



Universidade Técnica de Lisboa

Instituto Superior Técnico

*Aplicações do Processamento de Sinal a
Sistemas Multimédia*

Trabalho de Laboratório

**Calibração Automática de Câmaras
e
Identificação de Robots em Imagens**

Realizado por:

Jorge Paiva *n.º 49393*
Luís Tavares *n.º 50771*

Lisboa, 28 de Maio de 2006

Índice

1. Introdução.....	2
2. Problema	3
3. Abordagem.....	4
3.1 – Calibração automática da câmara	4
3.2 – Identificação do robot numa imagem.....	7
4. Implementação.....	10
4.1 – Calibração automática da câmara	10
4.1.1 – Pré-processamento	10
4.1.2 – Cálculo da <i>Transformada de Hough</i>	14
4.1.3 – Agrupamento de rectas em classes e identificação das pertencentes à grelha	16
4.1.4 – Determinação da matriz P	19
4.2 – Identificação do robot numa imagem.....	21
4.2.1 – Pré-processamento	21
4.2.2 – Identificação das marcas.....	24
4.2.3 – Determinação da posição e orientação do robot.....	25
5. Resultados.....	26
5.1 – Calibração automática da câmara	26
5.2 – Identificação do robot numa imagem.....	29
6. Conclusões.....	31
Apêndice A – <i>Transformada de Hough</i>	35
Apêndice B – Método dos Mínimos Quadrados.....	37
Apêndice C – Modelo da Câmara	39

1. Introdução

O presente trabalho foi desenvolvido como complemento para um Trabalho Final de Curso¹ e tem como objectivo auxiliar a localização de uma plataforma móvel que navega em cenários conhecidos *a priori*. Embora o robot seja capaz de estimar a sua posição e orientação por odometria, este processo integra erros que necessitam ser corrigidos. A solução implementada utiliza câmaras instaladas no tecto, que incidem ortogonalmente sobre o solo. Estas vão ser usadas para identificar o robot e actualizar a sua posição e orientação no cenário. A primeira parte deste trabalho descreve uma ferramenta de calibração automática de câmaras. Este procedimento é necessário para que se consiga relacionar a realidade observada pelas câmaras com a imagem captada. Na segunda parte procede-se à identificação automática do robot na imagem. Tendo esta informação e usando a calibração, procede-se à correcção da posição da plataforma móvel no cenário.

Neste relatório é descrito detalhadamente o funcionamento das duas ferramentas concebidas, sendo apresentados exemplos e estudada a robustez das mesmas.

¹ Robot de Serviço Híbrido para Cenário Interiores – 2005/2006 - Proposta N.º 52

2. Problema

A calibração de câmaras é um processo que pode ser descrito sucintamente como uma transformação que faz corresponder pontos numa imagem a pontos do mundo. Este processo é especialmente útil quando se pretende estimar a localização de um objecto usando câmaras, como é o caso. A realização automática da calibração é um passo essencial para o trabalho em que este projecto se insere, uma vez que permite uma instalação rápida e eficiente das câmaras, sem que seja necessário que o utilizador esteja familiarizado com a teoria de processamento de imagem envolvida. As questões principais a resolver na implementação desta ferramenta são como identificar os pontos de interesse na imagem, como relacioná-los com os pontos do mundo e qual a robustez do método face à luminosidade e qualidade das câmaras.

Relativamente à identificação do robot em imagens, trata-se de um problema potencialmente muito complexo, pelo que se optou pela detecção de marcas com cores conhecidas. Estas serão colocadas sobre a plataforma móvel, restringindo-se a procura da posição do robot à detecção das marcas. Esta simplificação não conduz no entanto a um problema trivial, uma vez que não é imposta nenhuma restrição cromática ao fundo na imagem, pelo que podem existir zonas com mesma cor das marcas. Também neste processo é necessário minimizar o efeito da luminosidade e da qualidade das câmaras. É de notar que ao contrário da calibração, a identificação do robot na imagem deve ser realizada num tempo de processamento muito reduzido, uma vez a sua execução é efectuada concorrentemente com os algoritmos de controlo da plataforma móvel.

3. Abordagem

3.1 – Calibração automática da câmara

Como foi referido anteriormente, o essencial numa calibração é conseguir relacionar correctamente um conjunto de pontos da imagem e do mundo. Como na aplicação em causa se assume que as câmaras estão posicionadas perpendicularmente ao plano do chão, a transformação que relaciona estes dois conjuntos de pontos é composta por uma translação, uma rotação e um factor de escala.

Para se efectuar a correspondência entre pontos da imagem e do mundo optou-se pelo uso da grelha abaixo, que deve ser posicionada no chão sob a câmara. É de notar que o ponto a vermelho assinala o que se assume como o centro do referencial do mundo e as rectas a azul os eixos XX e YY (ortogonais).

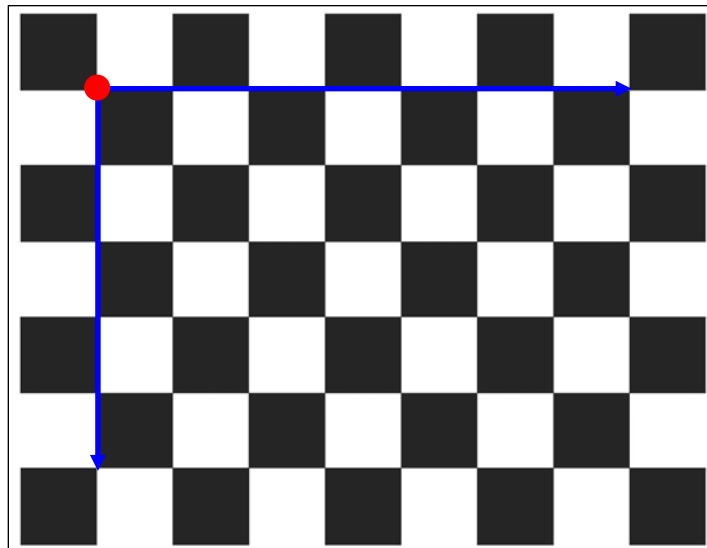


Figura 1 – Grelha utilizada para efectuar a calibração da câmara, com centro do referencial do mundo e eixos.

O padrão apresentado acima foi escolhido devido às características muito interessantes que possui, como facilidade de segmentação em preto e branco, periodicidade segundo dois eixos ortogonais e elevado número de pontos (cantos dos quadrados) com

distanciamento fixo e conhecido. Esta última característica permite determinar de forma trivial a posição de todos os pontos no referencial do mundo. Assumindo a origem sobre a bola a vermelho, sabendo que aresta dos quadrados mede 6 cm e considerando apenas pontos que não pertencem à fronteira da grelha, a posição espacial dos pontos é expressa pela seguinte matriz:

$$P_{\text{mundo}} = \begin{bmatrix} (0,0) & (0,6) & (0,12) & (0,18) & (0,24) & (0,30) & (0,36) \\ (6,0) & (6,6) & (6,12) & (6,18) & (6,24) & (6,30) & (6,36) \\ (12,0) & (12,6) & (12,12) & (12,18) & (12,24) & (12,30) & (12,36) \\ (18,0) & (18,6) & (18,12) & (18,18) & (18,24) & (18,30) & (18,36) \\ (24,0) & (24,6) & (24,12) & (24,18) & (24,24) & (24,30) & (24,36) \end{bmatrix}$$

Figura 2 – Matriz com a posição espacial dos pontos da grelha no referencial do mundo.

Para calcular a calibração da câmara falta obter uma matriz com a localização dos pontos correspondentes na imagem. Este problema é bastante mais complexo que o primeiro. Existem ferramentas muito poderosas para detecção de rectas em imagens, pelo que a determinação da posição dos pontos vai ser feita indirectamente, através da intercepção das linhas que pertencem à grelha. O processo de detecção de rectas é complexo, podendo ser separado em 5 partes: segmentação, filtragem, transformada de *Hough*¹, agrupamento e refinamento.

Na primeira parte a imagem captada é segmentada para preto e branco. Esta transformação reduz a complexidade da procura, de um domínio de \mathfrak{R}^3 (imagem a cores) para \mathfrak{R}^2 (imagem binária), eliminando toda a informação cromática (que é irrelevante para a procura) e diferenciando os tons de preto e branco da grelha.

Na segunda parte são aplicados filtros à imagem binária que possuem objectivos distintos. Os primeiros destinam-se a reduzir o ruído introduzido pela segmentação. Seguidamente suavizam-se e expandem-se os quadrados da grelha, para que estes se sobreponham. É finalmente aplicado um detector de contornos e realçadas as rectas resultantes, ficando as linhas da grelha visíveis.

¹ Consultar o Anexo A

Após o pré processamento descrito, é aplicada a transformada de *Hough*¹ à imagem. O espaço resultante é referenciado pela inclinação e distância à origem das rectas da imagem, correspondendo os máximos às rectas mais destacadas. É assim possível, através da análise deste espaço, retirar as rectas com maior probabilidade de pertencerem à grelha.

A parte de agrupamento permite agrupar as rectas obtidas da transformada de *Hough* em classes. O critério de ordenamento é a inclinação entre elas, sendo agrupadas numa mesma classe todas as rectas que possuam inclinações dentro da gama:

$$Gama = [-\Delta + \theta_{central}; \Delta + \theta_{central}]$$

Onde $\theta_{central}$ é a inclinação central da classe e Δ é a largura das classe. Este procedimento permite que rectas paralelas pertencentes à grelha, mas que devido à distorção radial ou outro tipo de perturbação foram captadas com inclinações diferentes, sejam agrupadas com sucesso. Dentro de cada classe são ainda removidas linhas que se interceptem, normalmente resultantes de múltiplas detecções da mesma recta e calculada a periodicidade das linhas na imagem.

O refinamento consiste na determinação das duas classes de rectas que contém as linhas da grelha. Este processo baseia-se no número mínimo de linhas que estas classes devem possuir, no facto das classes terem que estar desfasadas 90° e de terem que possuir periodicidades iguais. Caso não sejam determinadas duas classes que verifiquem estas características o processo de calibração automática termina.

Tendo as classes com as linhas da grelha, são calculados os pontos resultantes da sua intercepção, validados com base no padrão que os rodeia e extrapolados pontos intermédios cuja falta seja detectada. Caso esta fase termine com sucesso é calculada a calibração de matriz recorrendo ao Método dos Mínimos Quadrados² e ao Modelo para a Câmara³.

