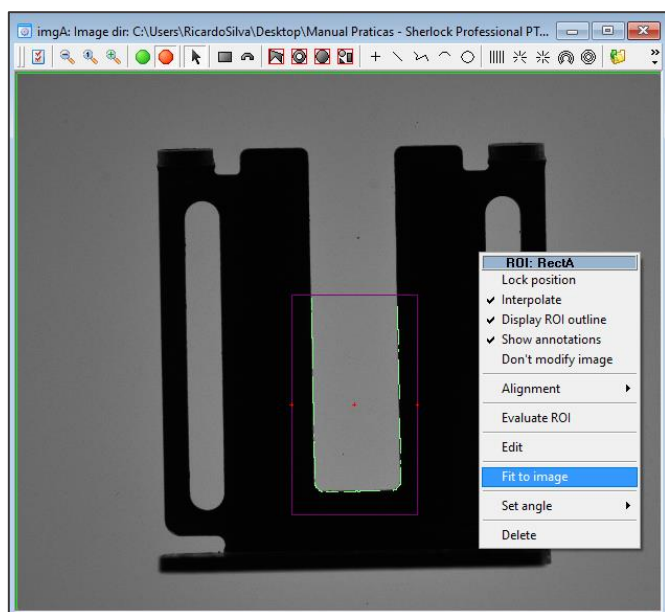
- 3.4. Definir os limites de rotação de -180° a $+180^\circ$ para que o padrão possa ser encontrado em qualquer orientação.
- 3.5. Definir os limites de escala entre 0.8 e 1.2 de forma a permitir alguma variação da forma da abertura central.
- 3.6. Ativar a visualização do padrão encontrado para uma interpretação mais visual dos resultados.
- 3.7. Aceder à edição do modelo.



Uma vez na janela de edição do modelo podem ser adicionadas ou retiradas arestas às identificadas a verde na janela. Neste caso, como se tem contraste suficiente no padrão e se tem apenas uma aresta não é necessário fazer qualquer tipo de alteração.

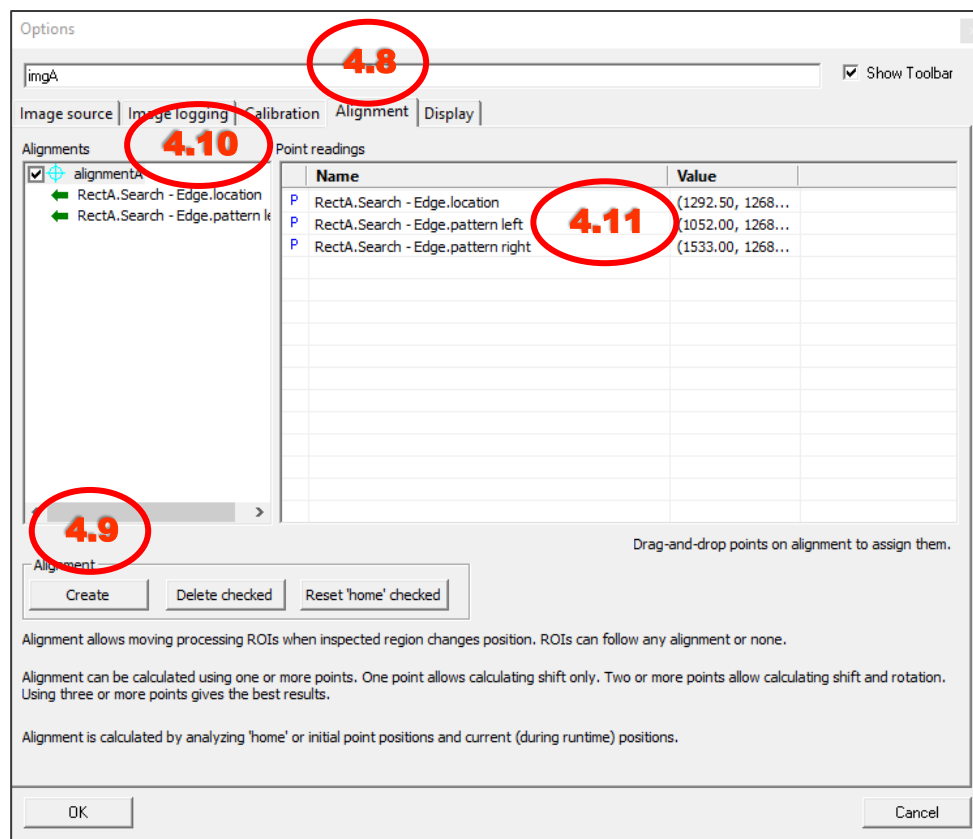


Agora, resta ajustar a região de procura a toda a imagem, como explicado no Exercício 1, e verificar que o padrão (representado pelo quadrado roxo), é encontrado em todas as imagens carregadas.



Por fim, resta a criação do alinhamento. Para criar o alinhamento é necessário abrir as opções da janela de imagem e seguir os seguintes passos:

- 3.8. Aceder ao separador de alinhamento.
- 3.9. Criar um novo alinhamento.
- 3.10. Selecionar o alinhamento criado para adicionar os pontos de alinhamento.
- 3.11. Arrastar dois dos pontos criados pela ferramenta *Search – Edge* para dentro do alinhamento de forma a criar uma referência de posição e orientação.



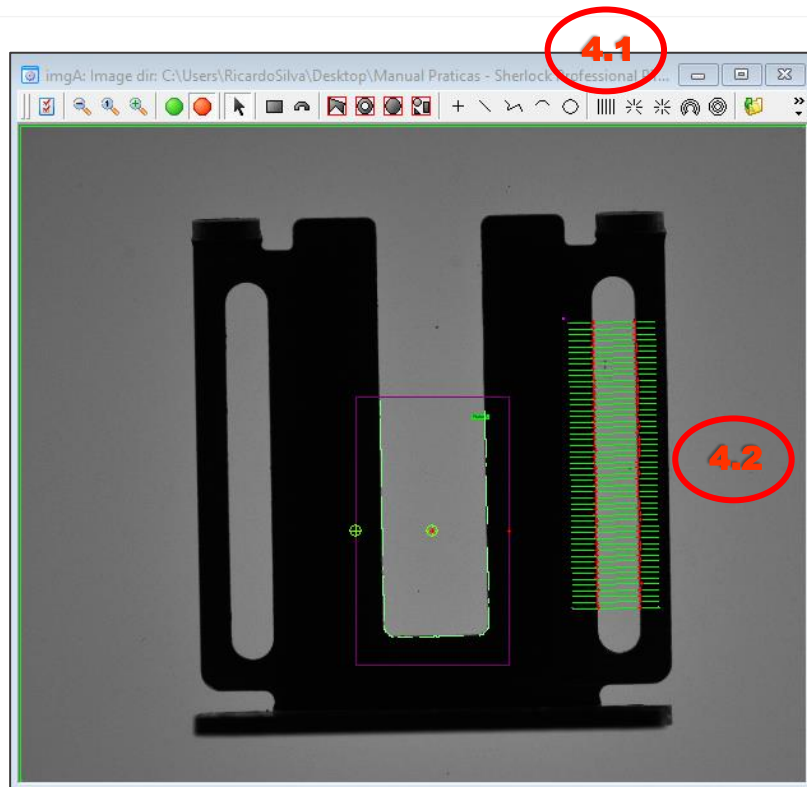
4. Aplicação de ferramenta para medição das aberturas da peça

Tal como alertado no Exercício 2, importa regressar à imagem onde foi criado o alinhamento antes de se aplicar mais ferramentas.

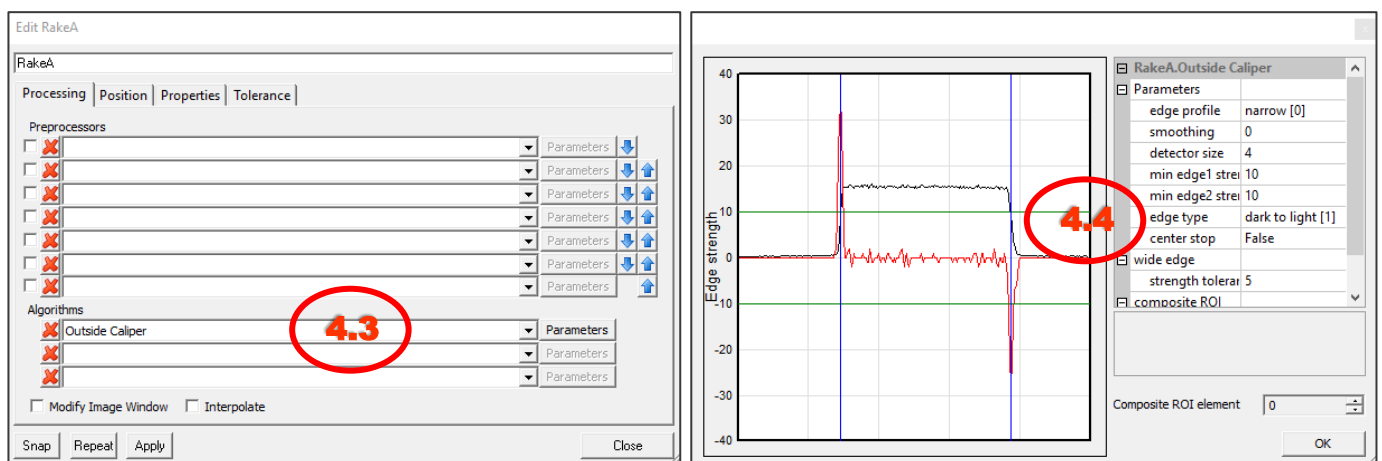
Antes de aplicar a ferramenta de medição é necessário definir uma região de interesse do tipo *Rake* que abranja toda a abertura lateral da peça plástica.

Para a criação e ajuste da região de interesse devem ser seguidos os seguintes passos:

- 4.1. Selecionar a região do tipo *Rake*.
- 4.2. Desenhar a região na zona da abertura direita da peça plástica.

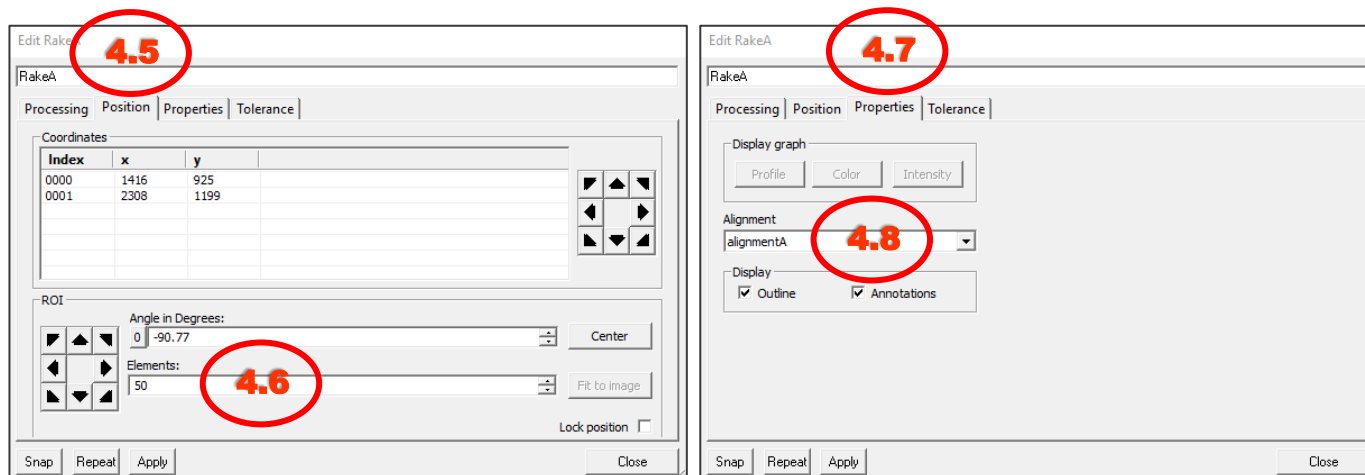


De seguida deve-se aceder à janela de configuração da ROI desenhada, clicando duas vezes com o botão esquerdo do rato, e selecionar o algoritmo *Outside Caliper* (4.3). A única parametrização que é necessário fazer é alterar o tipo de transição de procura. Como o algoritmo procura a primeira transição de fora para dentro das linhas da região, essa transição deve ser de escuro para claro (4.4).



