

Faculdade de Engenharia da Universidade do Porto



## Ensino Rápido de Manipuladores Industriais

Carlos Eduardo Pinto Teixeira

Dissertação/Relatório de Projecto realizada(o) no âmbito do  
Mestrado Integrado em Engenharia Electrotécnica e de Computadores  
Major Automação

Orientador: Prof. Dr. António Paulo Gomes Mendes Moreira

Co-orientador: Prof. Dr. Paulo José Cerqueira Gomes da Costa

Julho de 2009

© Carlos Teixeira, 2009

# Agradecimentos

Quero deixar uma palavra a todos os que me ajudaram ao longo do meu percurso acadêmico.

Aos meus orientadores Prof. Dr. António Paulo Moreira e Prof. Dr. Paulo Gomes Costa pela oportunidade de investigação no grupo de Controlo e Robótica (SDPO) da FEUP, e pelo apoio e suporte durante a execução deste trabalho.

Um agradecimento muito especial ao Paulo Malheiros e Héber Sobreira, pela ajuda, tempo e partilha de conhecimentos ao longo da execução deste trabalho final.

Aos meus amigos que até aqui me acompanharam e aos que aqui criei.

À Marlene por toda amizade e paciência.

Aos meus Pais, que sempre me apoiaram desde o início e pelas condições dadas para que tudo até aqui tivesse sido possível.

A todos eles, um Muito Obrigado.



# Resumo

A programação por demonstração (PpD) tem sido um tema cada vez mais abordado, com inúmeras abordagens e estratégias para a robótica. Em especial na área de programação de manipuladores industriais, devido à sua complexidade de programação e cada vez mais escassez de tempo para o fazer.

Principalmente para as pequenas e médias empresas (PMEs) este tipo de programação é muito vantajoso, pois muitas delas não possuem recursos de engenharia para que possam efectuar as alterações e implementar novas funcionalidades para os seus sistemas robóticos.

Estes sistemas de PpD têm-se mostrado cada vez mais robustos, eficazes e flexíveis.

Neste trabalho é apresentada uma estratégia para a PpD de manipuladores industriais, recorrendo ao sistema Sincrovision recentemente desenvolvido na FEUP.



# Abstract

The programming-by-demonstration (PbD) has been an issue increasingly addressed, with numerous approaches and strategies for robotics. In particular, for an industrial robotic system, because of the programming complexity and short time to do it.

Specially for small and medium enterprises (SMEs) this type of programming has great advantages, because most of these companies do not have the engineering resources to make changes or add new functionalities to their robotic systems.

These PbD has proven more robustness, efficient and flexible way to programming.

This work presents a strategy and one more PbD idea for programming industrial robotic systems, using one system recently developed at FEUP - the Sincrovision.





# Índice

Agradecimentos .....	iii
Resumo .....	v
Abstract .....	vii
Índice .....	ix
Lista de figuras .....	xii
Abreviaturas e Símbolos .....	xv
Capítulo 1    Introdução .....	1
1.1    Estado da Arte .....	2
1.2    Objectivos .....	4
1.3    Estrutura do documento.....	5
1.4    Notação usada no documento .....	5
Capítulo 2    Visão Artificial .....	7
2.1    Introdução.....	7
2.2    Modelo <i>pinhole camera</i> .....	8
Capítulo 3    Visão Estereoscópica .....	19
3.1    Introdução.....	19
3.2    Algumas noções .....	20
3.3    Modelo <i>pinhole camera</i> e o sistema estereoscópico.....	22
3.4    Geometria epipolar.....	24
3.5    Calibração do sistema de visão.....	27
3.5.1    O algoritmo de Tsai: versatile camera calibration technique.....	27
3.5.2    Eight-point algorithm.....	31