¹ Consultar o Anexo A

² Consultar o Anexo B

³ Consultar o Anexo C

3.2 – Identificação do robot numa imagem

A identificação do robot na imagem é feita através da detecção de duas marcas. A figura seguinte mostra a plataforma móvel que se pretende localizar, com as marcas colocadas sobre ela, para que sejam visíveis pelas câmaras colocadas no tecto do cenário.

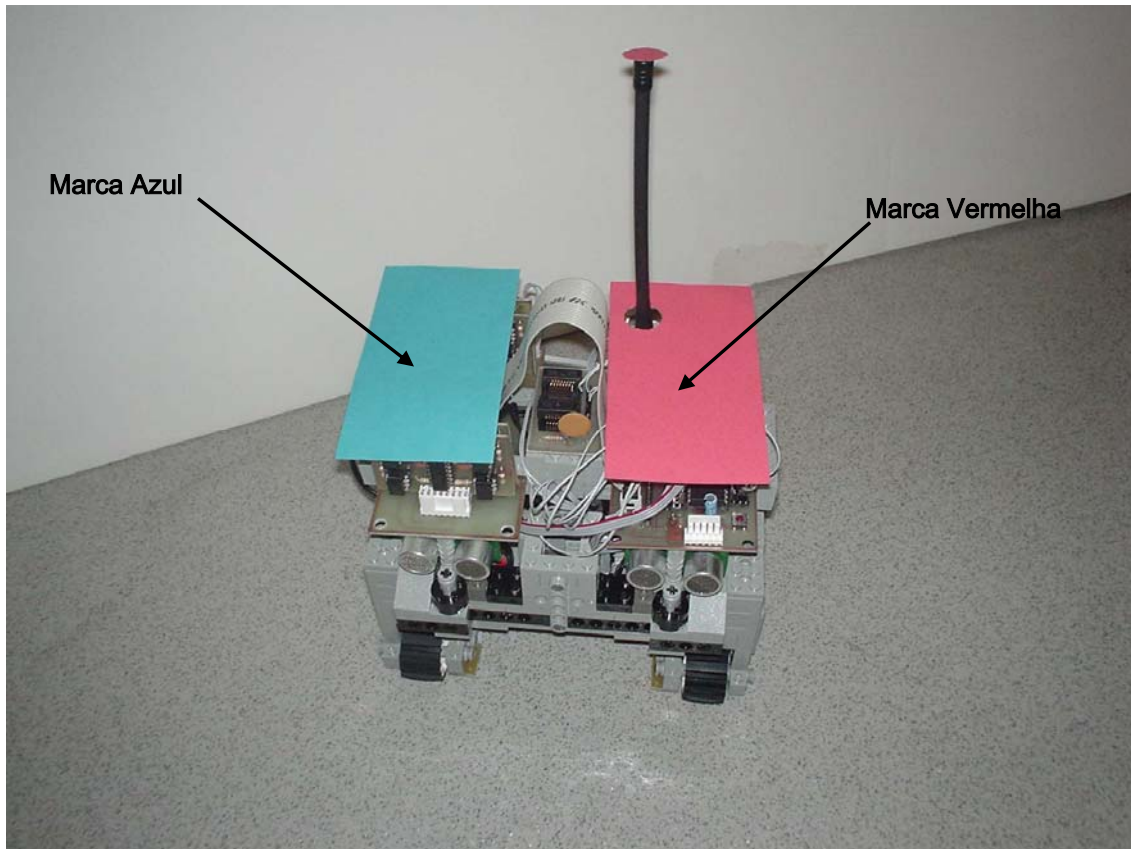


Figura 3 – Plataforma móvel com as duas marcas a detectar.

Como já foi referido, o processo de identificação tem que ser muito célere, uma vez que é executado concorrentemente com os algoritmos de controlo do robot. Este facto obriga ao delineamento de estratégias de procura eficientes e de rápido processamento. A escolha do espaço de procura é um passo fundamental neste processo. Embora exista um elevado número de espaços, com características muito distintas, optou-se pelo YCbCr. As componentes deste espaço são a luminosidade (Y), que quantifica a energia de cada píxel e as crominâncias azul (Cb) e vermelha (Cr), que são calculadas através da diferença entre os

valores das componentes azul e vermelha e o valor de referência, respectivamente. Para a aplicação em causa este espaço apresenta duas características muito vantajosas: a separação das cores vermelho e azul em componentes diferentes, o que permite a sua detecção através da localização dos máximos destes espaços e a imunidade em relação à luminosidade, o que confere a robustez desejada para este factor ambiental. Note-se que as marcas escolhidas são vermelho e azul saturado, como se pode verificar na **Figura 3**, o que permitirá uma detecção fácil nas imagens captadas. Fica assim por resolver a questão de como excluir zonas da imagens que são vermelhas ou azul saturadas e que não são marcas do robot. Tendo como intuito localizar objectos na imagem, são aplicados filtros Sobel às componentes Vermelha, Verde e Azul e calculada a união dos resultados. A esta imagem com contornos são posteriormente aplicados filtros que removem ruído, dilatados os contornos e preenchidas todas zonas que estes circunscrevem. Uma vez que as marcas são saturadas, existe uma elevada probabilidade de que este pré processamento as identifique como objectos na imagem e remova grande parte do fundo. Esta imagem binária obtida é assim aplicada às cromaticidades antes de se efectuar a procura dos máximos, obtendo-se um número reduzido de objectos que são potenciais marcas. A validação destes objectos é feita tendo em conta dois factores, distância entre marcas e área das mesmas. Estas grandezas são conhecidas uma vez que as câmaras estão calibradas e as dimensões da plataforma móvel são conhecidas. Caso no final desta procura exista um número de pares de marcas diferentes de um, que verifique as condições impostas, considera-se que falhou a detecção do robot.

Finalmente, a posição do robot é dada pela média dos centros de massa das marcas, que é calculada por:

$$P_{robot} = \frac{C_{vermelha} + C_{azul}}{2}$$

$$C_{vermelha} = \frac{1}{N_{vermelha}} \cdot \sum_{i \in \text{vermelha}} (x_i, y_i) \text{ e } C_{azul} = \frac{1}{N_{azul}} \cdot \sum_{i \in \text{azul}} (x_i, y_i)$$

Onde $N_{vermelha}$ e N_{azul} são o número de píxeis da marca vermelha e azul, respectivamente.

A determinação da orientação do robot é feita em relação aos eixos do referencial da imagem, considerando-se os 0° quando esta está na horizontal. A expressão seguinte efectua o cálculo da orientação em radianos:

$$\theta = \tan^{-1} \left(\frac{C_{vermelha}^Y - C_{azul}^Y}{C_{vermelha}^X - C_{azul}^X} \right) + (-1)^{lado} \cdot \frac{\pi}{2} \quad [rad]$$

Onde *lado* é especificado pelo utilizador e define de que lado do robot está cada uma das marcas.

4. Implementação

4.1 – Calibração automática da câmara

4.1.1 – Pré-processamento

Como descrito na abordagem, a detecção de rectas é antecedida por um pré processamento que tem como objectivo realçar as linhas da grelha. O primeiro passo é segmentar a imagem captada em preto e branco. Este procedimento elimina toda a informação cromática o que diminui a complexidade da procura e diferencia os tons de preto e branco da grelha, permitindo uma identificação mais fácil desta. As figuras seguintes ilustram um exemplo de segmentação de imagem. É de notar que a imagem segmentada (binária) foi invertida.

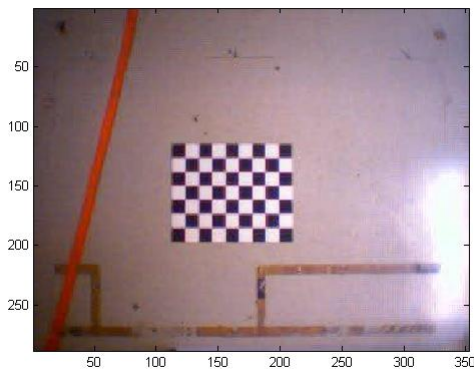


Figura 4 – Imagem original.

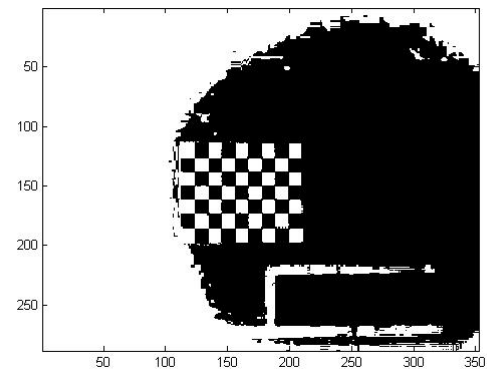


Figura 5 – Imagem segmentada.

O *threshold* usado na segmentação é nitidamente um parâmetro crítico, que neste caso é determinado pela função `graythresh` do *MatLab*. Esta implementa o método de Otsu, que gera um histograma e calcula o *threshold* que maximiza a variância entre classes (BCV).

Seguidamente são aplicados dois filtros a toda a imagem, com o intuito de reduzir o ruído introduzido pela segmentação. O primeiro é designado por *clean* e remove os píxeis rodeados de 0's. O segundo chama-se *spur* apaga os píxeis isolados numa vizinhança 4. A

função que implementa estes filtros é a `bwmorph`. As **Figuras 6, 7 e 8** ilustram o resultado da aplicação destes filtros à zona da grelha na imagem segmentada:

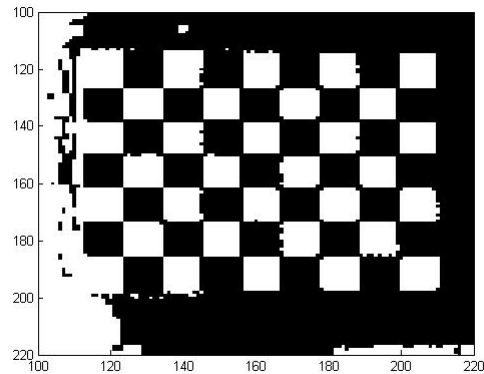


Figura 6 – Ampliação da zona da grelha na imagem segmentada.

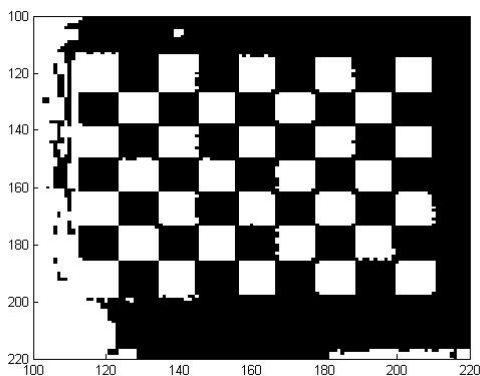


Figura 7 – Aplicação do filtro *clean* à imagem segmentada.