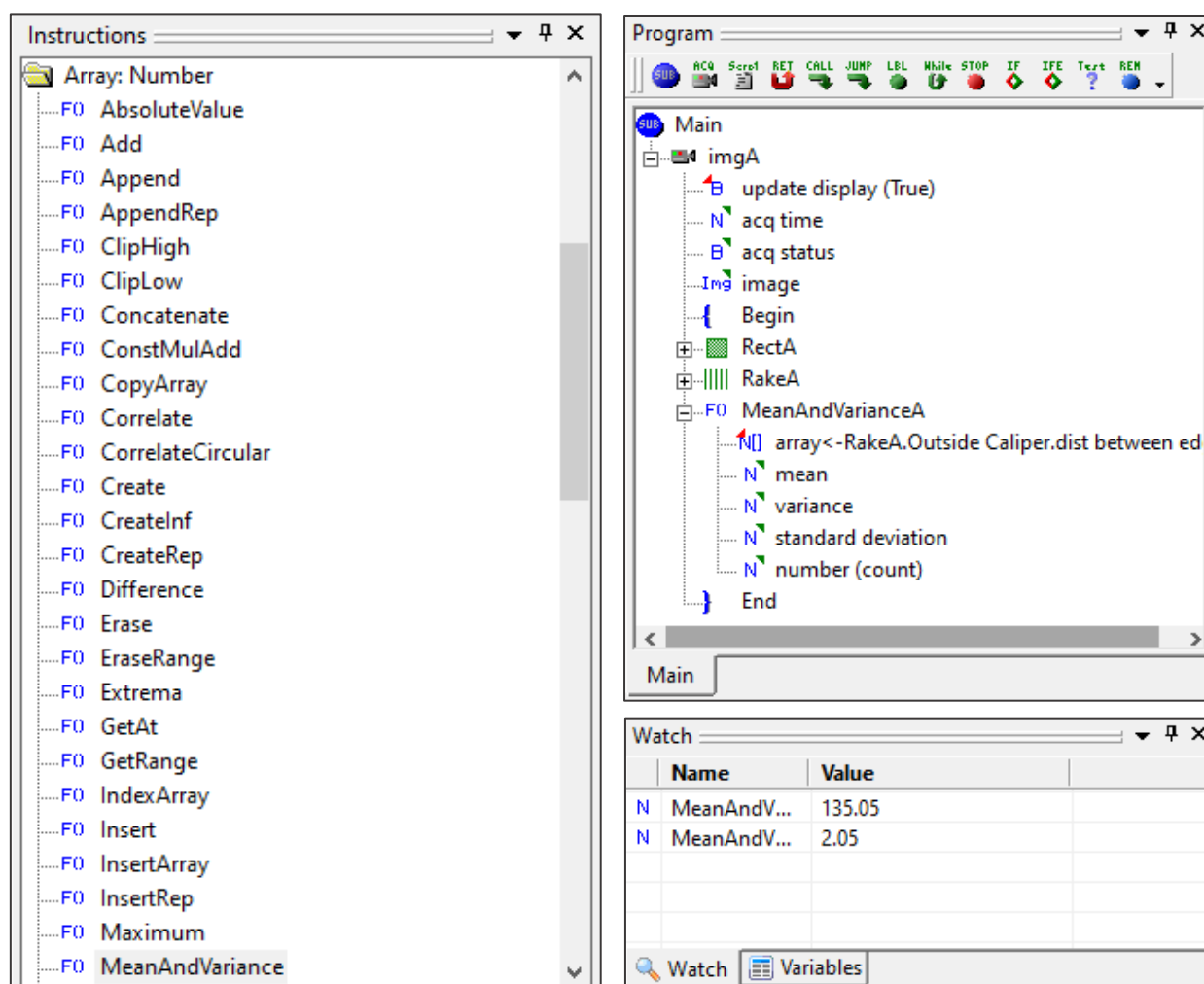
Como se está a medir a abertura da peça importa que essa medição seja feita num grande número de pontos para que o erro de medição seja menor. É ainda necessário aplicar o alinhamento para que se possam avaliar todas as imagens. Para a aplicação destas alterações devem ser seguidos os seguintes pontos:

- 4.5. Aceder ao separador *Position*.
- 4.6. Aumentar o número de elementos da região para 50.
- 4.7. Aceder ao separador *Properties*.
- 4.8. Definir o alinhamento da região de acordo com o alinhamento criado no ponto anterior.



Neste momento a investigação é capaz de fornecer as 50 distâncias das 50 linhas da região. Para converter estes valores num único só que possa ser avaliado é necessário utilizar a instrução *Array: Number* → *MeanAndVariance*. Depois de seleccionada a instrução, na janela de instruções, deve ser arrastada para a janela de programa logo abaixo da região *RakeA*.

De seguida deve ser associado o *array* de números, da distância entre arestas da região, à entrada da instrução. Para realizar esta operação o *array* deve ser arrastado para cima da entrada da mesma na janela de programa. Por fim deve-se arrastar o valor da média do desvio padrão para a janela *Watch* de forma a visualizar os resultados da ferramenta.



5. Aplicar uma calibração

Os valores da distância média e do desvio padrão do ponto anterior estão dados em pixels. Para que estes valores sejam retornados em unidades reais é necessário realizar uma calibração do sistema.

No Sherlock a calibração pode ser manual, com conversão direta entre pixels e as unidades reais, através de um padrão de xadrez, indicando as dimensões reais da grelha, ou dando a distância real de dois pontos calculados.

Para a resolução deste vai ser utilizado um padrão de xadrez para realizar a calibração do sistema.

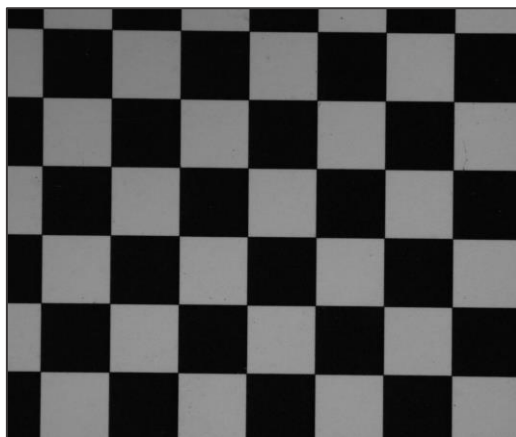
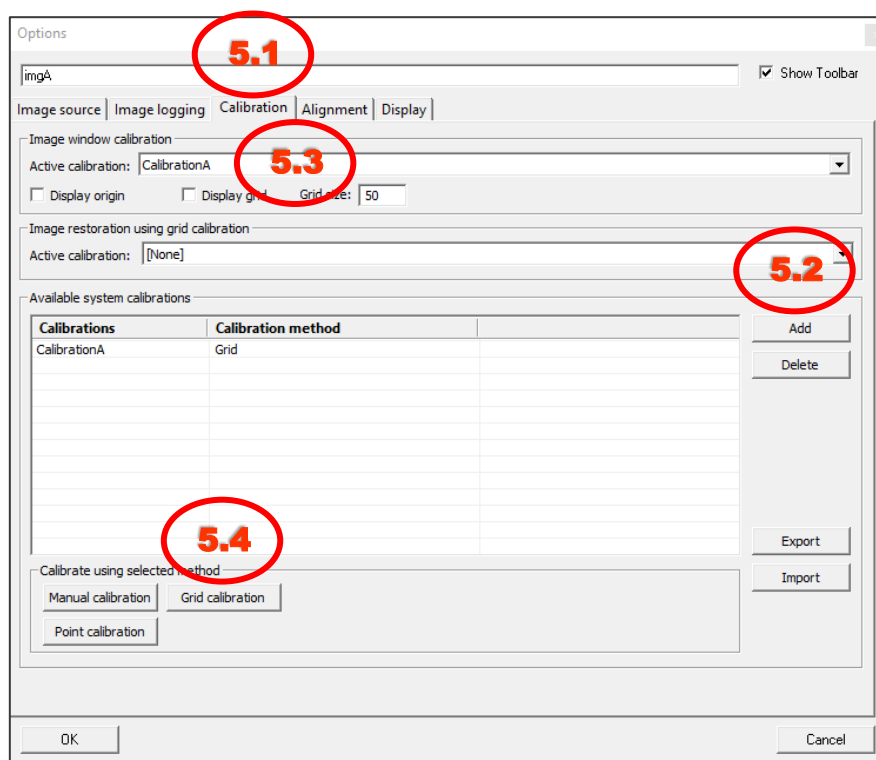


Figura 7 - Padrão de calibração.

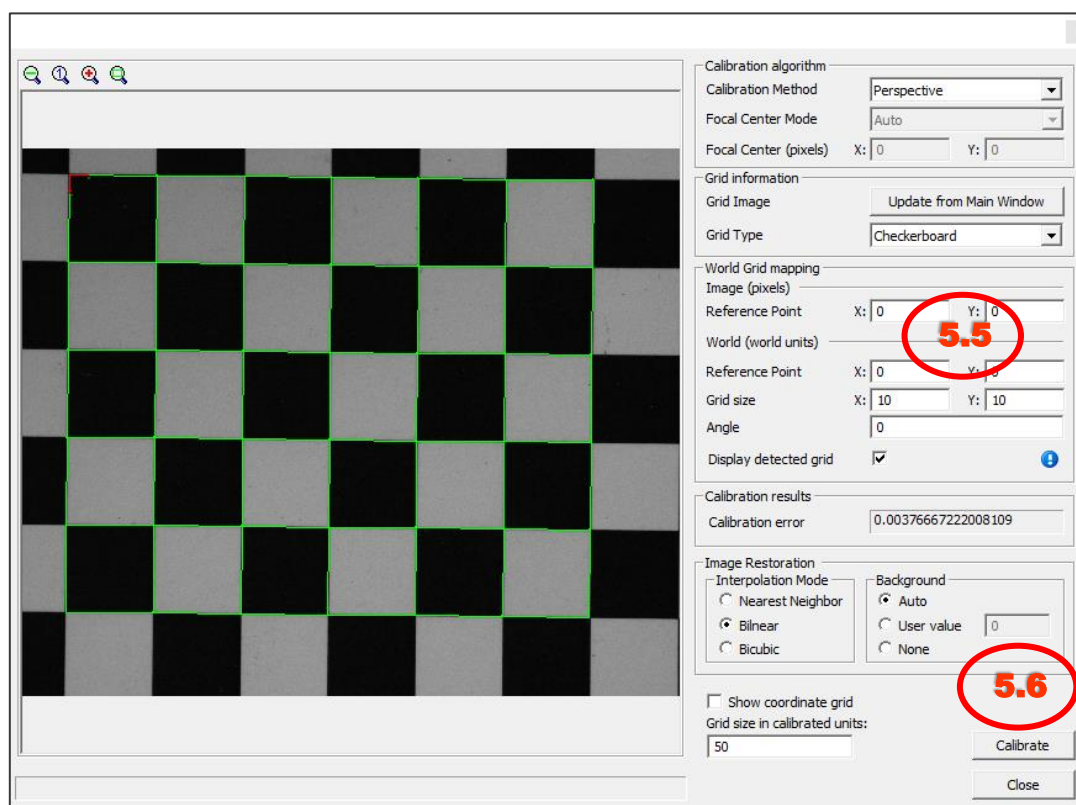
A calibração do sistema no Sherlock é bastante simples. Deve-se avançar nas imagens até encontrar a imagem do padrão, abrir as opções da janela e seguir os passos descritos para que o sistema nos dê a distâncias reais:

- 5.1. Aceder ao separador de calibração.
- 5.2. Adicionar uma nova calibração.
- 5.3. Definir a calibração criada como a ativa na janela.
- 5.4. Aceder ao assistente de calibração com grelha.

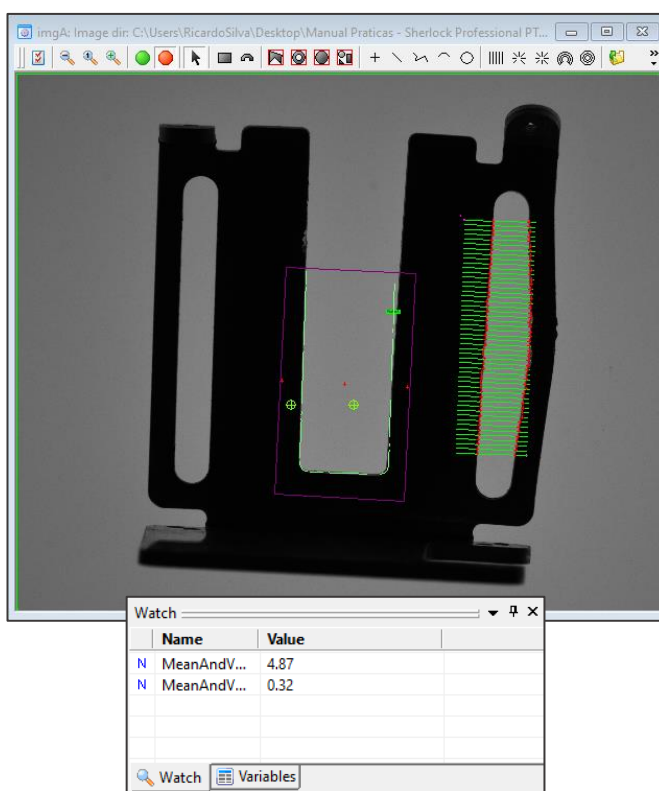
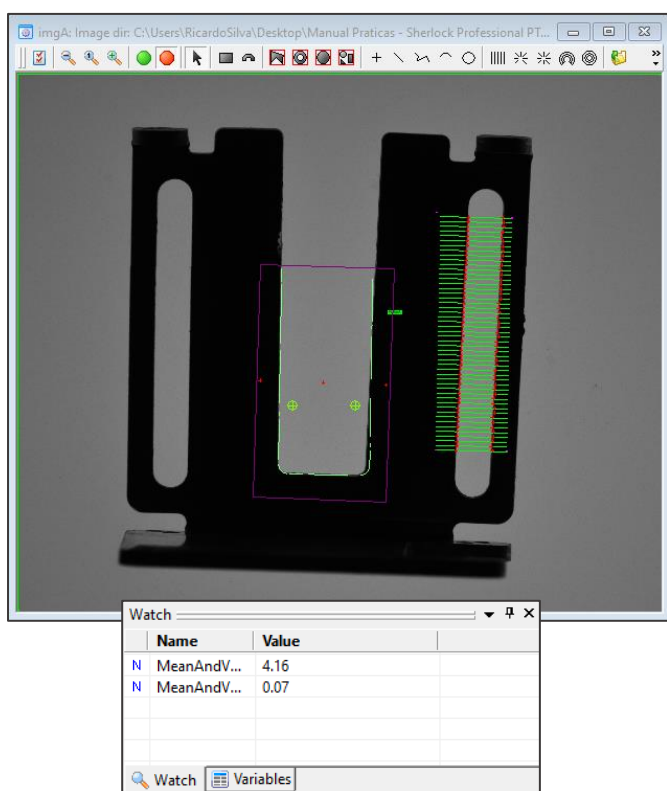