3.6	Reconstrução 3D .....	32
<b>Capítulo 4 Hardware.....</b>		<b>35</b>
4.1	Sistema estereoscópico .....	35
4.2	Marcadores Luminosos.....	37
4.3	Sistema de Sincronização .....	38
4.4	Manipulador industrial.....	39
4.5	Sistema Pneumático.....	40
4.6	Sensor de força .....	41
4.7	Máquina de Polimento .....	43
4.8	Sistema de monitorização de Corrente .....	43
4.9	Peças para polimento.....	48
<b>Capítulo 5 Sistema Sincrovision .....</b>		<b>49</b>
5.1	Calibração Tsai .....	49
5.2	Teste da Geometria epipolar .....	54
5.3	Teste reconstrução 3D .....	54
5.4	Posição e rotação da ferramenta.....	56
5.5	Validação de resultados .....	60
<b>Capítulo 6 Geração de trajectórias.....</b>		<b>65</b>
6.1	Exposição do problema.....	65
6.2	Abordagem adoptada .....	66
6.3	Instruções a usar no manipulador .....	66
6.3.1	Instrução <i>MOVL</i> .....	66
6.3.2	Instrução <i>MOVC</i> .....	67
6.4	Detecção de zonas de paragem.....	69
6.5	Detecção de zonas de velocidade constante.....	70
6.6	Detecção de rectas .....	72
<b>Capítulo 7 Relação Força vs Corrente eléctrica .....</b>		<b>75</b>
7.1	Hardware .....	75
7.1.1	Abordagem .....	75

Capítulo 8	Conclusões.....	79
8.1	Trabalho futuro .....	80
	Referências.....	83
Anexo A	Conceitos .....	85

## Lista de figuras

Figura 2.1.1 Câmara de vídeo .....	8
Figura 2.2.1 Representação típica do modelo <i>pinhole camera</i> .....	9
Figura 2.2.2: Modelo <i>pinhole camera</i> .....	9
Figura 2.2.3: Representação dos eixos coordenados no nosso modelo <i>pinhole camera</i> .....	10
Figura 2.2.4 vista de lado do modelo <i>pinhole camera</i> .....	11
Figura 2.2.5: Eixo de coordenadas $X_1Y_1$ no ponto principal e $X_2Y_2$ num dos cantos da imagem. ....	12
Figura 2.2.6 Referencial do mundo.....	14
Figura 3.1.1 Focando o polegar, obtemos a visão de duas bandeiras. Retirado de [15]. ....	19
Figura 3.1.2 Focando a Bandeira obtêm-se a visão de dois polegares. Retirado de [15]......	20
Figura 3.2.1 Efeito de Perspectiva .....	21
Figura 3.2.2: Efeito de Oclusão. Retirado de [15]......	21
Figura 3.2.3 Efeito de Sombra. Retirado de [15]. ....	22
Figura 3.3.1 Sistema de visão estereoscópica .....	23
Figura 3.4.1 Modelo de um sistema estereoscópico .....	25
Figura 3.5.1 Particularidades assumidas no algoritmo de Tsai. Retirada de [14] .....	28
Figura 3.5.2 <i>Setup</i> Calibração Tsai .....	28
Figura 3.5.3 Plano de calibração de 88 pontos. ....	30
Figura 3.6.1 Triangulação ideal .....	33
Figura 3.6.2 Triangulação sem intercepção dos raios .....	33
Figura 4.1.1 Câmara do sistema estereoscópico .....	35
Figura 4.1.2 Entradas USB e <i>Trigger</i> das câmaras usadas.....	36

Figura 4.1.3 Sistema estereoscópico montado .....	36
Figura 4.2.1 Suporte construído para colocar os marcadores .....	37
Figura 4.2.2 Disposição dos marcadores luminosos na ferramenta .....	38
Figura 4.3.1 Placa genérica Atmega164P .....	38
Figura 4.4.1 Manipulador Motoman .....	39
Figura 4.5.1 Gripper usado .....	40
Figura 4.6.1 Sensor de força JR3 .....	41
Figura 4.6.2 Representação de 3 dos 6 graus de liberdade mensuráveis pelo JR3. Imagem retirada da folha de características do sensor. ....	42
Figura 4.6.3 Aplicação JR3 .....	42
Figura 4.7.1 Máquina de polir .....	43
Figura 4.8.1 Analisador de energia Mach Smart da Ducati .....	44
Figura 4.8.2 Aplicação Mach Monitor .....	44
Figura 4.8.3 Circuito de Condicionamento de sinal .....	45
Figura 4.8.4 Saída em tensão do LEM .....	45
Figura 4.8.5 Valores Máximos e mínimos do circuito de leitura de corrente .....	46
Figura 4.8.6 Aspecto da aplicação de monitorização de corrente .....	47
Figura 4.9.1 Peças para polimento .....	48
Figura 5.1.1 Eixos de rotação .....	50
Figura 5.1.2 Rotação segundo eixo Z- a) - vista para a câmara 1, b) vista para a câmara 2 ...	50
Figura 5.1.3 Começando a referir por cima - Teste 2, 4 e 6 efectuados para a calibração Tsai. ....	51
Figura 5.1.4 Captura dos planos - Teste 2 .....	52
Figura 5.1.5 Aplicação de <i>threshold</i> .....	52
Figura 5.1.6 Localização de Pontos de calibração.....	52
Figura 5.2.1 Linha epipolar calculada através do ponto seleccionado na imagem 1 .....	54
Figura 5.3.1 Teste reconstrução 3D .....	55
Figura 5.3.2 Efeito de Barril .....	55
Figura 5.3.3 Correção do efeito de barril .....	56
Figura 5.4.1 Modelo da ferramenta com marcadores .....	57
Figura 5.4.2 Eixos coordenados .....	58