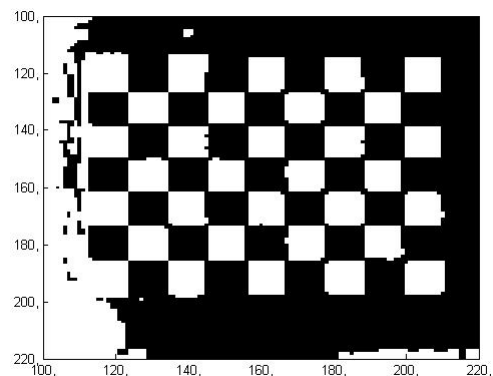


Figura 8 – Aplicação do filtro *spur* à imagem anterior.

Seguidamente suavizam-se os quadrados da grelha, para que a dilatação a realizar posteriormente não realce o ruído aí existente. São assim aplicadas duas operações morfológicas, uma referenciada como *majority* e outra por *open*. A primeira atribui a um píxel o valor da maioria (vizinhança 8) que o rodeia e o segundo efectua uma erosão seguida de uma dilatação, com o elemento estruturante seguinte:

$$\text{elemento} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Ambas as operações são efectuadas pela função de *MatLab* `bwmorph`. As Figuras 9 e 10 são relativas à aplicação destas operações à imagem da Figura 8.

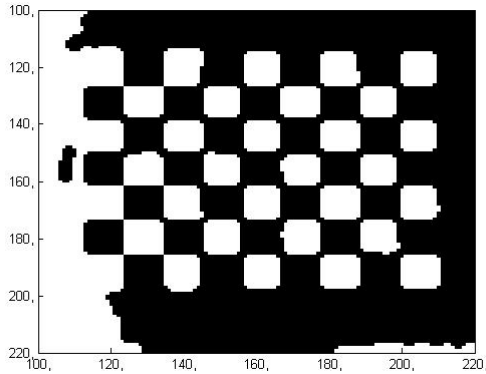


Figura 9 – Aplicação da operação morfológica *majority*.

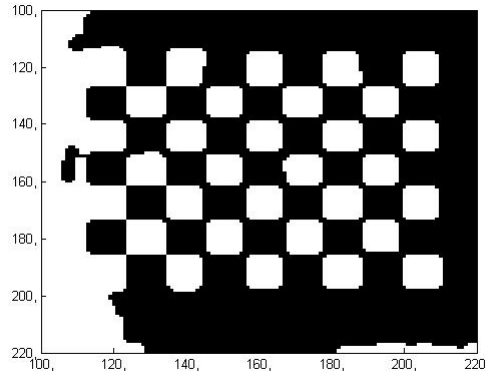


Figura 10 – Aplicação da operação morfológica *open*.

Como se pode verificar na Figura 10, as arestas dos quadrados das grelhas encontram-se suavizados, mas também com os cantos arredondados devido ao *majority*. Este facto não é no entanto relevante, uma vez que a dilatação que se aplica de seguida atenua este efeito. O elemento estruturante desta operação morfológica é igual ao usado anteriormente. Após a dilatação, os quadrados da grelha ficam sobrepostos. Usa-se então a função `edge`, que vai permitir obter os contornos dos quadrados justapostos, que são as linhas da grelha. A função `edge` é aplicada com o método *Zero-Cross*, onde se calcula a segunda derivada numa vizinhança reduzida de cada píxel e se retorna 1 quando esta cruza o valor zero e 0 caso contrário. É usual aplicar-se uma máscara de suavização 2D, antes de calcular as segundas derivadas, que tipicamente é dada por:

$$G(x, y) = e^{-\frac{x^2 + y^2}{\sigma^2}}$$

A máscara de convolução completa, que inclui já o cálculo das segundas derivadas, e que a função `edge` aplica para o método *Zero-Cross* é assim dada por:

$$h(x, y) = \frac{1}{\sigma^2} \cdot \left(\frac{x^2 + y^2}{\sigma^2} - 1 \right) \cdot e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

Os resultados das duas operações descritas acima são os apresentados seguidamente:

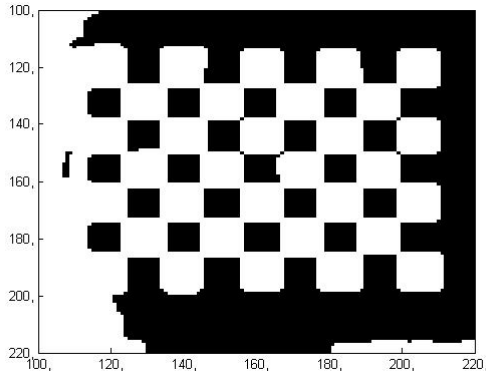


Figura 11 – Aplicação da operação morfológica *dilate*.

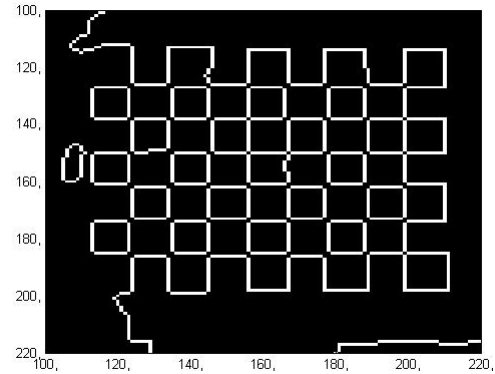


Figura 12 – Aplicação do detector de contornos.

As últimas duas operações do pré processamento visam realçar as linhas da grelha apresentadas na Figura 12. A primeira é uma dilatação em tudo semelhante à já aplicada e a segunda é um *close*. Esta última efectua nova dilatação seguida de erosão, usando o elemento estruturante apresentado anteriormente, que preenche pequenas falhas residuais nas linhas. As figuras seguintes ilustram o resultado destas operações:

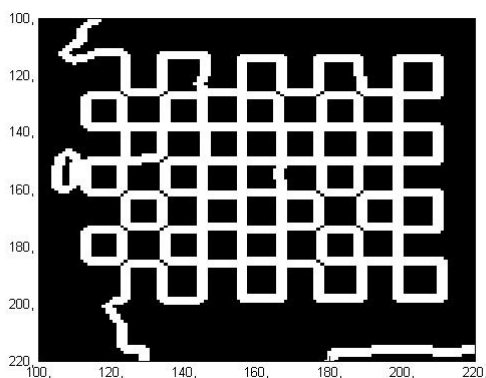


Figura 13 – Aplicação da operação morfológica *dilate*.

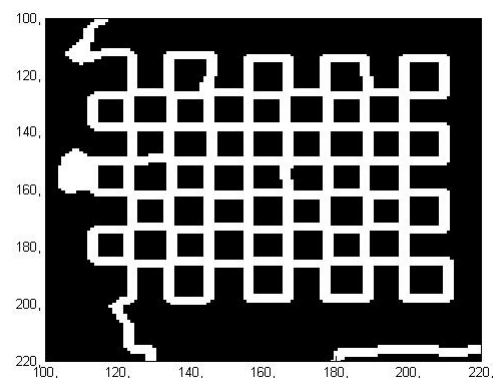


Figura 14 – Aplicação da operação morfológica *close*.

Finalmente, a Figura 14 representa a imagem sobre a qual vai ser aplicada a transformada de *Hough* para detectar as linhas da grelha.

4.1.2 – Cálculo da Transformada de Hough

A detecção de rectas na imagem gerada pelo pré processamento é realizada recorrendo ao Espaço de *Hough*. Este é construído pela Transformada de *Hough*¹, que de uma forma simplista se pode descrever como uma operação que projecta todas as rectas que passam por cada ponto da imagem num espaço que as acumula. Este espaço percorrido em (ρ, θ) que representam respectivamente a distância e o ângulo entre a origem e um vector perpendicular às rectas que passam em cada ponto. A detecção de rectas da imagem neste espaço resume-se então à procura de máximos, onde cada par $(\rho, \theta): Hough(\rho, \theta) > Hough(\rho + \delta_\rho, \theta + \delta_\theta)_{\delta_\rho, \delta_\theta > 0}$ determina univocamente os parâmetros de uma recta.

Seguidamente está ilustrado o Espaço de *Hough* gerado para a imagem completa da

Figura 14.

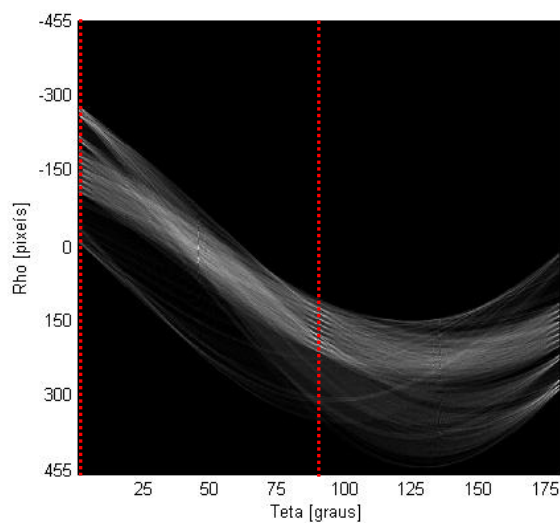


Figura 15 – Transformada de *Hough* da imagem gerada no pré processamento.

Como se pode observar, o eixo dos XX está em graus, de 0 a 180°, representando a inclinação de qualquer recta da imagem e o eixo dos YY indica a distância em píxeis do vector

¹ Consultar o anexo A para mais informações

perpendicular à recta que cruza a origem. A distância em píxeis vai até aproximadamente -455 a 455, uma vez que as dimensões da imagem são 288 x 352, logo:

$$|\rho| = \sqrt{288^2 + 352^2} \approx 454,8 \text{ píxeis}$$

Os tracejados vermelhos sobre o Espaço de *Hough* identificam, com grande probabilidade, as linhas da grelha, uma vez que interceptam um número suficiente de máximos (presumíveis rectas) com um espaçamento aparentemente periódico, que é aproximadamente igual para ambos os tracejados. Note-se ainda que os tracejados distam exactamente 90° um do outro, o que indica que as rectas indicadas por estes são ortogonais entre si.