Uma vez dentro do assistente restam dois pequenos passos para terminar a calibração:

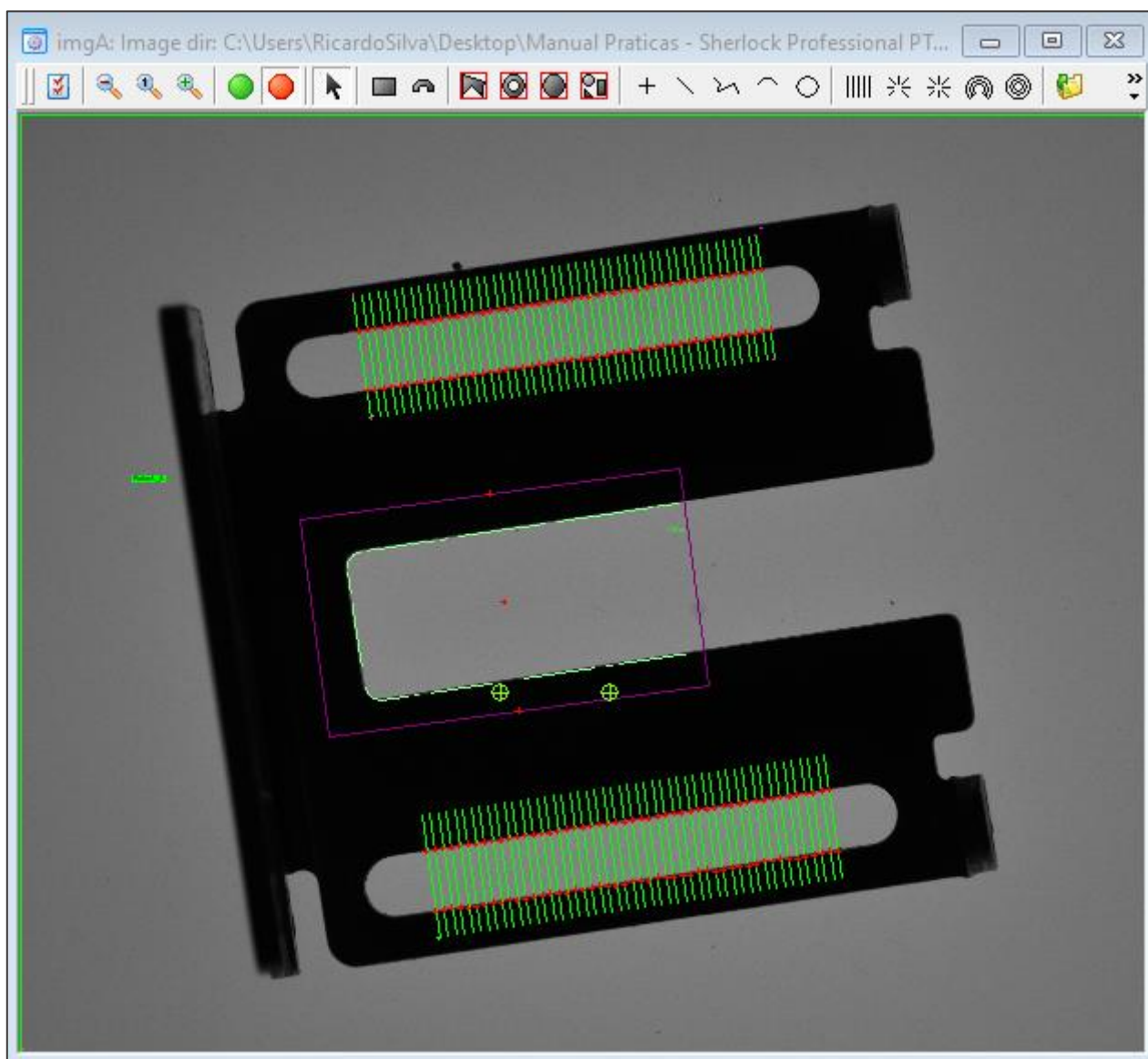
- 5.5. Indicar as dimensões reais da grelha, neste caso 10x10 mm.
- 5.6. Carregar em calibrar, e assim que as linha verdes da grelha aparecerem na imagem fechar o assistente.



Uma vez que se tenha o sistema calibrado já se consegue obter o valor real das aberturas da peça.



Tendo os valores da média e do desvio padrão reais, já é possível avaliar se as peças cumprem com os requisitos ou não. Resta agora replicar a ferramenta para a outra abertura da peça.



Exercício 4 – Cor e comunicação

O objetivo deste exercício é a verificação de tablets de comprimidos através de uma análise por cor de forma a garantir que todos os compartimentos contêm o comprimido correto.

De seguida é demonstrada, passo a passo, a construção de um programa para a análise da presença dos comprimidos com recurso à ferramenta de cor *Color Map* e envio da informação do número de comprimidos por comunicação do tipo *TCP/IP*.

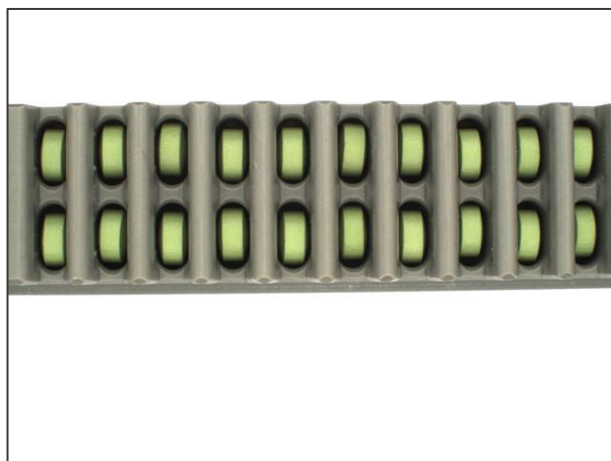


Figura 8 – Tablete com todos os comprimidos OK.

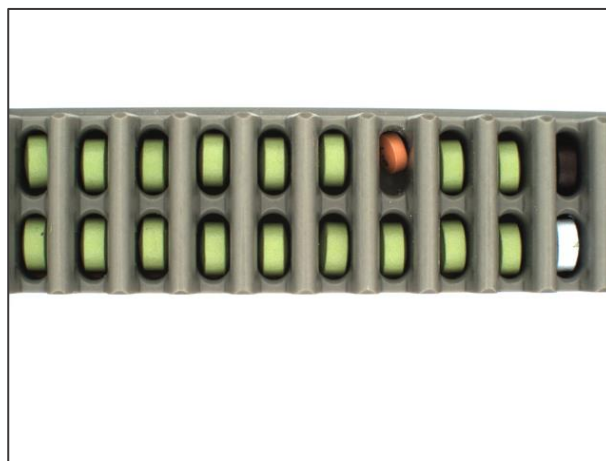


Figura 9 – Tablete com varios comprimidos NOK.

1. Iniciação da investigação

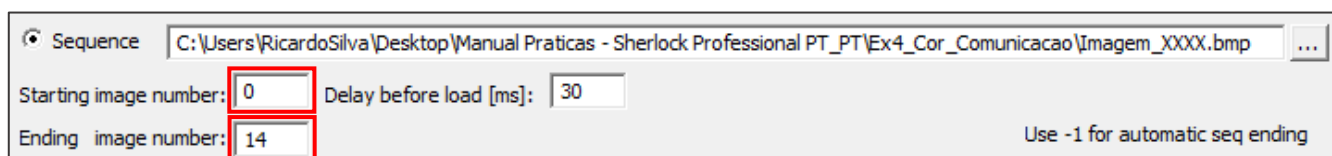
Para dar início a uma nova investigação devem ser seguidos os passos descritos no Exercício 1.

2. Carregar uma imagem

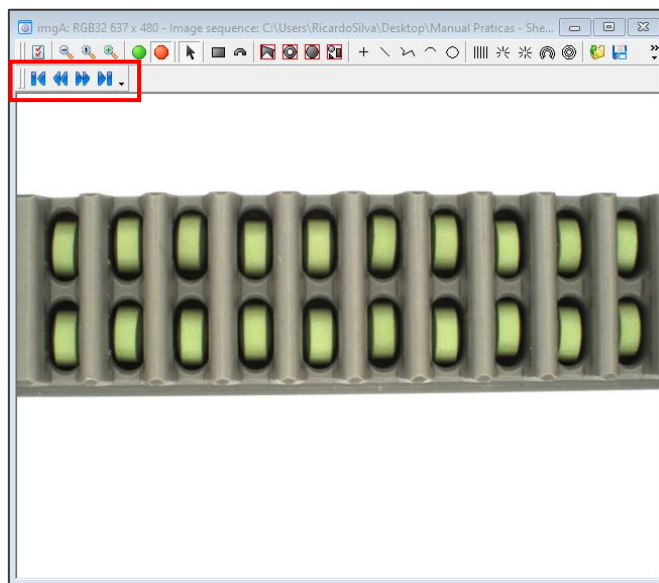
Depois de criada e guardada a solução é necessário carregar imagens para se iniciar o programa. Para isso, basta aceder às opções da janela e, no separador de fontes de imagem, selecionar opção *Sequence*. Este modo de carregamento permite carregar uma sequência numerada de imagens.

Deve de seguida ser fornecida a localização do ficheiro de imagem que se pretende carregar, neste caso irá ser carregada a imagem "*Imagem_0000.bmp*" da pasta "*Ex4_Cor_Comunicacao*" fornecida junto com este manual. Ao ser carregada esta imagem o programa vai ler todas as imagens da pasta com o nome do tipo "*Imagem_XXXX.bmp*" onde "*XXXX*" é o número da imagem.

Neste caso existem imagens desde o número 0000 até ao 0014. Esta informação também pode ser indicada para limitar a sequência de imagens carregadas.



A imagem da sequência a visualizar na janela de imagem pode ser alterada sem que seja realizada uma inspeção. Existem também comandos que permitem retornar automaticamente ao fim ou ao início da sequência de imagens carregada.

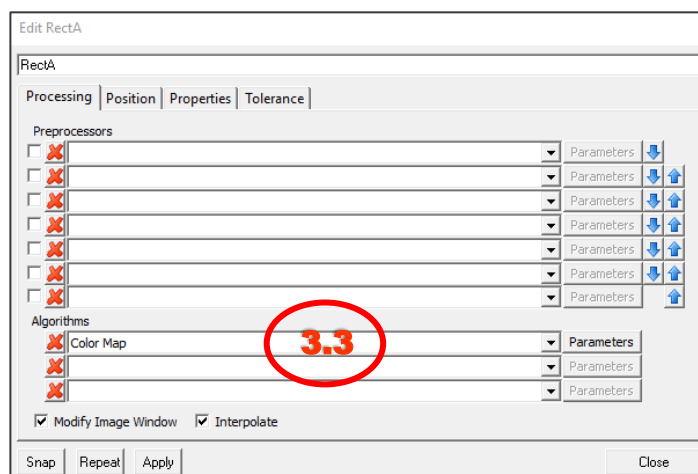


3. Aplicação da ferramenta *Color Map*

Para a detecção dos comprimidos e verificação da cor dos mesmos vai ser utilizada a ferramenta de cor *Color Map*. Esta ferramenta permite ensinar uma cor e segmentar a imagem por essa cor.