Figura 5.4.3 Eixos coordenados Manipulador .....	59
Figura 5.5.1 Cubo de teste.....	62
Figura 5.5.2 Cubo deformado .....	62
Figura 5.5.3 Configuração do sistema Sincrovision .....	63
Figura 6.1.1 Caminho gerado pelo NURBS .....	65
Figura 6.3.1 Instrução <i>MOVL</i> .....	66
Figura 6.3.2 <i>Position Level</i> - retirada dos manuais da MOTOMAN .....	67
Figura 6.3.3 Resultado da instrução <i>MOVC</i> .....	67
Figura 6.3.4 Circunferências consecutivas .....	68
Figura 6.3.5 Circunferências consecutivas de forma correcta - figura retirada do manual de programação do manipulador .....	68
Figura 6.3.6 Instrução <i>MOVC</i> .....	68
Figura 6.4.1 Esferas em torno dos pontos de análise.....	70
Figura 6.4.2 Filtragem de pontos .....	70
Figura 6.5.1 Trajectória em recta .....	71
Figura 6.5.2 Diferentes segmentos da mesma recta com velocidades diferentes .....	71
Figura 6.6.1 Recta dos 3 primeiros pontos .....	72
Figura 6.6.2 Rectas calculadas dos 6 primeiros pontos .....	72
Figura 6.6.3 Trajectória final aproximada por rectas .....	73
Figura 7.1.1 Peça número 2 .....	76
Figura 7.1.2 Referencial para o teste .....	76
Figura 7.1.3 Gráficos Força/Corrente eléctrica .....	78
Figura 8.1.1 Disparidade binocular .....	85

# Abreviaturas e Símbolos

## Abreviaturas

**FEUP** – Faculdade de Engenharia da Universidade do Porto;  
**DEEC** – Departamento de Engenharia Electrotécnica e de Computadores;  
**ADC** – *Analog to digital converter*;  
**PpD** - Programação por demonstração;  
**PMEs** – Pequenas e médias empresas;  
**PbD** – *Programming by Demonstration*;  
**SMEs** – *Small and medium enterprises*;  
**CCD** – *Charge - coupled device* – Dispositivo de carga acoplado;  
**DLT** - *Direct Linear Transformation*;  
**SVD** – *Singular value decomposition*;  
**USB** – *Universal Serial Bus*;  
**CAD** – *Computer-Aided Design*;

## Símbolos

Símbolo	Significado
C	Centro óptico ou ponto de focagem;
$\Pi$	Plano de imagem;
$\Omega$	Plano focal;
p	Ponto principal;
$f$	Distância focal;
<b>I</b>	Matriz identidade;
$X_1Y_1$	Referencial de Coordenadas no plano de imagem;
$X_wY_wZ_w$	Referencial de Coordenadas do Mundo;
XYZ	Referencial de coordenadas da câmara;
ms	Milissegundo;
Hz	Unidades SI da frequência;
<b>R</b>	Matriz de Rotação;
T	Matriz de translação;
cm	Centímetro;
m	Metro;
kHz	QuiloHertz;
mA	miliampère