A construção do Espaço de *Hough*, a procura de máximos e a determinação dos parâmetros das rectas indicados por estes são operações implementadas, respectivamente, pelas seguintes funções de *MatLab*: `hough`, `houghpeaks` e `houghlines`. A estas são fornecidos parâmetros como a resolução do ρ e θ , o número de rectas a retirar do Espaço de *Hough*, entre outros, que permitem otimizar o desempenho da calibração automática e que podem ser especificados pelo utilizador. Para mais informações faça `help camera_calibration` na linha de comandos do *MatLab*.

4.1.3 – Agrupamento de rectas em classes e identificação das pertencentes à grelha

Como referido no anexo A, o espaço resultante da Transformada de *Hough* permite obter as rectas dominantes da imagem. Uma vez que não são impostas restrições ao panorama de fundo, é possível que sejam capturadas rectas que não pertencem à grelha. Existe ainda o problema, visível na **Figura 15**, de rectas horizontais serem detectadas nos 0° e 180° , pelo que uma procura segundo uma dada inclinação pode não captar todas as linhas com essa orientação. Assim foi necessário desenvolver um mecanismo de classificação que permita agrupar e seleccionar apenas as rectas de interesse. Usando o ângulo retornado pela Transformada de *Hough*, agrupam-se as rectas em classes, cuja largura pode ser definida pelo utilizador. Após estarem agrupadas, são removidas rectas que se interceptem dentro da mesma classe. Este procedimento é necessário para remover múltiplas detecções da mesma recta. No passo seguinte é calculada, para todas as classes com ocorrências, a Transformada de *Fourier* do Espaço de *Hough*, segundo o ângulo central das classes. Desta forma é possível obter a periodicidade das rectas de cada classe na imagem. A imagem seguinte ilustra o resultado da Transformada de *Fourier*, para as inclinações seleccionadas na **Figura 15** pelos tracejados vermelhos.

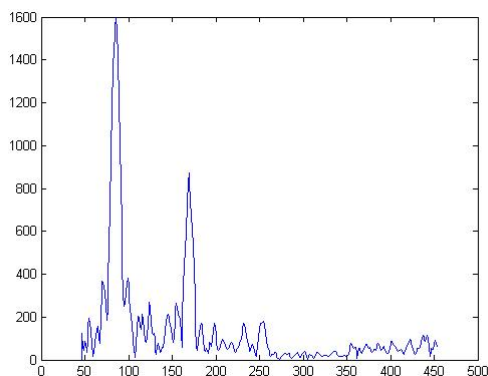


Figura 16 – Transformada de *Fourier* para $\theta = 89,5^\circ$.

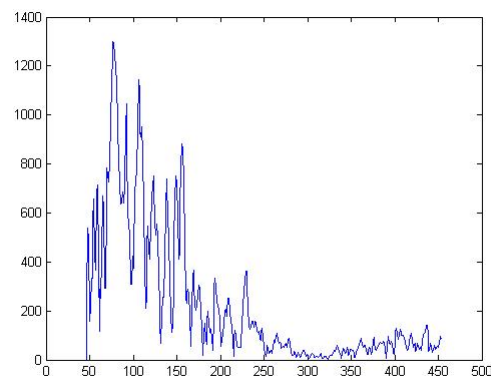


Figura 17 – Transformada de *Fourier* para $\theta = 179,5^\circ$.

O período espacial das rectas é proporcional ao inverso da frequência e é dado por:

$$T = \frac{2 \cdot |\rho|}{index_Max} \quad [píxeis]$$

Onde *index_Max* é o índice do valor máximo da Transformada de *Fourier*, que assinala a frequência fundamental. Como é possível observar nas **Figuras 16 e 17**, as baixas frequências são removidas, pois sabendo-se que a grelha possui 10 linhas segundo uma direcção, a sua frequência na imagem tem que ser superior a 10 Hz. Na **Figura 16** é possível visualizar particularmente bem a frequência fundamental e as suas harmónicas. Existem no entanto casos, onde devido a ruído, as harmónicas possuem amplitudes semelhantes à da fundamental, sendo por isso aplicada uma atenuação exponencial à transformada. O resultado desta operação encontra-se explicitado nas figuras seguintes:

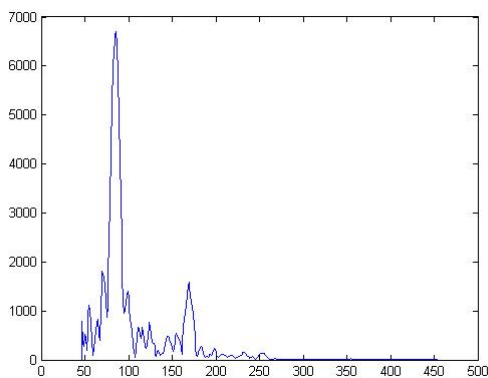


Figura 18 – Atenuação da Transformada para $\theta = 89,5^\circ$.

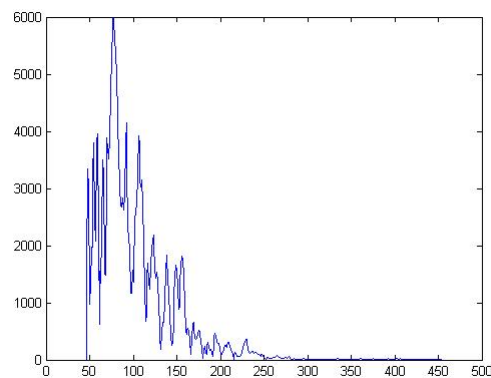


Figura 19 – Atenuação da Transformada para $\theta = 179,5^\circ$.

Como se pode verificar, a frequência fundamental encontra-se realçada, enquanto que as harmónicas são atenuadas, resolvendo-se assim o problema referido.

O período de cada classe, juntamente com o número de ocorrências e os ângulos entre estas é usado para determinar as rectas pertencentes à grelha. Caso não existam duas classes que verifiquem estas condições, o processo de calibração falha.

Os pontos da grelha são obtidos interceptando as rectas das classes seleccionadas, sendo a sua posição calculada em coordenadas homogéneas. Sendo a equação genérica de uma recta dada por:

$$y = m \cdot x + b \Leftrightarrow m \cdot x - y + b = 0$$

Que se pode expressar vectorialmente em coordenadas homogéneas por:

$$\begin{bmatrix} m & -1 & b \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = l' \cdot \tilde{x} = 0 \Leftrightarrow \begin{bmatrix} x & y & 1 \end{bmatrix} \cdot \begin{bmatrix} m \\ -1 \\ b \end{bmatrix} = \tilde{x}' \cdot l = 0$$

É possível provar que a intercepção de duas rectas, expressas em coordenadas homogéneas, é dada por:

$$\tilde{x} = l_1 \times l_2$$

Onde \tilde{x} é a intercepção das rectas l_1 e l_2 . As coordenadas cartesianas dos pontos podem ser facilmente obtidas através das equações:

$$x = \frac{\tilde{x}(1)}{\tilde{x}(3)} \text{ e } y = \frac{\tilde{x}(2)}{\tilde{x}(3)}$$

Nesta fase os pontos obtidos são validados, verificando-se se estes estão rodeados pelo padrão da grelha. Após esta verificação testa-se o número de pontos obtidos e extrapolam-se pontos interiores da grelha que não tenham sido detectados. A estimação da posição destes pontos é feita com base na média de todos os vizinhos, pelo que todos eles têm que ter sido detectados. Caso contrário a calibração automática termina. Tendo a correspondência entre os pontos da grelha no mundo e na imagem, procede-se à calibração da câmara.

4.1.4 – Determinação da matriz P

Denomina-se por matriz P a matriz que relaciona pontos da imagem com pontos do mundo. Na aplicação em causa, como a câmara incide ortogonalmente sobre o plano do cenário, basta uma homografia para transformar os pontos entre os dois referenciais. Esta transformação leva assim pontos 2D do plano da imagem para pontos 2D do plano do cenário e vice-versa. Estas operações são realizadas em coordenadas homogéneas, pelo que a matriz que a realiza tem dimensão 3x3. Do Modelo da Câmara¹, sabe-se que:

$$\tilde{x} = P \cdot \tilde{X}, \text{ onde } P = \begin{bmatrix} P_1^T \\ P_2^T \\ P_3^T \end{bmatrix}$$

Sendo \tilde{x} a posição do ponto na imagem e \tilde{X} a posição do mesmo ponto no mundo, ambos em coordenadas homogéneas. Reescrevendo a equação acima obtém-se:

$$\tilde{x} - P \cdot \tilde{X} = 0 \Leftrightarrow \begin{cases} x = \frac{P_1^T \cdot \tilde{X}}{P_3^T \cdot \tilde{X}} \\ y = \frac{P_2^T \cdot \tilde{X}}{P_3^T \cdot \tilde{X}} \end{cases} \Leftrightarrow \begin{cases} x \cdot P_3^T \cdot \tilde{X} - P_1^T \cdot \tilde{X} = 0 \\ y \cdot P_3^T \cdot \tilde{X} - P_2^T \cdot \tilde{X} = 0 \end{cases} \Leftrightarrow \begin{cases} x \cdot \tilde{X}^T \cdot P_3 - \tilde{X}^T \cdot P_1 = 0 \\ y \cdot \tilde{X}^T \cdot P_3 - \tilde{X}^T \cdot P_2 = 0 \end{cases}$$

Na equação acima x e y são as coordenadas do ponto na imagem. Explicitando o sistema acima na forma vectorial tem-se:

$$\begin{bmatrix} -\tilde{X}^T & [0 & 0 & 0] & x \cdot \tilde{X}^T \\ [0 & 0 & 0] & -\tilde{X}^T & y \cdot \tilde{X}^T \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix} \cdot \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} = 0 \Leftrightarrow M \cdot p = 0$$

É de notar que cada ponto gera duas linhas, sendo por isso necessário no mínimo 6 pontos para que o sistema esteja bem dimensionado. São retirados da grelha 48 pontos, pelo que a estimação dos parâmetros da matriz P vai ser feita usando o Método dos Mínimos

¹ Consultar o anexo C

Quadrados¹. O excesso de informação usado na estimação permite compensar por pequenos erros que tenham sido cometidos na estimação dos pontos da grelha na imagem.