Para a criação da região de interesse e aplicação do algoritmo devem ser seguidos os seguintes passos:

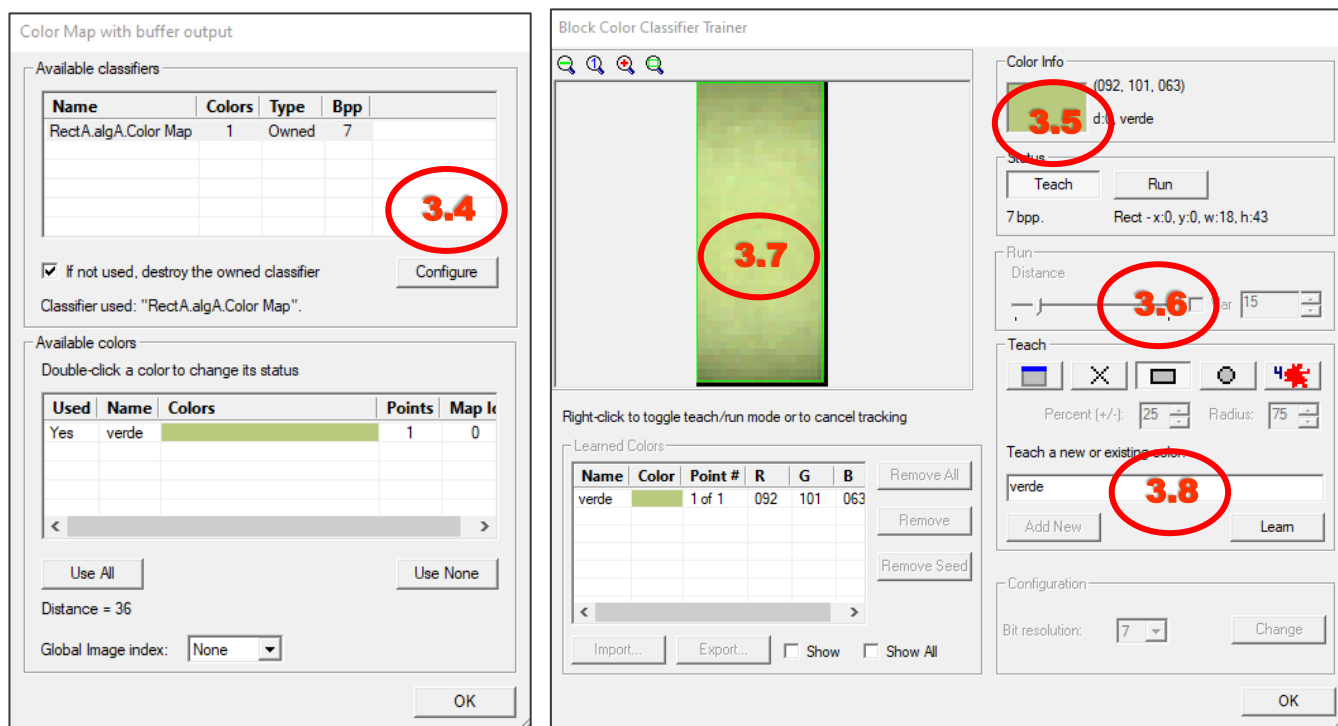
- 3.1. Selecionar a região do tipo retângulo.
- 3.2. Desenhar a região de forma a incluir a zona de cor verde num dos comprimidos da tablete.
- 3.3. Aplicar o algoritmo *Color Map* para ensinamento da cor dos comprimidos.



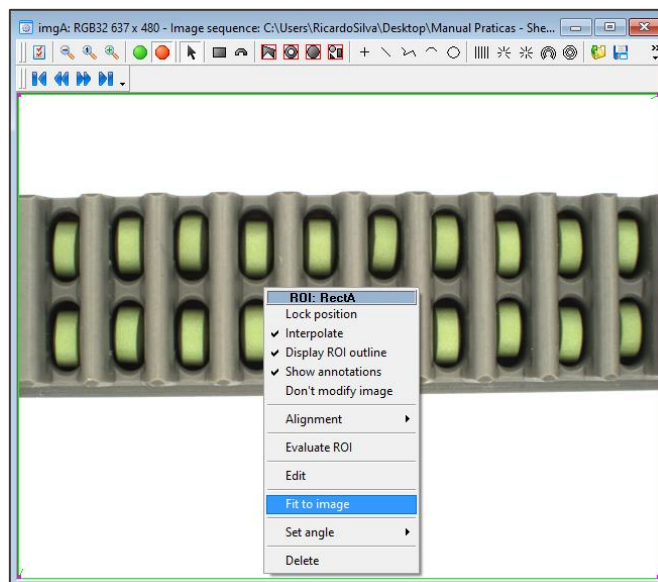
O próximo passo é configurar os parâmetros do algoritmo para ensinar o padrão de cor dos comprimidos. Para tal importa configurar os seguintes parâmetros:

- 3.4. Abrir a janela de configuração na parametrização do algoritmo.
- 3.5. Carregar em *Teach* para iniciar uma nova aprendizagem de cor.
- 3.6. Adicionar uma região de ensinamento do tipo retângulo.
- 3.7. Ajustar a região a toda a área de cor da janela.

- 3.8. Dar um nome à cor que se pretende ensinar, fazer *Add New* para adicionar a nova cor à lista e por fim carregar em *Learn* para completar a aprendizagem



Uma vez ensinada e parametrizada a ferramenta, resta ajustar a região de procura a toda a imagem.

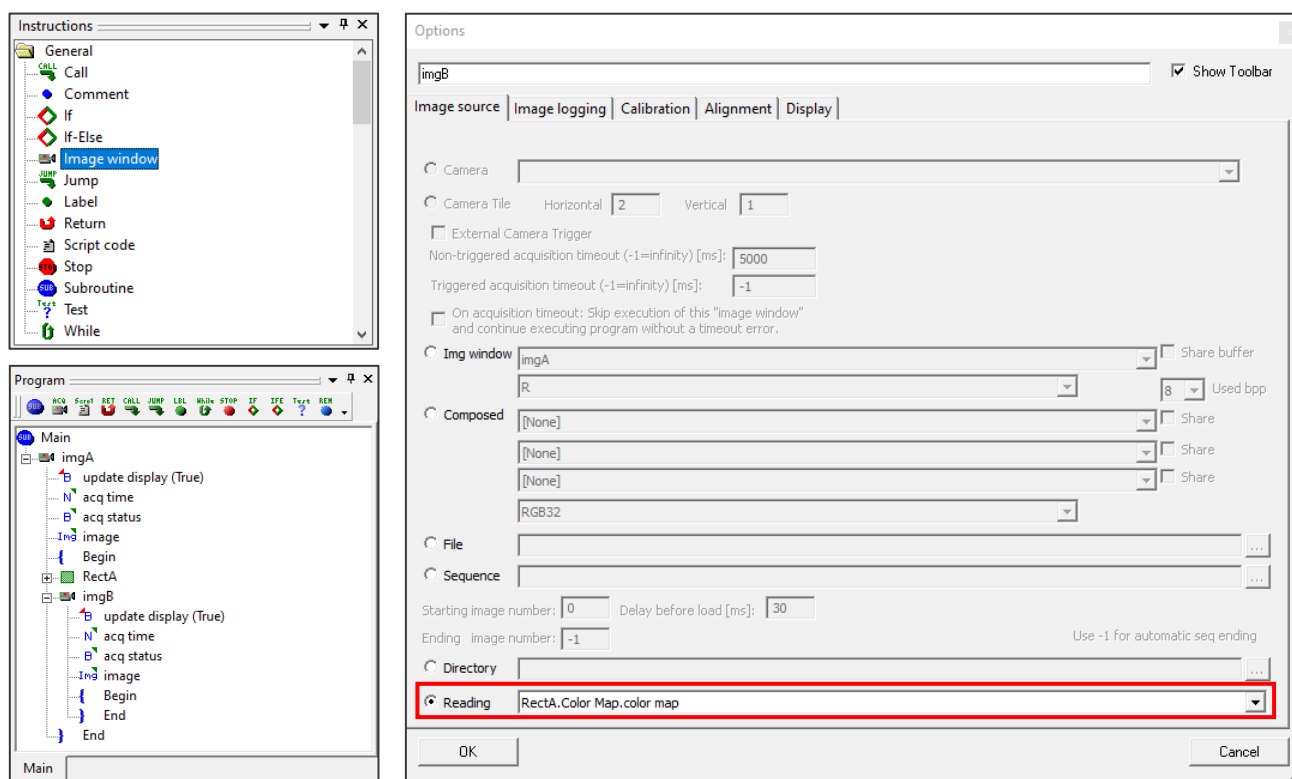


4. Segmentação da imagem

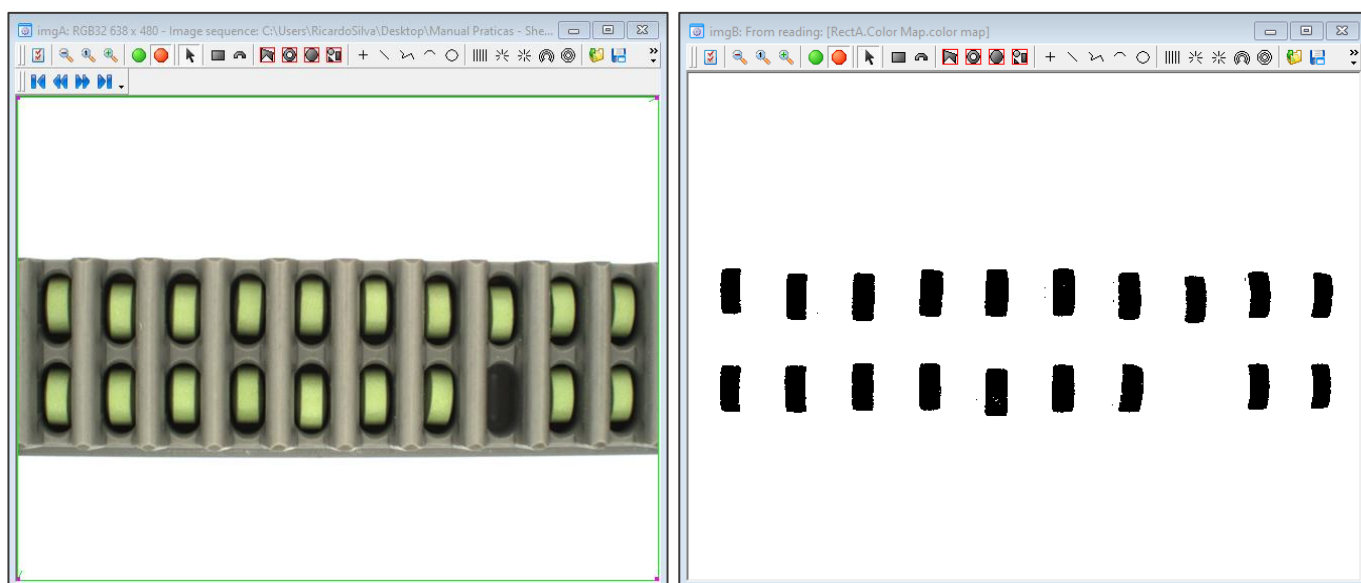
A segmentação da imagem por cor vai utilizar a cor ensinada na ferramenta do ponto anterior para isolar os comprimidos numa nova janela de imagem para a contagem dos mesmos.

A criação de uma nova janela de é um processo simples no Sherlock. Para criar uma nova janela de imagem com informação da cor ensinada segmentada devem ser executadas as seguintes tarefas:

- 4.1. Adicionar uma nova janela de imagem, arrastando a instrução *Image window* para dentro da janela *imgA* dentro da janela de programa.
- 4.2. Abrir as opções da janela criada, *imgB*, e escolher a fonte de imagem como *Reading*.
- 4.3. Selecionar como fonte de imagem a ferramenta *Color Map*.



A imagem da nova janela criada são objetos negros em fundo branco que representam os comprimidos de cor verde da imagem original. Estes objetos podem ser facilmente contados com ferramentas simples.