# Capítulo 1

## Introdução

Com a deslocação das grandes séries de produção para outros países, em toda a Europa, e particularmente em Portugal, restaram as pequenas séries com grande valor acrescentado e geralmente muito específico. Na produção robotizada recorrendo a manipuladores industriais este facto gerou um problema grave. O tempo de reprogramação de um manipulador robótico tornou-se muito elevado comparativamente ao tempo de produção. Na aquisição de um manipulador para uma linha de montagem previamente é necessário considerar factores tais como, o tempo necessário para o desenvolvimento do sistema, instalação e integração na empresa, custos de formação de operadores, gestão do projecto, manutenção, custos com actualizações do software e respectivas modificações. Assim, as técnicas de ensinamento rápido de manipuladores têm sido alvo de grande investigação e têm conduzido ao desenvolvimento de novas abordagens dos mesmos, nos últimos tempos.

O ensinamento por gestos ou por imitação de um operador foi abordado por alguns grupos de trabalho, no entanto a complexidade do *setup* necessário e a falta de robustez por vezes existente no sistema, impediram o desenvolvimento e a utilização frequente desta técnica de ensinamento. Com a nova tecnologia SincroVision, recentemente desenvolvida na FEUP<sup>1</sup> e em processo de patentear esta tecnologia; através do uso da visão estereoscópica para localização tridimensional no espaço de um marcador luminoso, será possível, utilizando mais do que um marcador luminoso sincronizados com câmaras, retirar a posição e rotação tridimensional da ferramenta. A ideia é acoplar estes marcadores na ferramenta de um operador de modo a registar-se num computador, diversos pontos da trajectória efectuada por este e posteriormente analisar estes movimentos tentando transmiti-los ao manipulador industrial. O sistema é bastante simples de instalar e extremamente robusto, o que possibilita a sua utilização em praticamente todos os ambientes industriais mais comuns.

Com estes pontos da trajectórias registados, pretende-se determinar uma sequência de pontos em linha recta ou em arco de circunferência 3D, que consiga aproximar a trajectória inicial o melhor possível segundo um dado critério. A aproximação à trajectória deve ser feita quer em termos de posição no espaço quer em termos de

---

<sup>1</sup> Pode ser consultado em:

[http://gnomo.fe.up.pt/~robotic/index.php?option=com\\_content&view=article&id=84&Itemid=94](http://gnomo.fe.up.pt/~robotic/index.php?option=com_content&view=article&id=84&Itemid=94)

velocidade. Após esta análise, a trajectória será traduzida para uma sequência de instruções na linguagem do manipulador robótico, criando assim, automaticamente, um programa capaz de executar a trajectória pretendida.

O trabalho foi desenvolvido com a colaboração das empresas NS- Máquinas que fabrica máquinas de polir e a empresa de manipuladores industriais MOTOMAN.

## 1.1 Estado da Arte

O uso de manipuladores industriais tem vindo a desempenhar um papel cada vez mais importante no sector industrial, por vezes não sendo maior a sua entrada no mercado devido à complexidade de programação destes dispositivos. O problema está na dificuldade de passar a sabedoria e experiência de um operador para o manipulador. A programação por demonstração (PpD) é um dos rumos que se está a tomar para a programação deste tipo de robôs. A motivação para este trabalho vai nesse sentido, pois a imitação da trajectória humana tem as seguintes vantagens:

- 1) PpD simplifica a programação do robô. Apenas é necessário efectuar o movimento desejado.
- 2) O movimento humano tem implícito muito potencial no que diz respeito à realização das tarefas, (exemplo: área de trabalho, direcções preferenciais que devem ser tomadas e regiões/movimentos que devem ser evitados) que seria difícil de representar através de outro método.
- 3) Passagem de velocidades recomendadas para determinados movimentos.
- 4) Habilita qualquer pessoa a operar e programar um manipulador industrial mesmo que esta não possua qualquer conhecimento de programação.

O crescente desenvolvimento dos computadores, assim como o aumento da capacidade de processamento e de armazenamento dos mesmos, têm aberto portas para a utilização de algoritmos mais complexos e robustos na área da robótica.