A solução usando este método é dada pelo vector próprio associado ao menor valor próprio da matriz $M^T M$. Como é possível verificar pelo Modelo da Câmara², o último elemento da matriz P tem que ser 1, logo a relação entre o vector próprio p e a matriz P estimada é:

$$P = \frac{1}{p(9)} \cdot \begin{bmatrix} p(1) & p(2) & p(3) \\ p(4) & p(5) & p(6) \\ p(7) & p(8) & p(9) \end{bmatrix}$$

¹ Consultar o anexo B

² Consultar o anexo C

4.2 – Identificação do robot numa imagem

4.2.1 – Pré-processamento

O objectivo desta primeira fase é isolar as zonas da imagem onde existam potenciais marcas e efectuar a transformação para o espaço YCbCr, onde se irá proceder à procura das mesmas. O primeiro passo baseia-se na aplicação de filtros Sobel às componentes Vermelha, Verde e Azul da imagem. Esta operação calcula uma aproximação do gradiente, efectuando a convolução com as seguintes máscaras:

$$Sobel_{Horizontal} = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \text{ e } Sobel_{Vertical} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

O resultado da aplicação destes filtros a uma imagem exemplo é apresentado de seguida:

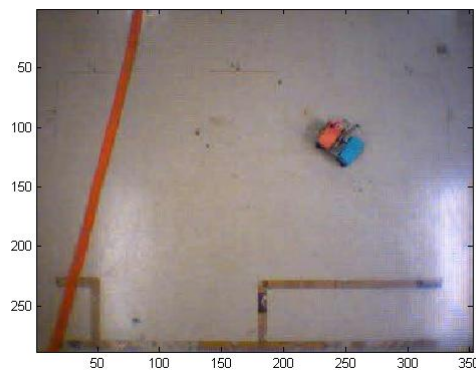


Figura 20 – Imagem original de onde se pretende retirar a posição e atitude do robot

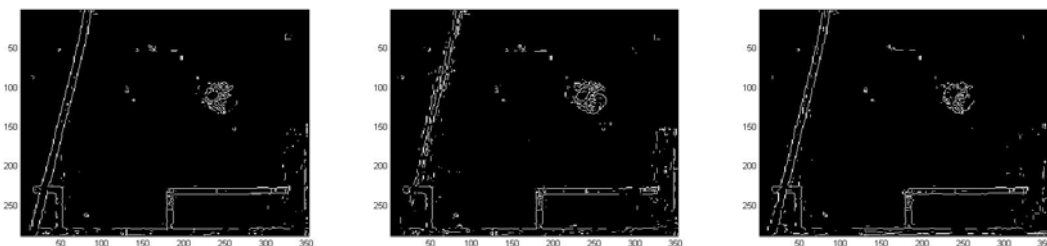


Figura 21 – Aplicação dos filtros Sobel às componentes Vermelha, Verde e Azul, respectivamente.

A união da filtragem nas três componentes permite assim obter os contornos de todos os objectos na imagem.

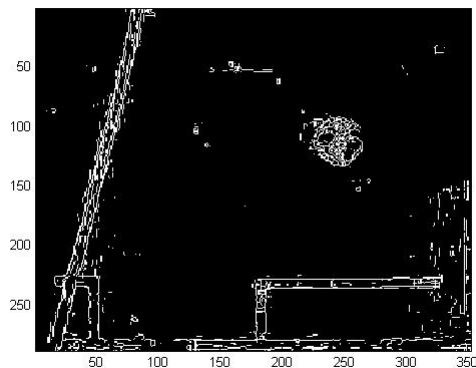


Figura 22 – União dos contornos detectados nas três componentes.

Como se pode verificar na figura acima, existem falhas nas linhas que limitam os contornos dos objectos. Pretendendo-se isolar estas zonas e excluir o resto, procede-se a uma dilatação das linhas, antes de se preencherem as áreas por eles circunscritas. A dilatação é implementada pela função `bwmorph`, e o seu resultado encontra-se explicitado abaixo.

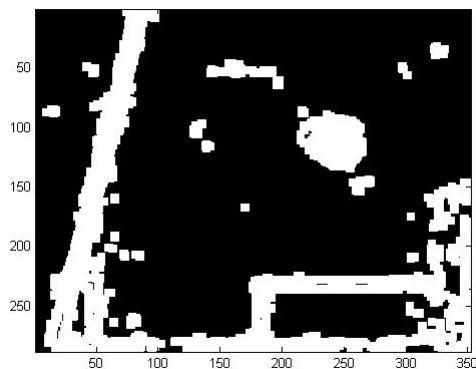


Figura 23 – Dilatação dos contornos da imagem.

O preenchimento das zonas que se encontram rodeadas é efectuado pela função `imfill`. O resultado da sua aplicação à imagem anterior é ilustrado de seguida.

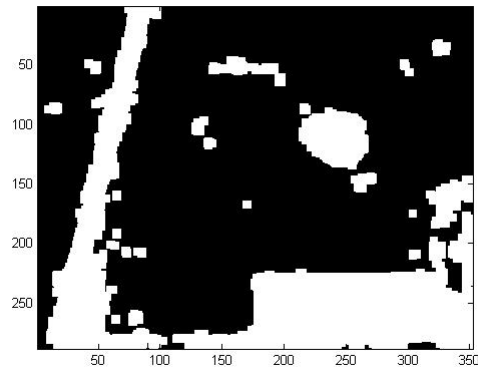


Figura 24 – Preenchimento das zonas delimitadas por contornos.

Como se pode verificar, o tratamento aplicado à imagem permitiu excluir grande parte do fundo, assinalando um grupo de zonas que devem ser analisadas para procurar as marcas. A imagem binária acima é então aplicada às crominâncias vermelha e azul de forma a limitar a procura nestas. O resultado desta intercepção é apresentado abaixo:

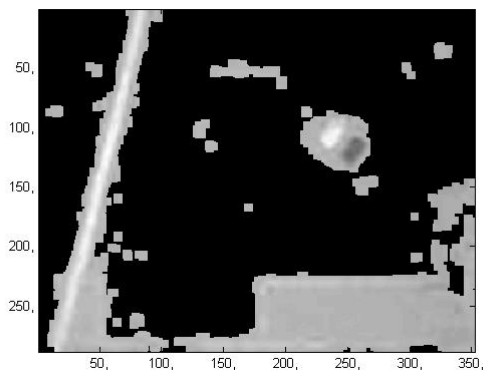


Figura 25 – Crominância vermelha da imagem.

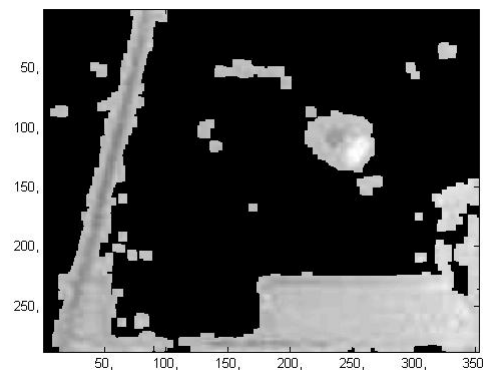


Figura 26 – Crominância azul da imagem.

É nestas imagens que vão ser procuradas as marcas instaladas na plataforma móvel.

4.2.2 – Identificação das marcas

Na identificação das marcas começa por se determinar o valor máximo na cromaticidade vermelha e azul. Estes valores vão servir de referência para estabelecer um *threshold* que será usado para segmentar as respectivas componentes. As imagens seguintes ilustram o resultado desta segmentação:

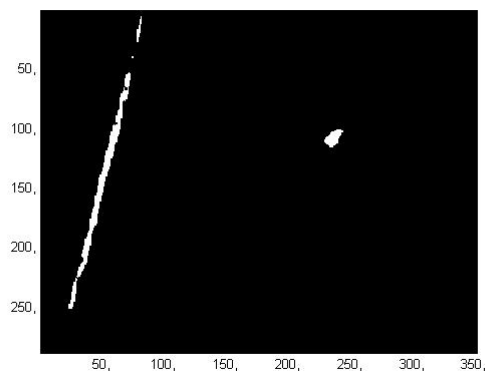


Figura 27 – Segmentação na cromaticidade vermelha.

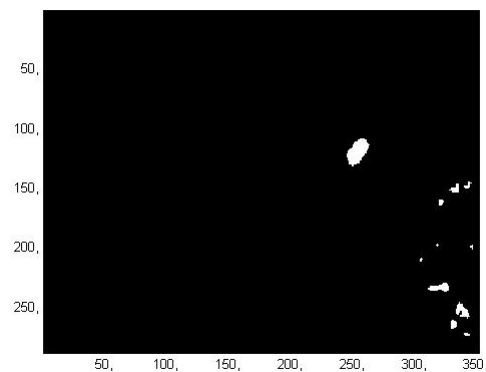


Figura 28 – Segmentação na cromaticidade azul.

As zonas resultantes deste processo representam, respectivamente, as áreas saturadas de vermelho e azul. A identificação das marcas é realizada com base na distância entre os seus centróides e nas áreas que estas devem ocupar. A obtenção destes dados é feita pela função `bwlabel`, que cataloga as áreas e calcula o seu centróide e a sua área. Estes dados são posteriormente analisados e identificam-se os pares que respeitam as condições descritas. Caso não seja encontrado nenhum par ou mais que um, considera-se que falhou a detecção do robot.

4.2.3 – Determinação da posição e orientação do robot

A determinação da posição do robot é feita usando os centróides das marcas identificadas. A expressão usada no cálculo da posição é:

$$P_{robot} = \frac{C_{vermelha} + C_{azul}}{2} \quad [Píxeis]$$

Onde $C_{vermelha}$ e C_{azul} são os centróides das marcas vermelha e azul.

A determinação da atitude do robot é realizada por geometria, considerando-se que o robot está orientado segundo a recta ortogonal à que passa pelos centróides das marcas. É ainda utilizado o parâmetro (*SIDE*) que permite ao utilizador especificar de que lado do robot é que está cada uma das marcas. A expressão que permite calcular a atitude do robot é então:

$$\theta = \tan^{-1} \left(\frac{C_{vermelha}^Y - C_{azul}^Y}{C_{vermelha}^X - C_{azul}^X} \right) + (-1)^{lado} \cdot \frac{\pi}{2} \quad [rad]$$

5. Resultados

De modo a demonstrar as potencialidades das aplicações desenvolvidas, são apresentados de seguida exemplos que procuram testar o seu comportamento em diversas situações.