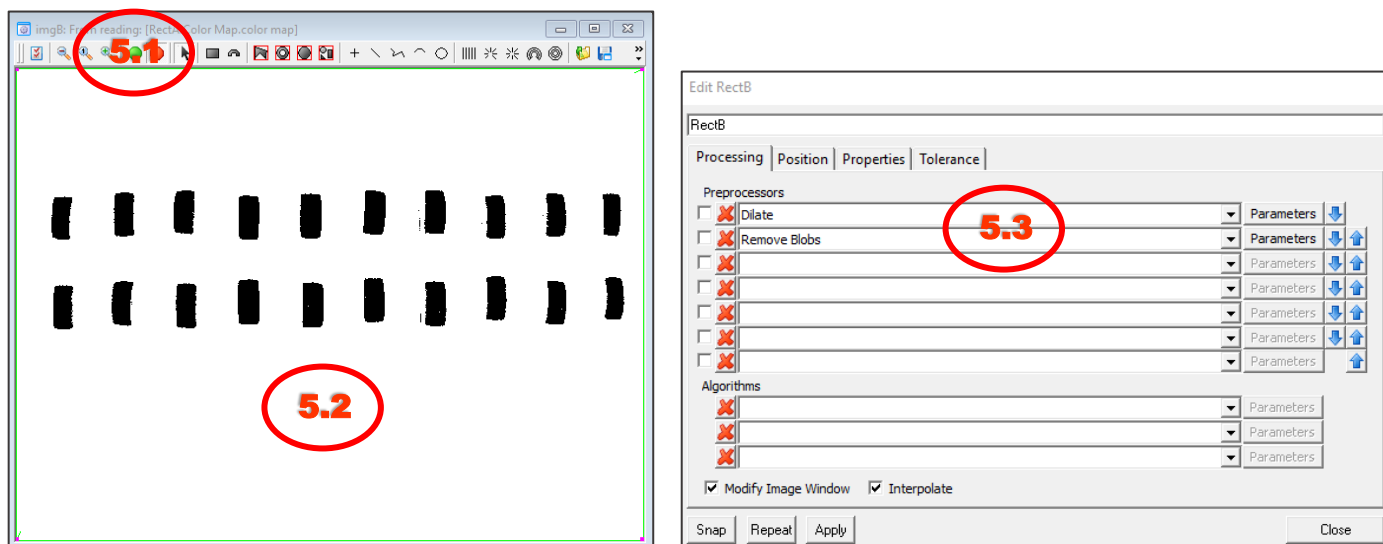


5. Contagem dos comprimidos verdes

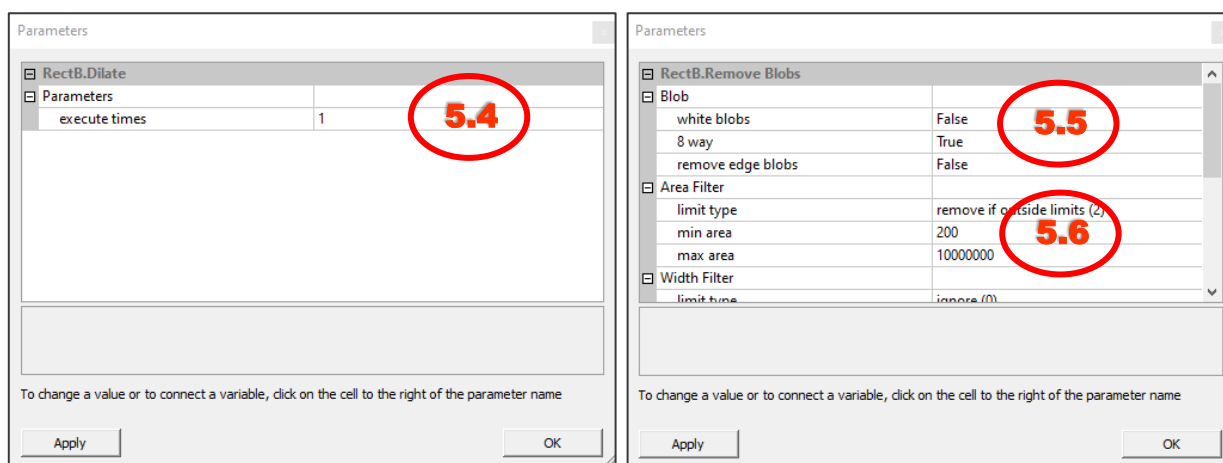
A contagem dos comprimidos vai ser realizada com o algoritmo *Connectivity – Binary*, no entanto existe algum ruído na nova imagem.

Para preparar a imagem para a aplicação do algoritmo de forma a remover o ruído da mesma devem ser aplicados alguns pré-processadores de acordo com as seguintes indicações:

- 5.1. Selecionar a região do tipo retângulo.
- 5.2. Ajustar a região a toda a imagem
- 5.3. Aplicar os pré-processadores *Dilate* e *Remove Blobs*.

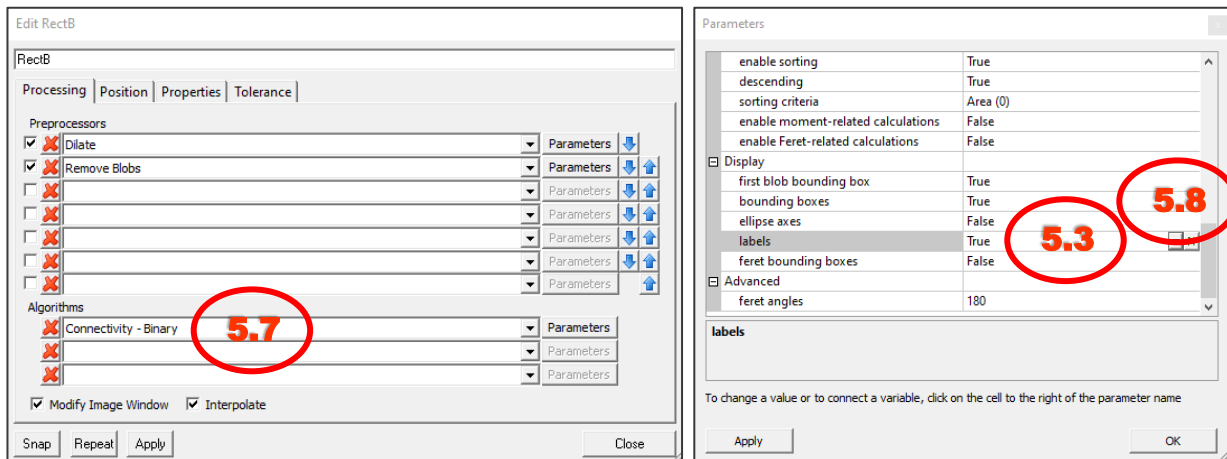


De seguida deve-se proceder à parametrização dos pré-processadores aplicados. No pré-processador de dilatação deve ser considerada apenas uma execução (5.4). Já na remoção de objetos têm de ser realizadas duas configurações: indicar que não são *blobs* brancos (5.5), e definir o valor mínimo da área dos objetos como 200 (5.6).



Uma vez que se tenha a imagem tratada pode-se aplicar o algoritmo *Connectivity – Binary*. Para a aplicação do algoritmo devem ser seguidos os seguintes pontos e configurações:

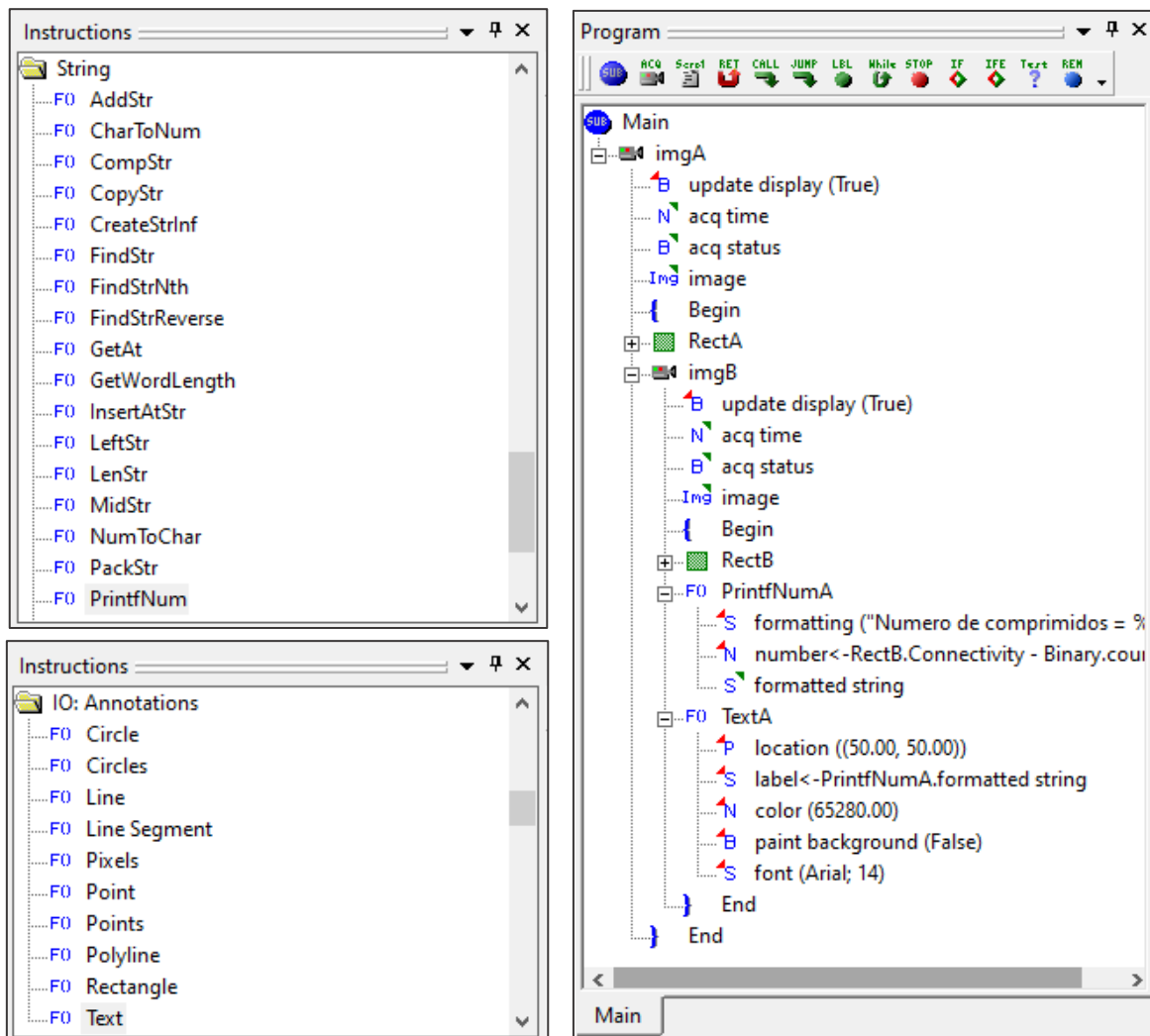
- 5.7. Aplicar o algoritmo *Connectivity – Binary*.
- 5.8. Uma vez que o número pré-definido de objetos é 20, resta ativar as *bounding boxes* para todos.
- 5.9. E por fim ativar também as numerações.



6. Apresentação dos resultados

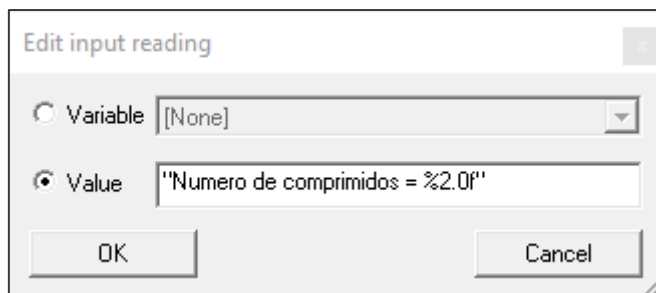
Para se conseguir uma visualização dos resultados mais intuitiva podem ser apresentadas informações de texto na janela de imagem. Neste ponto vai ser exemplificada a apresentação no número de comprimidos na janela de imagem.

O primeiro passo é a conversão do número de comprimidos, dado pelo algoritmo aplicado, em formato de texto. Para converter um número em formato de texto é utilizada a instrução *String* → *PrintfNum*. Depois é necessário apresentar esta nova variável de texto no ecrã. Isso pode ser realizado com o arrastamento da instrução *IO: Annotations* → *Text* para a janela de programação.



É ainda necessário configurar as instruções adicionadas. Para a instrução *PrintfNum* é necessário configurar duas entradas:

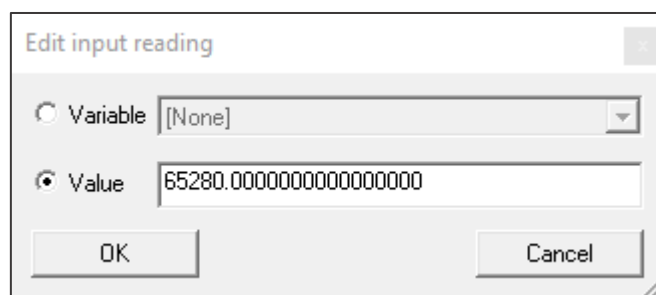
- 6.1. Configurar a formação da *string* de acordo com a imagem abaixo. Esta formação é explicada na ajuda do Sherlock procurando por *PrintfNum: string instruction*.



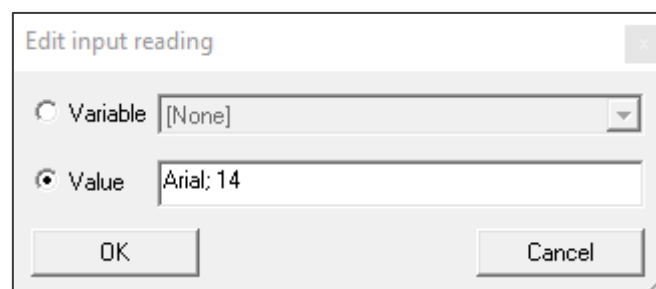
- 6.2. Arrastar a saída *Count* do algoritmo de contagem de objetos para a entrada numérica desta instrução.

Para a instrução *Text* é necessário configurar quatro entradas:

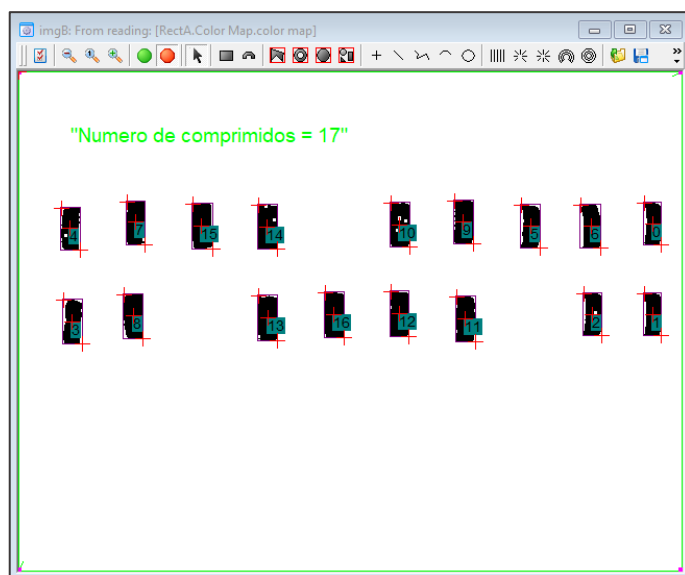
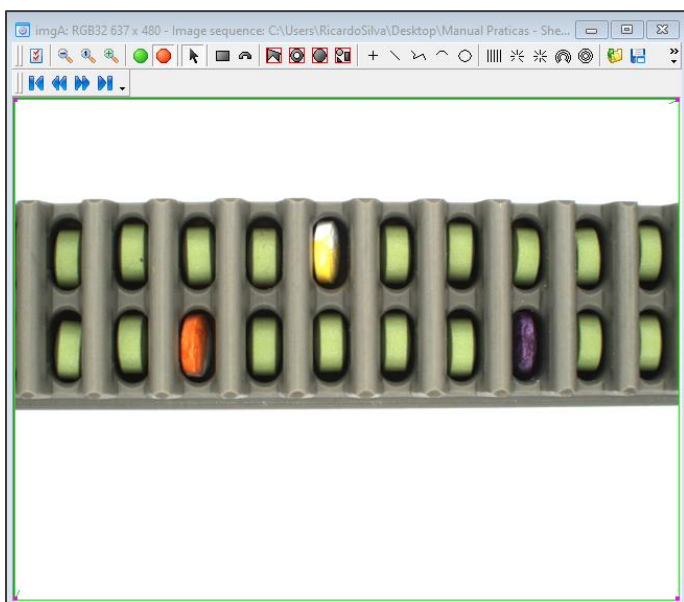
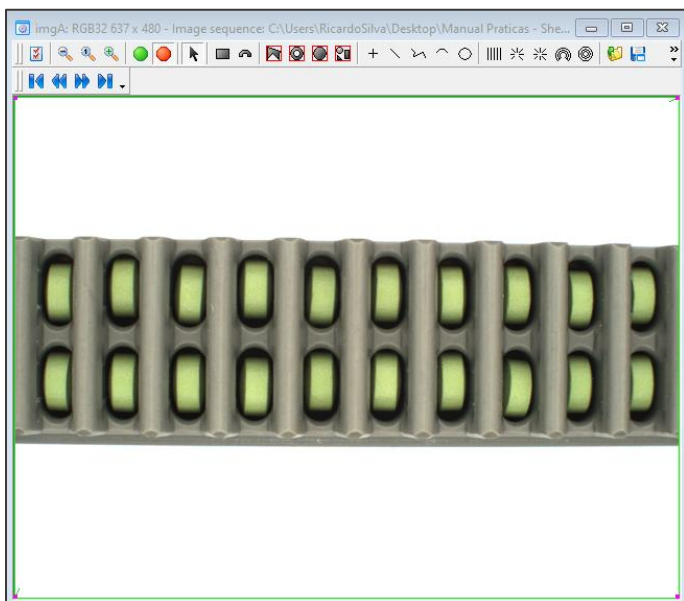
- 6.3. A localização na imagem para as coordenadas (50; 50) para que o texto não seja apresentado no canto superior esquerdo da janela de imagem.
- 6.4. Arrastar a saída *Formatted string* da instrução anterior para a entrada de texto desta instrução.
- 6.5. Configurar a cor do texto para verde fluorescente de acordo com a imagem abaixo. Esta formação é explicada na ajuda do Sherlock procurando por *Annotation colors*.



- 6.6. Configurar o tipo e tamanho da fonte do texto de acordo com a imagem abaixo. Esta formação é explicada na ajuda do Sherlock procurando por *Text: annotation IO instruction*.



Agora resta testar a investigação em todas as imagens da sequência de forma a comprovar que os comprimidos são contados corretamente.

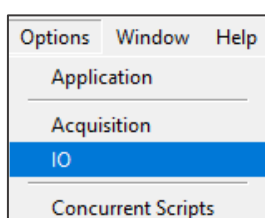


7. Comunicação TCP/IP

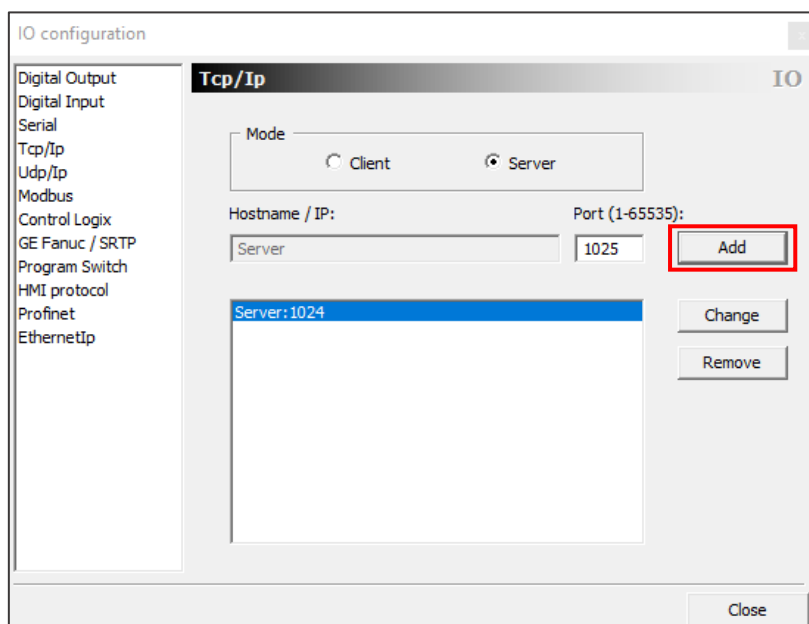
Muitas vezes é necessário enviar os valores da inspeção para outros dispositivos anexos ao sistema de visão. Para tal o Sherlock permite vários modos de comunicação com por exemplo: comunicação série, comunicação *TCP/IP*, comunicação *Modbus*, comunicação *Profinet*, etc.

Para exemplificar a criação de uma comunicação em Sherlock vai ser enviado o número de comprimidos contados em cada imagem via *TCP/IP*.

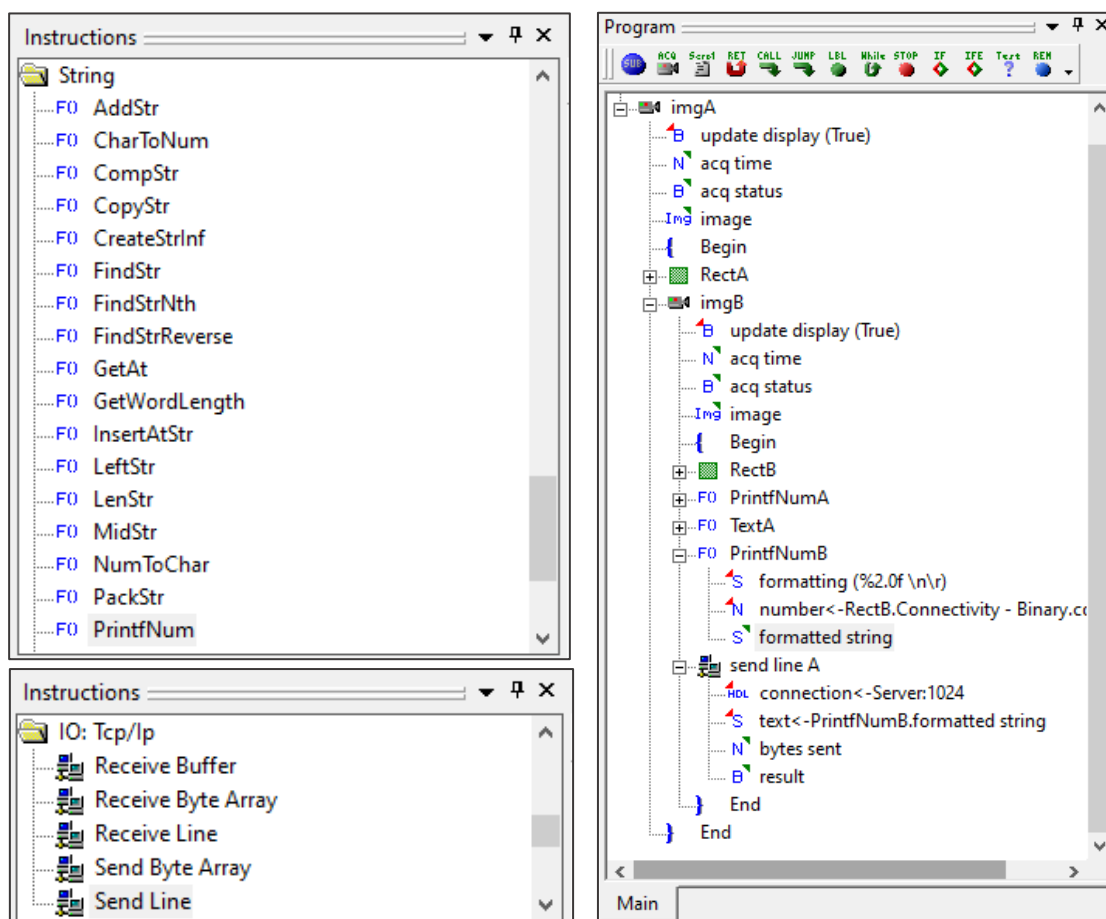
O primeiro passo é a criação de um servidor de comunicação *TCP/IP*. Para isso é necessário abrir o menu de opções e iniciar o assistente de *IO*.



Dentro do assistente deve ser selecionado o tipo de comunicação *TCP/IP* e adicionado um novo servidor numa porta disponível.

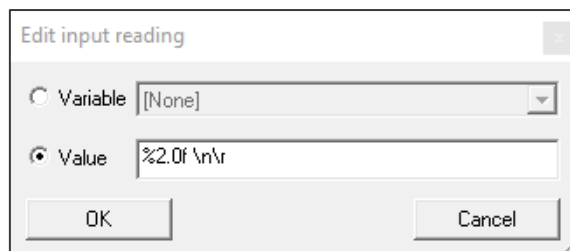


O envio da contagem de comprimidos é realizado através de instruções. Primeiro é necessário converter o número de comprimidos, dado pelo algoritmo aplicado, em formato de texto. Para converter um número em formato de texto é utilizada a instrução *String* → *PrintfNum*. Depois é necessário enviar a mensagem através da comunicação criada. Isso pode ser realizado com o arrastamento da instrução *IO: TCP/IP* → *Send Line* para a janela de programação.



É agora necessário configurar as instruções adicionadas. Para a instrução *PrintfNum* é necessário configurar duas entradas:

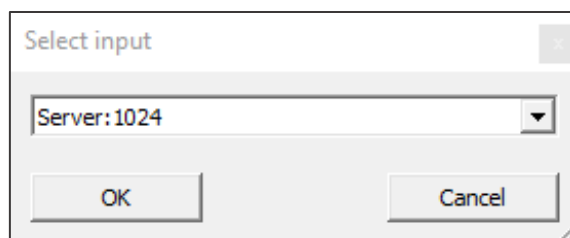
- 7.1. Configurar a formação da *string* de acordo com a imagem abaixo, onde “/n” e “/r” são, respetivamente o *line feed* e o *carriage return*.



- 7.2. Arrastar a saída *Count* do algoritmo de contagem de objetos para a entrada numérica desta instrução.

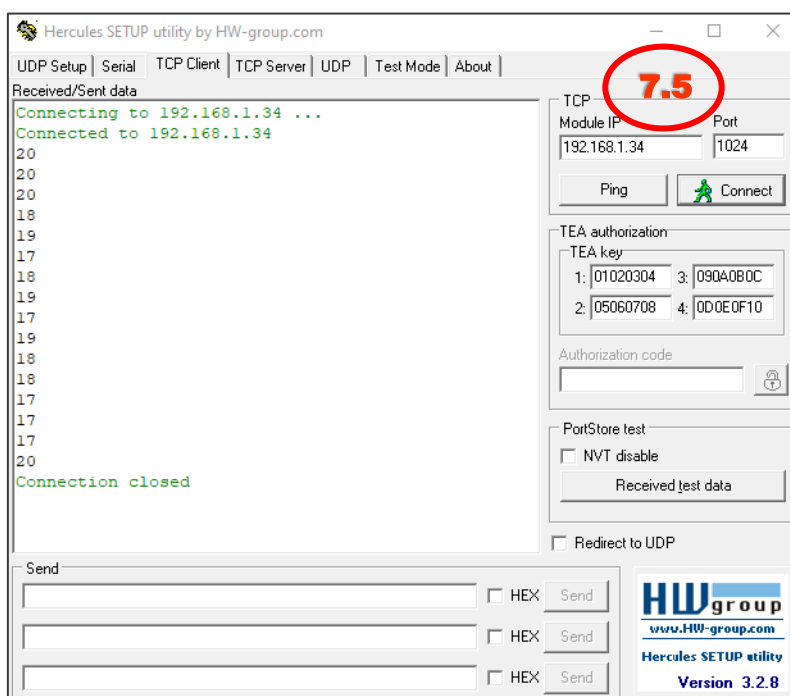
Para a instrução *Send Line* é necessário configurar duas entradas também:

- 7.3. Arrastar a saída *Formatted string* da instrução anterior para a entrada de texto desta instrução.
- 7.4. Selecionar a porta do servidor criado no assistente de comunicação.