5.1 – Calibração automática da câmara

Os resultados apresentados foram obtidos usando os parâmetros por defeito da função `camera_calibration`. O utilizador pode no entanto variar estes valores, devendo para isso consultar ajuda da função. A origem do referencial do mundo pode também ser configurada para se situar no canto da grelha mais próximo/afastado da origem do referencial do mundo. Nos exemplos apresentados é sempre seleccionado o mais próximo.

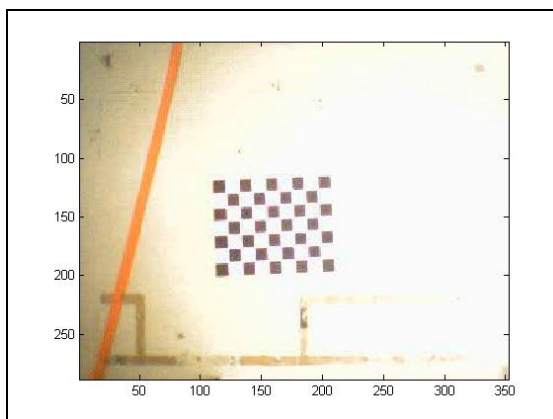


Figura 29 – Imagem original a analisar.

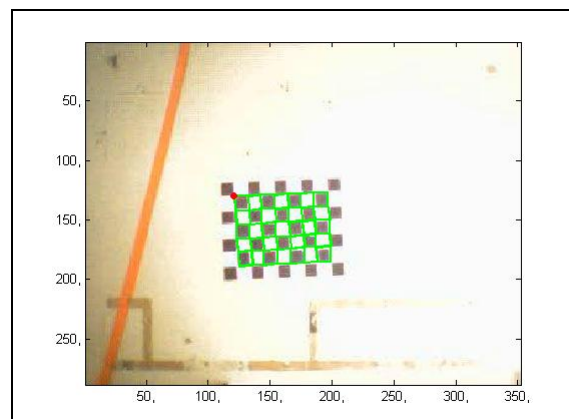


Figura 30 – Rectas da grelha identificadas.

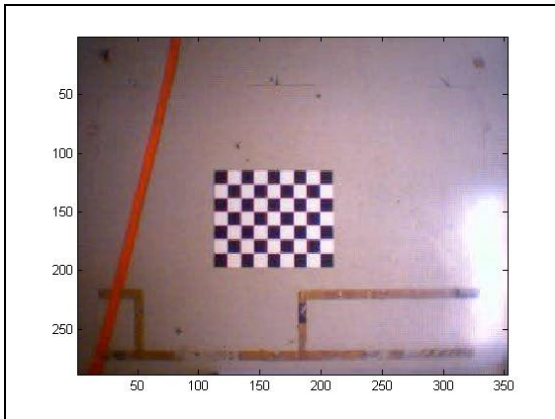


Figura 31 – Imagem original a analisar.

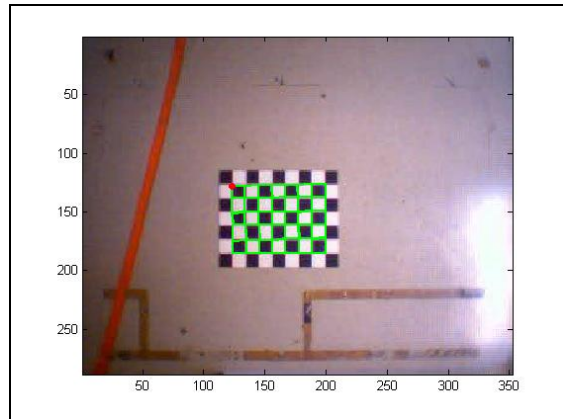


Figura 32 – Rectas da grelha identificadas.

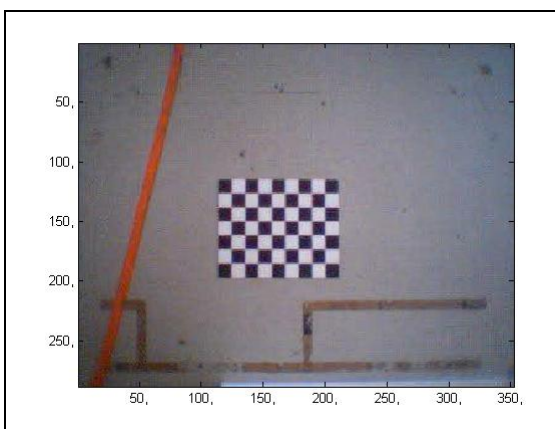


Figura 33 – Imagem original a analisar.

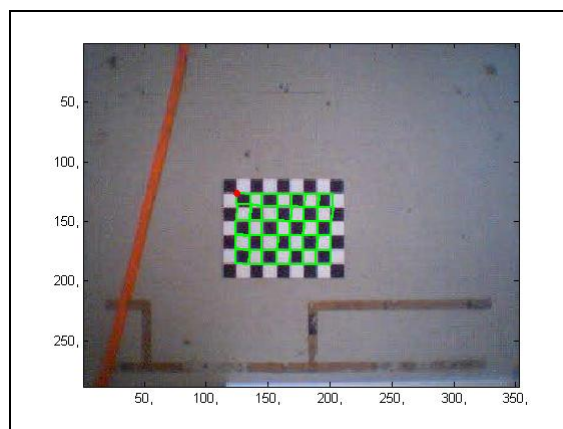


Figura 34 – Rectas da grelha identificadas.

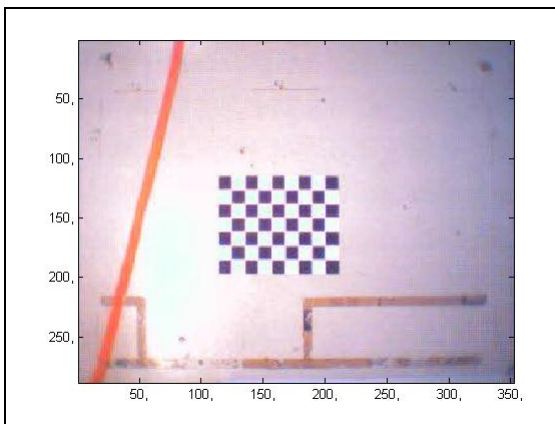


Figura 35 – Imagem original a analisar.

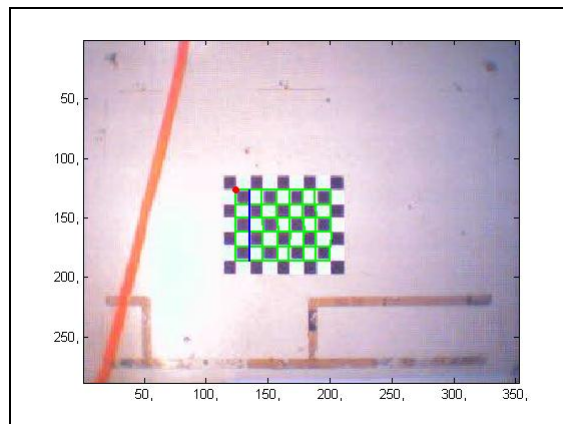


Figura 36 – Rectas da grelha identificadas.

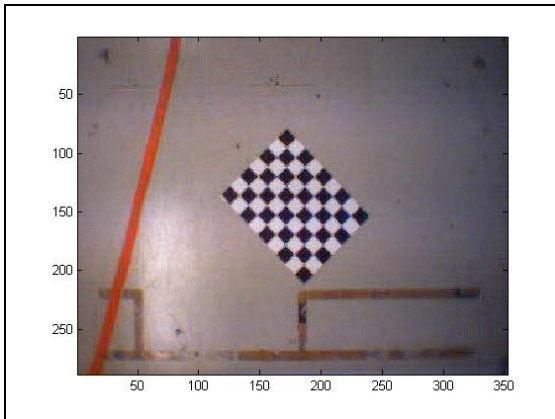


Figura 37 – Imagem original a analisar.

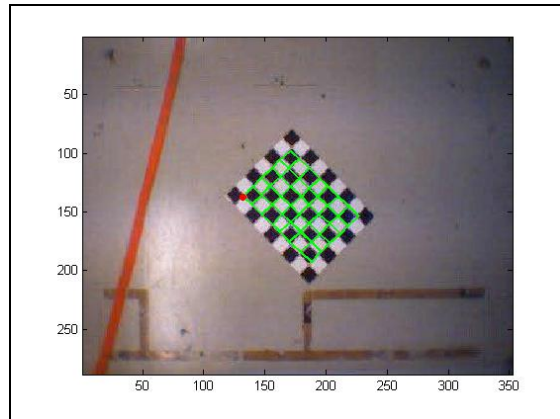


Figura 38 – Rectas da grelha identificadas.

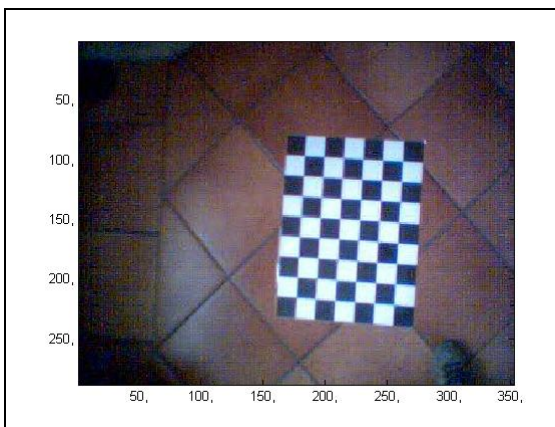


Figura 39 – Imagem original a analisar.

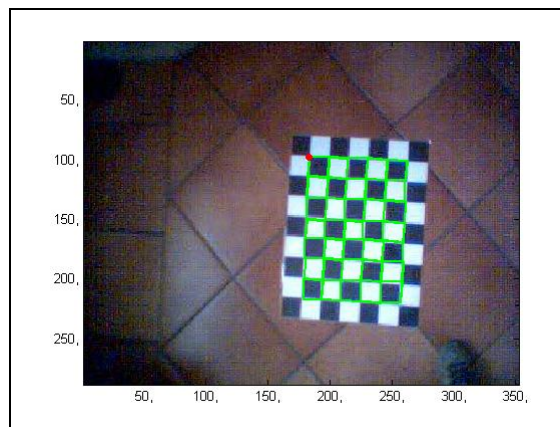


Figura 40 – Rectas da grelha identificadas.