Como se configurou o Sherlock como *socket server*, é necessário um *client* para receber as mensagens da comunicação. Para tal pode-se utilizar aplicação HERCULES.

A aplicação deve ser configurada como *TCP Client* e deve ser introduzido o endereço IP do computador e a porta de saída previamente configurada (2005), (7.5).



Exercício 5 – Ligação de uma câmara

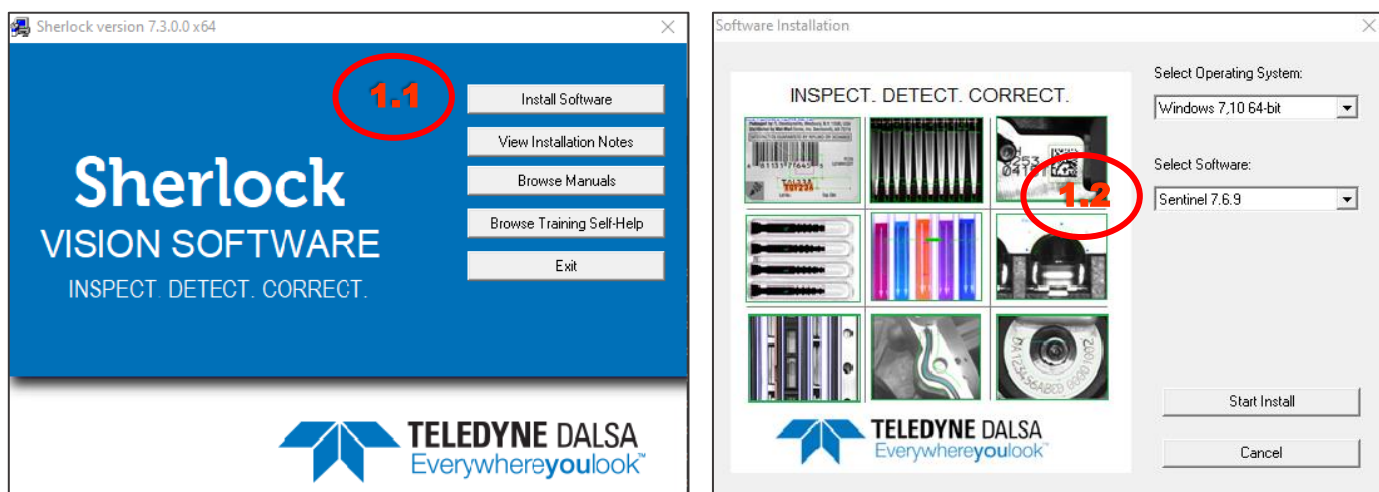
O objetivo deste exercício é demonstrar a ligação de uma câmara ao Sherlock para uma aquisição de imagem em tempo real.

Como referido no início deste manual, a aquisição de imagem a partir de uma câmara só é possível na versão profissional do Sherlock. Desta forma, a realização deste exercício está condicionada à existência de uma licença. Ainda assim, é importante que o processo de aquisição direta de uma câmara seja exemplificado para aplicações reais futuras.

A câmara a ligar é uma *Teledyne Dalsa Genie Nano M4020* e como tal deve ser utilizado o *Sapera LT* como *driver* de controlo da câmara e o *Sapera CamExpert* para configuração da mesma.

1. Instalação do Sentinel

Antes de dar início a qualquer configuração é necessário instalar o *Sentinel*. Esta aplicação é responsável pela gestão de licenças *dongle* do Sherlock. Para instalar esta aplicação é necessário executar o assistente de instalação do Sherlock, como ilustrado no Anexo A. Depois de abrir o assistente de instalação deve ser selecionada a instalação de *software* (1.1). De seguida resta selecionar o *Sentinel* e instalar (1.2).

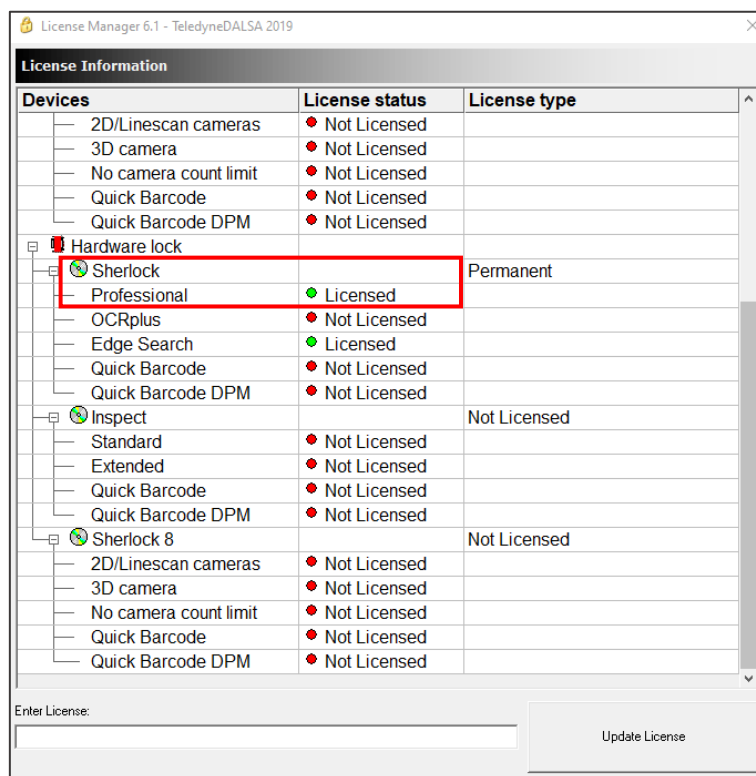


2. Licenças

As licenças do Sherlock podem ser de dois tipos: podem ser associadas ao *MAC Address* do PC onde o Sherlock está instalado ou podem ser licenças físicas sob a forma de *pen USB* ligada ao PC (*dongles*).

Para introduzir os códigos de licenças associadas ao *MAC Address* e verificar e gerenciar a existência de licenças válidas é instalado, juntamente com o Sherlock, o *License Manager*.

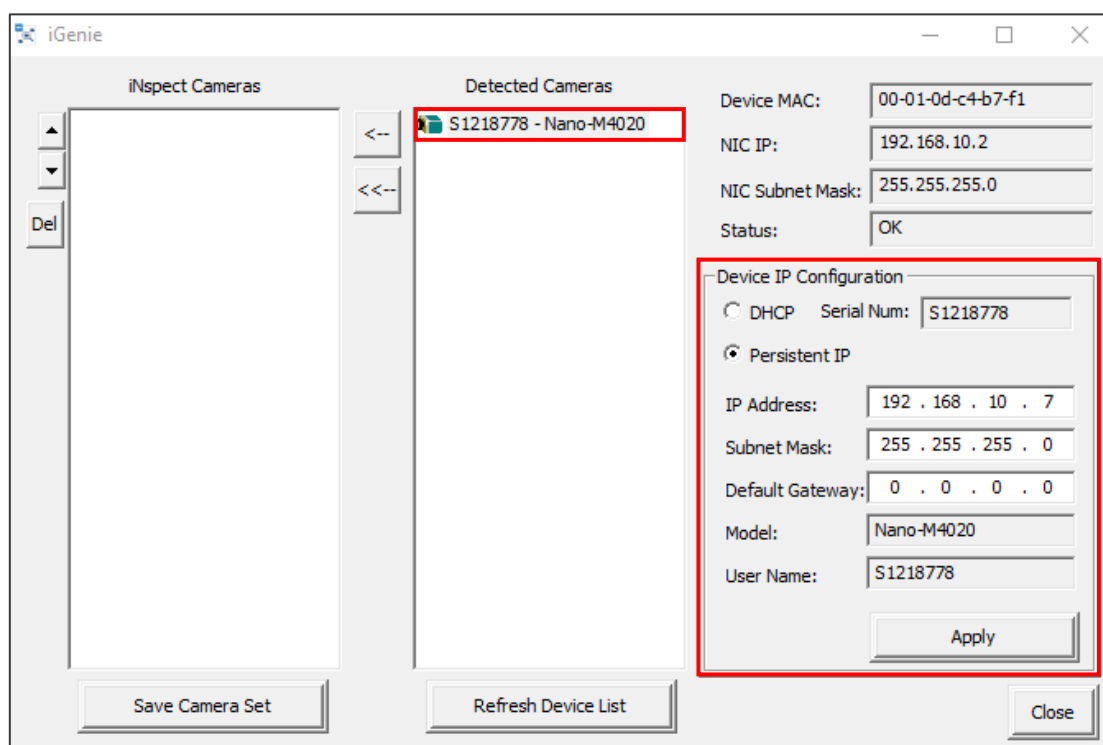
Para a resolução deste exercício é necessário que o Sherlock profissional esteja licenciado.



3. Configuração dos parâmetros da câmara no *Sapera CamExpert*

A configuração dos parâmetros da câmara tais como tempo de exposição, ganho, etc. deve ser realizada no *Sapera CamExpert*.

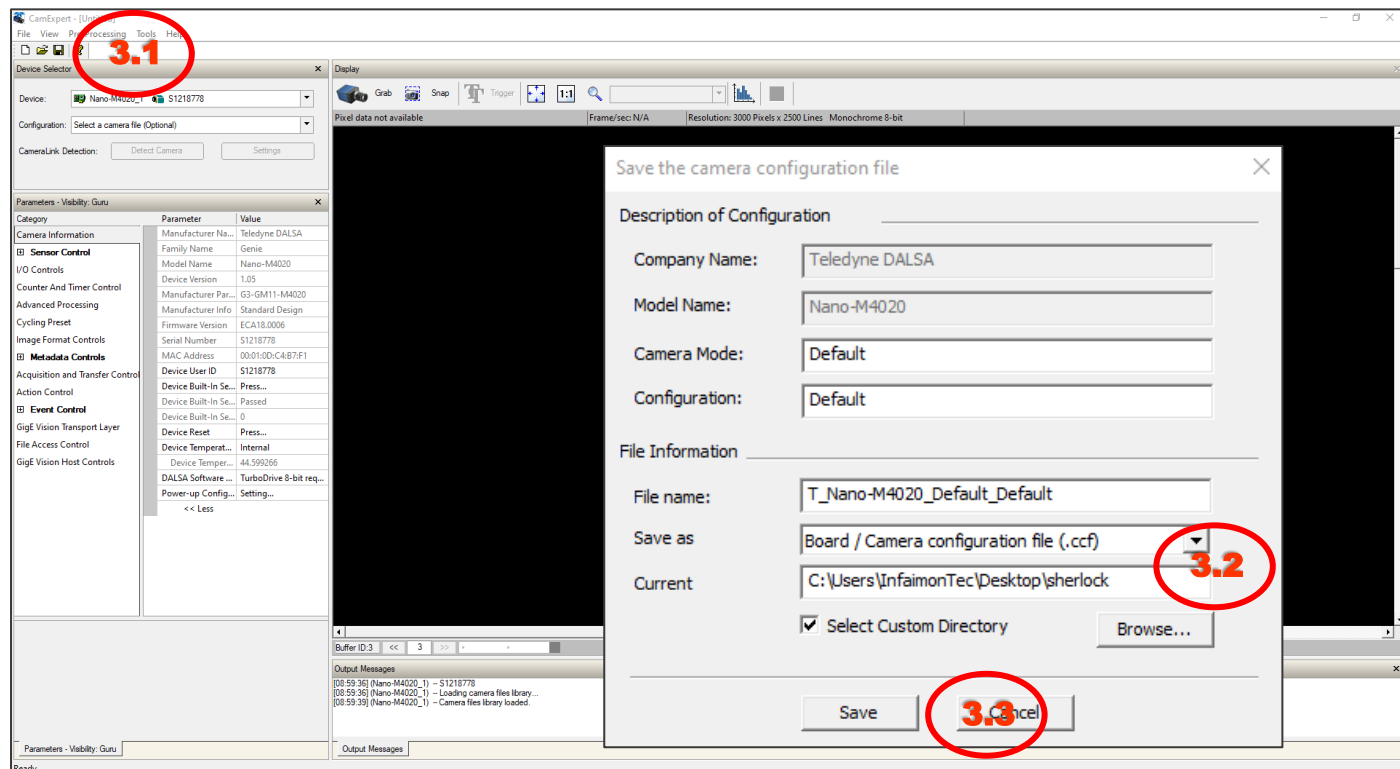
Antes de iniciar o *Sapera CamExpert* é necessário configurar os parâmetros de rede da câmara. Para isso deve ser iniciado o *iDiscoverGenie* e o IP da câmara deve estar na mesma gama da porta do PC à qual está ligada. Uma vez configurada a câmara esta deve aparecer a verde na janela de câmaras detetadas.



De seguida, pode iniciar-se a configuração dos parâmetros da câmara no *Sapera CamExpert* até que a imagem obtida esteja de acordo com os objetivos da aplicação. Os parâmetros definidos têm de ser então guardados para posterior carregamento no Sherlock.