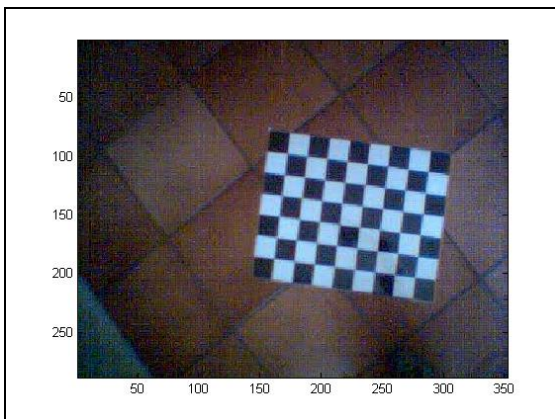


Figura 41 – Imagem original a analisar.

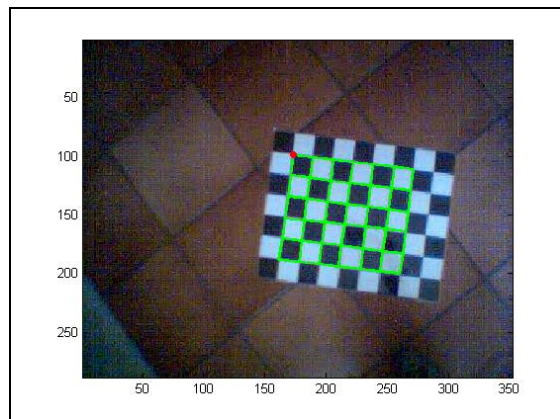


Figura 42 – Rectas da grelha identificadas.

5.2 – Identificação do robot numa imagem

A identificação do robot pressupõe uma calibração prévia da câmara, uma vez que o utilizador tem que especificar as dimensões das marcas e a distância entre elas em píxeis. Os argumentos de entrada da função `thor_detect` são descritos em detalhe na ajuda da mesma.

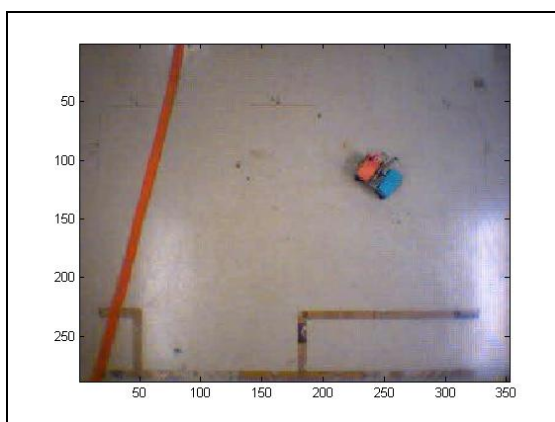


Figura 43 – Imagem original a analisar.

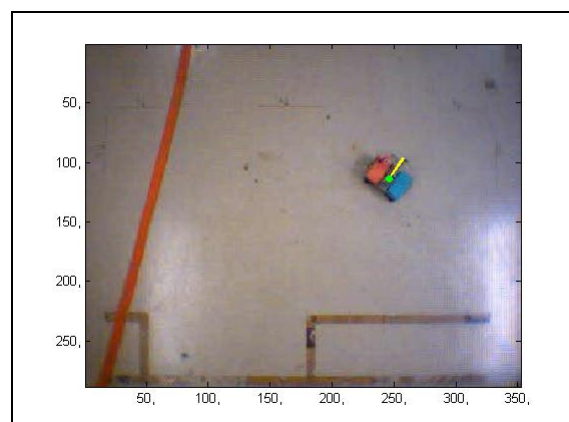


Figura 44 – Apresentação da posição e atitude do robot.

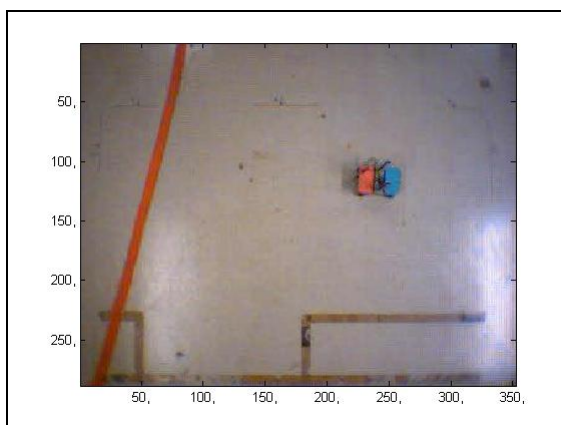


Figura 45 – Imagem original a analisar.

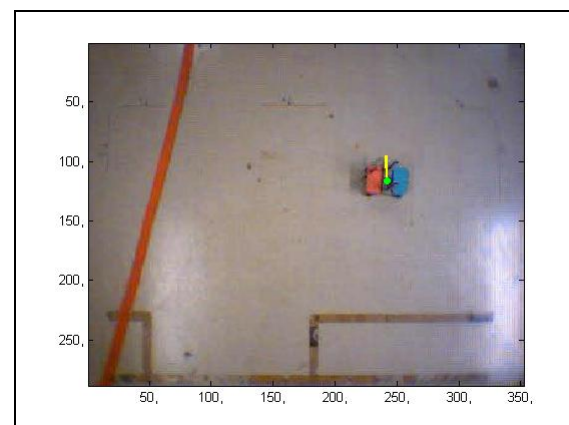


Figura 46 – Apresentação da posição e atitude do robot.

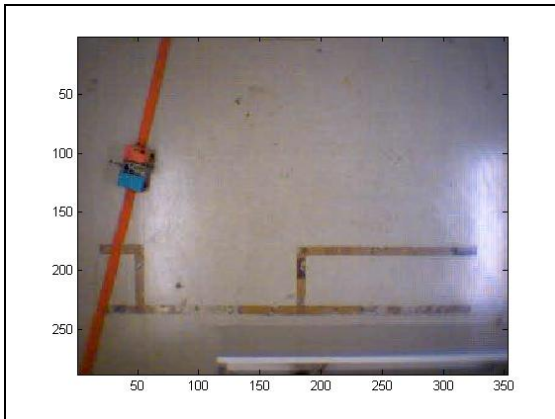


Figura 47 – Imagem original a analisar.

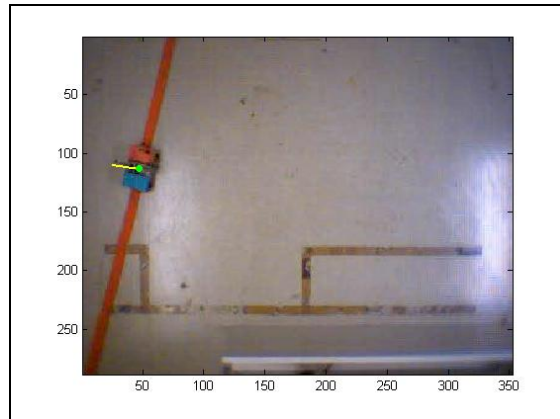


Figura 48 – Apresentação da posição e atitude do robot.

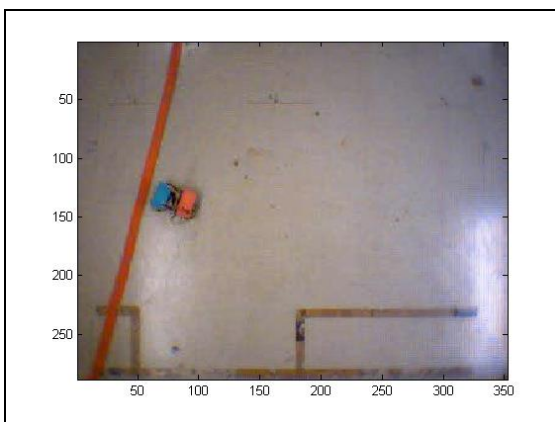


Figura 49 – Imagem original a analisar.

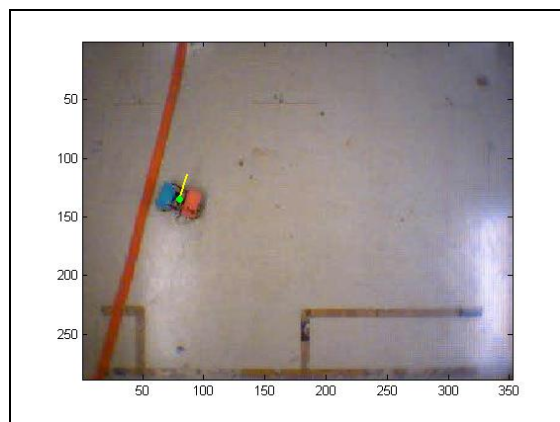


Figura 50 – Apresentação da posição e atitude do robot.

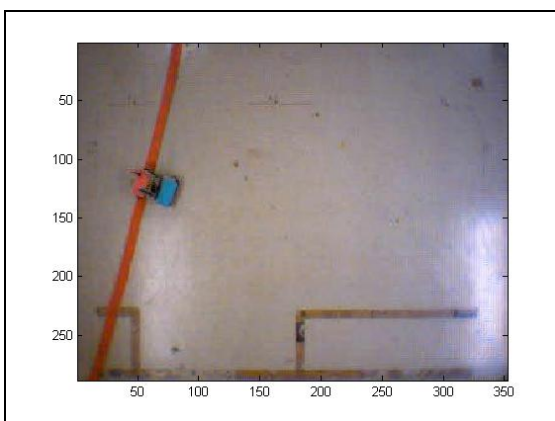


Figura 51 – Imagem original a analisar.

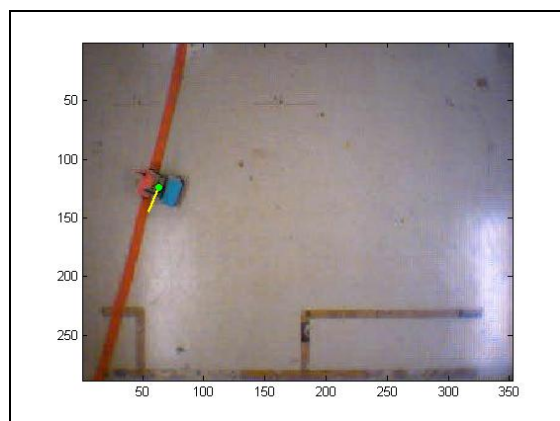


Figura 52 – Apresentação da posição e atitude do robot.

6. Conclusões

Os resultados apresentados permitem realçar aspectos interessantes de ambas as aplicações.

Relativamente à calibração automática, quis-se averiguar qual a influência de quatro factores críticos no comportamento do sistema: existência de linhas dominantes no cenário, tons de fundo diferentes, luminosidades variadas e orientações da grelha distintas. O ruído nas imagens captadas não foi um factor testado individualmente, uma vez que este é introduzido na sua maioria na captação e quantização das imagens, não podendo ser controlado. Relativamente ao primeiro factor referido, escolheram-se fundos para todos os exemplos onde existissem rectas dominantes. Como se pode verificar, estas foram excluídas com sucesso em todos os casos, o que demonstra a eficácia do processo de agrupamento em classes e do refinamento das mesmas. A variação de tons de fundo é perceptível da **Figura 37** para a **Figura 39**, não se tendo no entanto verificado alterações significativas no comportamento do algoritmo. Este facto era de esperar uma vez que a segmentação inicial proporciona uma imunidade elevada a este factor. Relativamente à luminosidade, foram usadas iluminações distintas e disparados *flashes* sobre diversas zonas da imagem, como se pode observar nas **Figuras 29** à **38**. Uma vez mais a segmentação e o pré processamento efectuados revelaram-se eficientes na atenuação da influência destas perturbações, tendo a calibração terminado correctamente. Por fim, quanto ao estudo da influência da orientação da grelha, observou-se que esta não altera o comportamento da aplicação, sendo detectada sem problemas nos casos apresentados nas **Figuras 37, 39** e **41**. É de notar que na **Figura 36** uma das rectas interiores da grelha não é identificada, sendo extrapolada através da posição dos vizinhos, motivo pelo qual esta é apresentada a azul. Repare-se ainda que no último caso, **Figura 42**, a calibração falha devido à não detecção de um linha exterior da grelha. Isto ocorre, pois alguns dos quadrados pretos em seu redor possuem píxeis brancos, o que impede o seu correcto reconhecimento. Neste caso a recta não é extrapolada, visto que não se trata de uma recta

interior, logo os seus pontos não estão rodeados de vizinhos. Conclui-se assim que o programa desenvolvido para calibração automática revela um desempenho muito aceitável em condições diversas, falhando apenas em alguns casos onde as restrições impostas são demasiado exclusivas. Estas garantem no entanto que a grelha nunca é detectada incorrectamente, o que se considerou preferível, em detrimento de a identificar neste conjunto mais restrito de casos.

Na aplicação referente à determinação da posição e atitude da plataforma móvel, usando marcas coloridas, os problemas a estudar são diferentes. Estes referem-se principalmente à existência de zonas na imagem com cores do mesmo tom e tão saturadas como as das marcas e à influência da orientação do robot. A questão da luminosidade não foi abordada tão exaustivamente, uma vez que o espaço de procura usado possui uma imunidade elevada a este factor. Pode no entanto observar-se nas **Figuras 47, 49 e 51** que a presença de reflexos intensos das lâmpadas, em redor das marcas, não impossibilita o seu correcto reconhecimento. Relativamente à existência de zonas na imagem com a mesma cor das marcas, testaram-se várias situações. Nas **Figuras 43 e 45** o robot é posicionado longe desta zona e orientado segundo direcções diferentes. Como se pode verificar, este é identificado correctamente, não se sentindo a influência da faixa vermelha. Constata-se ainda que a atitude do robot não altera a eficácia do algoritmo, facto que também é comprovado pelos três exemplos seguintes. Nas **Figuras 47, 49 e 51** a plataforma móvel é posicionada junto da faixa vermelha. Pretende-se assim averiguar se a proximidade daquela zona leva o algoritmo a não identificar a marca com a mesma cor, ou a detectá-la incorrectamente. Tal não acontece, demonstrando-se o bom desempenho na separação de objectos distintos e dos parâmetros usados na análise destes. Embora a aplicação tenha detectado correctamente a posição e orientação do robot em todos os testes realizados, este algoritmo é menos robusto que o apresentado anteriormente. Isto deve-se ao facto da velocidade de processamento desejada não permitir o emprego de técnicas mais elaboradas. É no entanto de mencionar que embora não se garanta que o robot é sempre identificado, a probabilidade deste ser detectado incorrectamente é praticamente nula, uma vez que é necessário que as zonas reconhecidas

como marcas cumpram critérios muito restritos que só se podem verificar uma vez em toda a imagem.

Apêndices

Apêndice A – Transformada de Hough

A transformada de *Hough* na sua forma original permite a detecção de rectas numa imagem. No entanto, esta pode ser modificada para permitir a detecção de outro tipo de formas.

O princípio básico da transformada de *Hough* consiste no pressuposto de que por um ponto arbitrário passam infinitas rectas, cada uma com uma orientação diferente. Para determinar se uma recta une dois pontos, é necessário criar uma representação para uma recta. Uma recta pode ser representada por dois parâmetros, uma distância ρ e um ângulo θ que correspondem, respectivamente, à distância e ao ângulo entre a origem da imagem e um vector perpendicular à recta. A aplicação da transformada de *Hough* a uma imagem resulta na criação do chamado espaço de *Hough*. A formulação matemática da transformada de *Hough* é a seguinte:

$$\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta)$$

Para cada ponto na imagem é calculado um comprimento, resultante da variação do ângulo e é traçada uma sinusóide, resultante da representação do ângulo em função do comprimento $\theta(\rho)$. A esta representação dá-se o nome de Espaço de *Hough*.

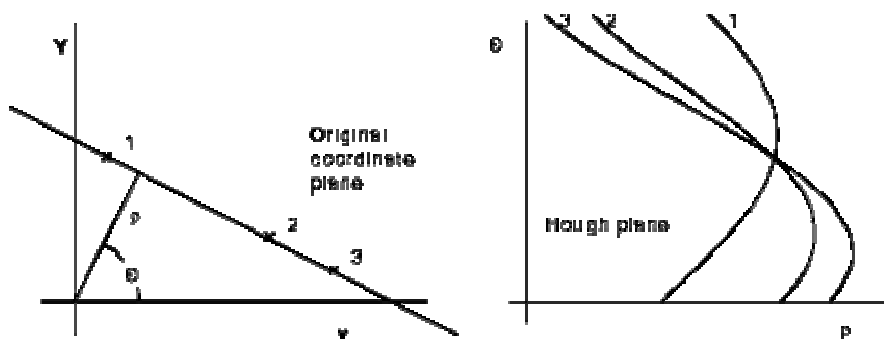


Figura A.1 – Coordenadas e espaço de Hough.

Como se pode verificar na figura anterior, no Espaço de *Hough* o ponto de intersecção das diferentes sinusóides indica que os respectivos pontos pertencem à mesma recta no espaço cartesiano.

Apêndice B – Método dos Mínimos Quadrados

O Método dos Mínimos Quadrados é uma técnica que permite encontrar os parâmetros que melhor ajustam um modelo aos dados existentes. Na aplicação em causa, pretende-se solucionar um problema linear sob a forma:

$$M \cdot p = 0$$

Onde p é o vector com os parâmetros a estimar. A solução deste problema é equivalente à de encontrar o vector p que minimiza o resíduo r da equação:

$$M \cdot p = r$$

Impõe-se ainda, de forma a garantir a unicidade da solução, que o vector de parâmetros verifique a restrição:

$$|p|^2 = 1$$

Que vectorialmente pode ser descrita por:

$$p^T \cdot p - 1 = 0$$

Recorrendo ao Método dos *Multiplicadores de Lagrange*, o funcional de custo a minimizar fica dado por:

$$J(p) = p^T \cdot M^T \cdot M \cdot p + \lambda \cdot (p^T \cdot p - 1)$$

Derivando a expressão anterior em ordem a p e igualando a zero, obtém-se:

$$\frac{dJ(p)}{dp} = 2 \cdot M^T \cdot M \cdot p - 2 \cdot \lambda \cdot p = 0$$

Que reordenando fica,

$$M^T \cdot M \cdot p = \lambda \cdot p$$

Sabendo que a definição de vector próprio de uma matriz é dada por:

$$(A - I \cdot \lambda_i) \cdot v_i = 0 \Leftrightarrow A \cdot v_i = \lambda_i \cdot v_i$$

Comparando com a expressão obtida na página anterior, verifica-se:

$$M^T \cdot M \cdot p = \lambda \cdot p \Leftrightarrow A \cdot v_i = \lambda_i \cdot v_i$$

Logo as soluções do problema são dadas pelos vectores próprios da matriz $A = M^T \cdot M$. Substituindo $M^T \cdot M$ por λ_i no funcional de custo e sabendo que o valor da restrição se anula para a solução óptima, fica:

$$J(p) = p^T \cdot \lambda_i \cdot p$$

Pelo que o vector próprio que minimiza o funcional de custo é o que está associado ao menor valor próprio. Conclui-se assim que a solução do problema é dada pelo vector próprio associado ao menor valor próprio da matriz $M^T \cdot M$.

Apêndice C – Modelo da Câmara

O modelo de uma câmara é descrito por um conjunto de parâmetros intrínsecos (relacionados com a distância focal, factores de escala, etc.) e extrínsecos (relativos rotações e translações). Assim a calibração consiste na estimação destes parâmetros a partir de um conjunto de pontos conhecidos, i.e., efectua-se a correspondência entre um conjunto de pontos do mundo e da imagem:

$$\{(X_i, x_i), i = 1, \dots, n\} \quad X_i \in \mathbb{R}^3, x_i \in \mathbb{R}^2$$

Onde x_i corresponde às coordenadas na imagem e X_i às coordenadas no mundo.

Em coordenadas homogéneas, o modelo interno da câmara é dado por: $\tilde{x} = K \cdot \tilde{x}'$, onde K é uma matriz que converte coordenadas métricas em píxeis. O modelo externo é dado por: $\tilde{x}' = \begin{bmatrix} R & t \end{bmatrix} \cdot \tilde{X}$, onde R é a matriz de rotação e t uma matriz de translação. Conjugando as duas expressões anteriores, obtém-se uma nova equação que define completamente o modelo da câmara: $\tilde{x} = P \cdot \tilde{X}$, com $P = K \cdot \begin{bmatrix} R & t \end{bmatrix}$. Tipicamente, a matriz P é de dimensões 3x4, uma vez que relaciona pontos 3D do mundo (vector 4x1 em coordenadas homogéneas) com pontos 2D na imagem (vector 3x1 em coordenadas homogéneas). Na aplicação em causa, pretende-se apenas relacionar pontos 2D, do plano do chão, com pontos 2D, do plano de imagem. O modelo da câmara pode assim ser estimado por uma homografia, que possui dimensão 3x3 e permite os graus de liberdade necessários.