Para guardar o ficheiro de configurações da câmara basta realizar os seguintes passos:

- 3.1. Carregar e guardar.
- 3.2. Selecionar o nome e o diretório onde se pretende guardar o ficheiro de configuração .ccf.
- 3.3. Guardar o ficheiro.

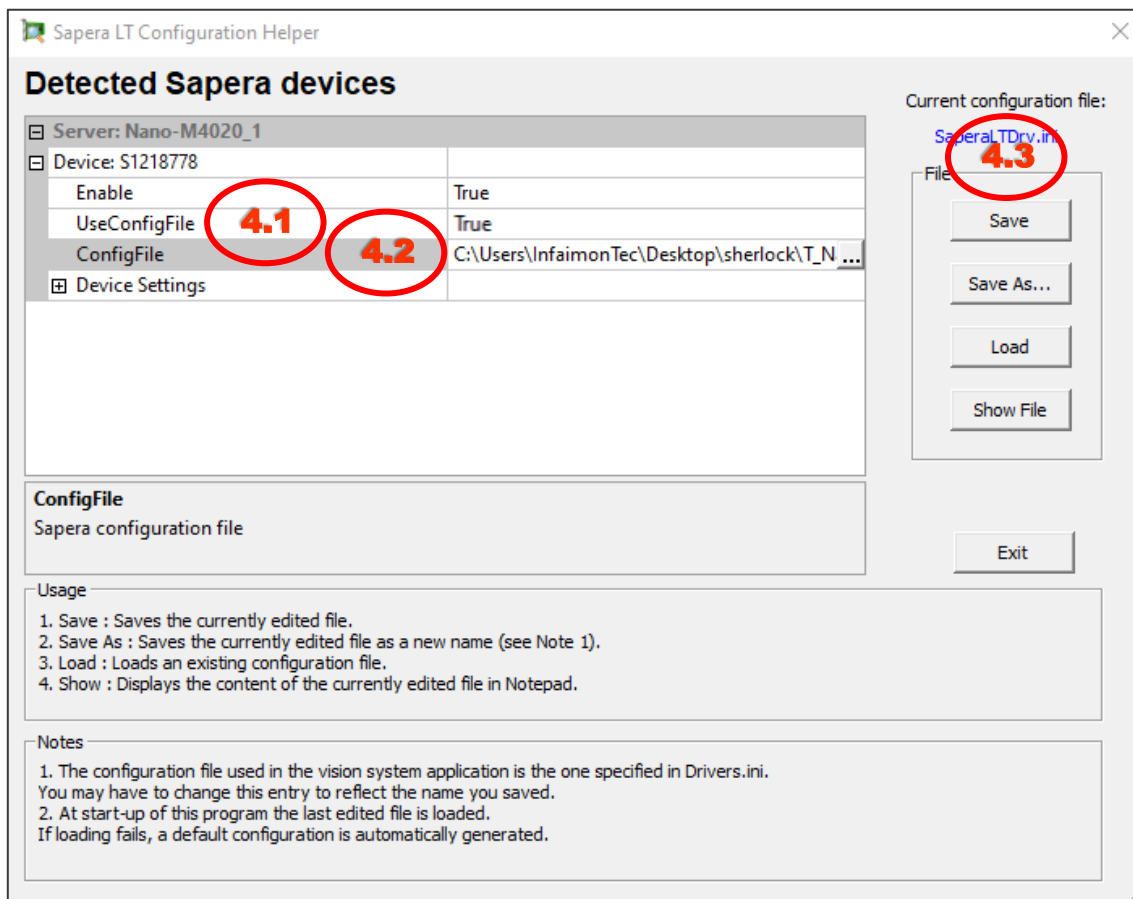


4. Adicionar a câmara ao *Sapera Acquisition Wizard*

Ao iniciar o *Sapera Acquisition Wizard* é mostrada a lista de dispositivos detetados. Aqui também é possível configurar os parâmetros da câmara em *Device Settings*, no entanto não existe janela de visualização para verificar as alterações. Uma vez que se havia guardado o ficheiro de configuração da câmara vão-se utilizar esses parâmetros.

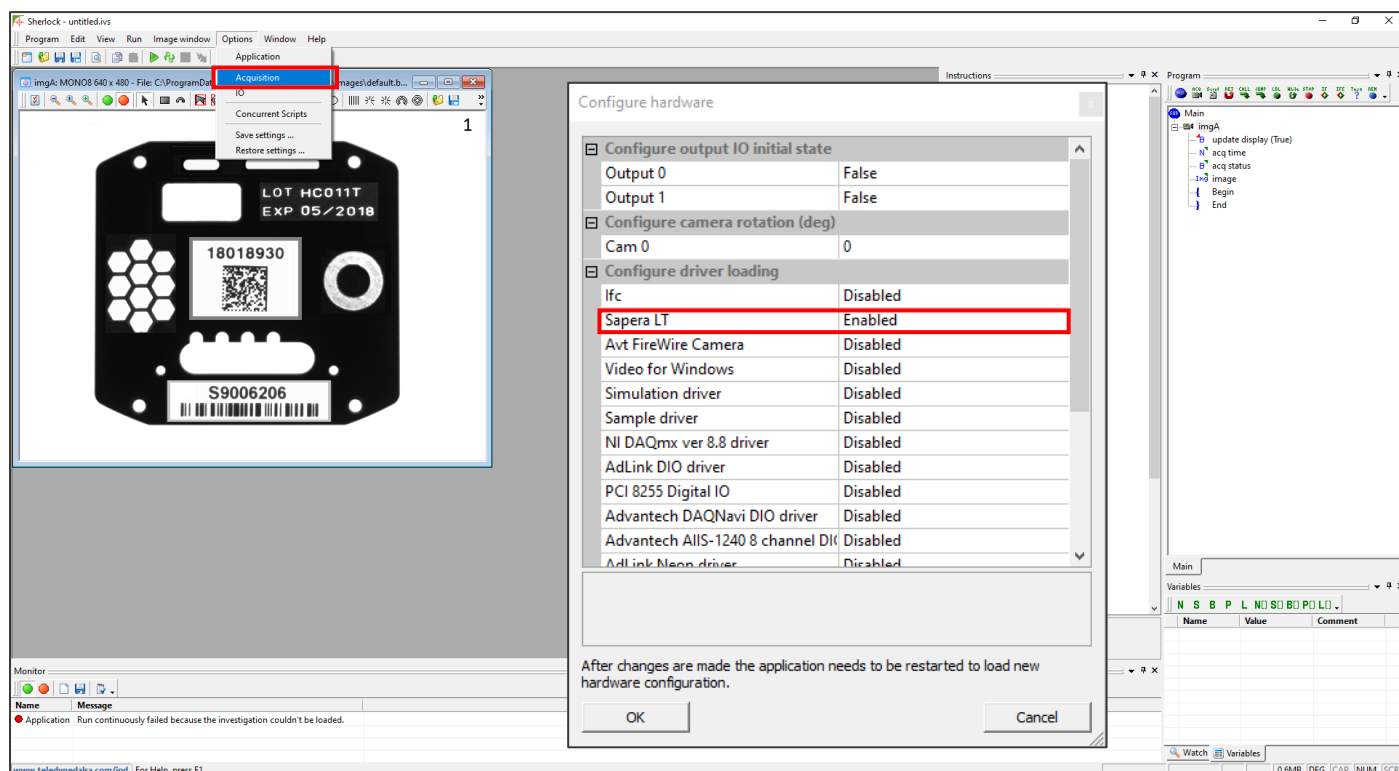
Para carregar o ficheiro de configuração seguem-se os seguintes pontos:

- 4.1. Ativar a utilização de ficheiro de configuração;
- 4.2. Carregar o ficheiro de configuração .ccf do diretório.
- 4.3. Guardar as alterações realizadas.

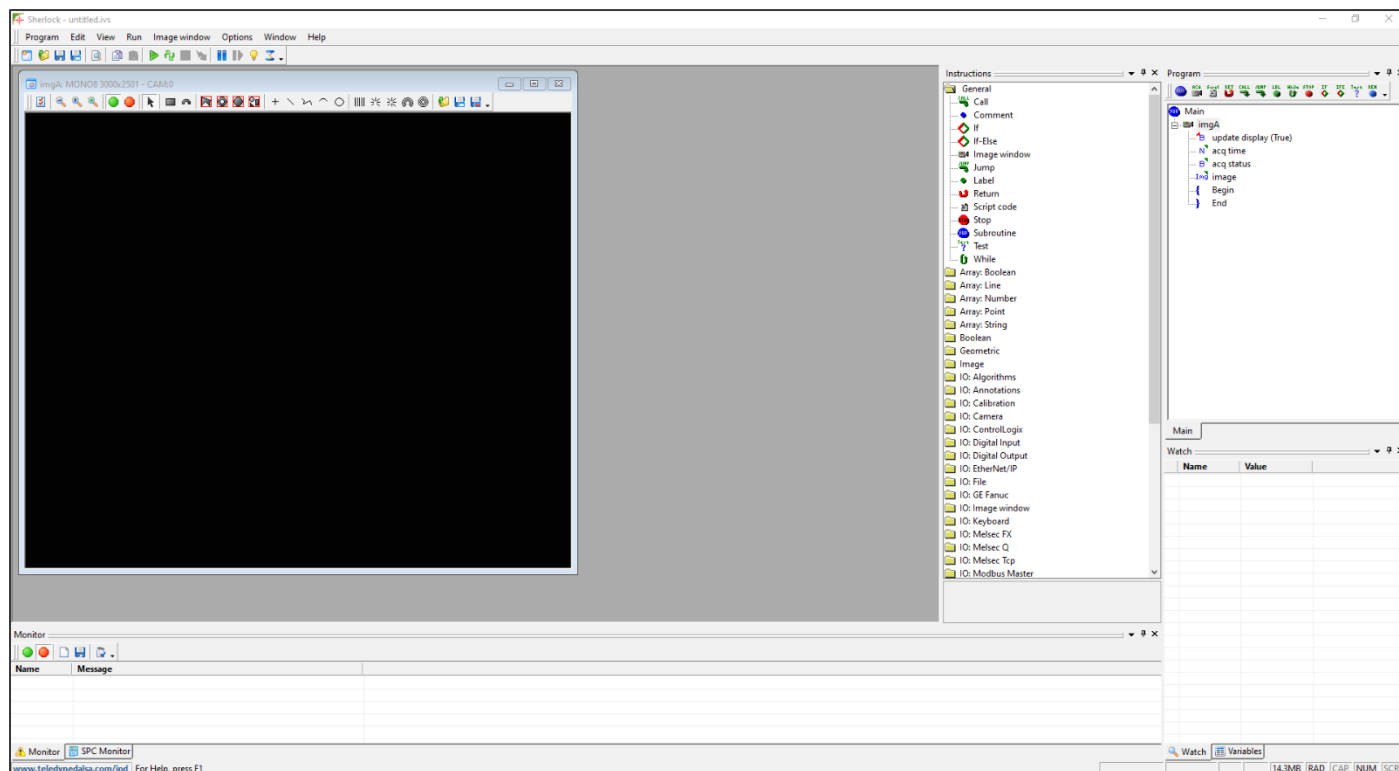


5. Adicionar a câmara ao Sherlock

Para adicionar a câmara ao Sherlock deve-se aceder às opções de aquisição e ativar o *Sapera LT* como *driver* a carregar pela investigação.



Uma vez guardada esta configuração, o Sherlock está pronto para a aquisição a partir da câmara. Caso a definição de abertura automática da câmara, quando se cria uma nova janela, esteja ativada basta reiniciar o Sherlock para se ter a aquisição em tempo real. Caso contrário, basta ativar a câmara nas opções da janela de aquisição.



ANEXO A

Instalação do Sherlock

Para efetuar a instalação do software, apenas necessita de executar em modo de administrador o ficheiro de instalação e seguir os passos descritos abaixo.

Nome	Data de modificação	Tipo	Tamanho
ADLink Drivers	13/01/2020 08:29	Pasta de ficheiros	
CVBRuntime	13/01/2020 08:30	Pasta de ficheiros	
Documentation	13/01/2020 08:30	Pasta de ficheiros	
ExampleImages	13/01/2020 08:30	Pasta de ficheiros	
GeniCam	13/01/2020 08:30	Pasta de ficheiros	
GuiBuilder	13/01/2020 08:30	Pasta de ficheiros	
SaperaDrivers	13/01/2020 08:31	Pasta de ficheiros	
SaperaLTRuntime	13/01/2020 08:31	Pasta de ficheiros	
Sentinel	13/01/2020 08:31	Pasta de ficheiros	
Setup	13/01/2020 08:31	Pasta de ficheiros	
Sherlock	13/01/2020 08:31	Pasta de ficheiros	
Training Self-Help	13/01/2020 08:31	Pasta de ficheiros	
Utilities	13/01/2020 08:31	Pasta de ficheiros	
Readme7300	12/12/2019 21:11	Documento de te...	5 KB
setup	18/07/2012 17:01	Aplicação	208 KB
setup	22/12/1997 02:28	Ícone	1 KB
setup	13/01/1998 01:52	Definições de con...	1 KB