O principal objectivo da PpD é permitir a programação de trajectórias para o manipulador num curto espaço de tempo e de uma forma intuitiva. Diferentes abordagens para a PpD têm sido investigadas, diferindo entre si no tipo de manipuladores, estratégias e dispositivos usados. Sobre dispositivos e estratégias temos por exemplo o uso:

- da fala;
- de um guia;
- de um *joystick*;
- de desenhos;
- por gestos;
- por imitação / observação;
- de uma luva antropomórfica;

Yusuke Maeda et al (2008) [1] propõem que a PpD seja efectuada recorrendo a um guia preso na ponta do manipulador juntamente com um sensor de força acoplado ao guia. Neste trabalho, a aproximação ao problema passa por ter acesso ao ambiente de trabalho, pegar no guia e movimentar o robô no espaço envolvente indicando os pontos inicial e final do objecto a mover – todos os movimentos são guardados. De seguida, o manipulador calcula o espaço de trabalho e aquele que precisa para mover o volume. A melhor trajectória para o manipulador mover o objecto do ponto inicial até ao final, é então planeada.

Aleotti J et al (2006) [2] propõem que o manipulador funcione como uma aproximação a um braço humano, usando uma fita moldável para que seja possível determinar a sua forma – o mesmo tipo de sensor que tem sido usado em aplicações na área de realidade virtual para introduzir movimentos humanos nas personagens virtuais. Neste trabalho existem 3 fases: fase de treino, fase de aprendizagem e fase de decisão.

Na primeira fase, o robô efectua uma série de movimentos e o operador copia-os. Os valores da fita são guardados. De seguida, o operador controla o robô em tempo real, movendo o seu braço. O robô tenta executar os movimentos, baseando-se nos valores adquiridos na primeira fase. Na fase de decisão, o operador avalia se o movimento executado pelo manipulador é o pretendido. Se assim for, dá-se por finalizada a aprendizagem; caso contrário volta-se a repetir o algoritmo até obter resultados favoráveis.

O uso de luvas antropomórficas, para a programação de manipuladores é sugerida por Ekvall, S. et al (2005) [3] e por Aleotti J et al (2006) [2]. No primeiro caso, é estudada a maneira de como pegar nos objectos – objectos de diferentes tamanhos e de diferentes formas: esféricos, cilíndricos, discos. No operador, é colocada a luva para segurar nos objectos pretendidos, através de modelos de Markov, rotação e translação da mão ou ambos simultaneamente. É retirada essa informação e transferida para o manipulador.

No estudo de Aleotti J et al (2006) [2] todo o trabalho foi realizado em ambiente virtual, tendo-se usado a luva para interacção com os objectos virtuais. Neste ambiente, é permitido ao operador mover objectos, cujos movimentos serão posteriormente analisados. É então gerada uma trajectória aproximada ao movimento inicial através do algoritmo NURBS.

No caso da imitação dos gestos humanos através da observação, os trabalhos de R. Dillmann(2004) [4], Markus Ehrenmann et al [5], Manuel Lopes et al (2005) [6] demonstram que um sistema de visão é usado para observar os movimentos do operador para simples tarefas. Nestes estudos, o principal objectivo é tentar passar o conhecimento humano, realização de tarefas simples e complexas, para os robôs humanóides.

F. Nagata et al (2001) [7] usa um *joystick* em conjunto com um controlador de força. O operador pode controlar o manipulador indicando a direcção e orientação da ferramenta e ter a noção de força através da rigidez do *joystick*.

Com recurso à fala, temos o ensinamento de um tele-robô e a programação de um manipulador feita por Jayawardena,C. (2005) [8] e Pires, J.N.(2007) [9] respectivamente.

No primeiro estudo de Jayawardena,C. (2005) [8] o operador tem acesso ao panorama do ambiente de trabalho, e através da fala, descreve o objecto pretendido (forma, cor, tamanho). No caso de estas características estarem já na base de dados do robô, e recorrendo à visão artificial, este tentará reconhecer e recolher o objecto pretendido; caso contrário, o robô questionará o operário sobre a posição real do objecto e guardará posteriormente estas novas características na sua base de dados.

No segundo trabalho [9], é utilizada a fala como meio de programação do manipulador. A programação é feita através da demonstração, movendo-se o manipulador através de um guia preso na ponta do manipulador. É possível mover o manipulador nas direcções pretendidas e simultaneamente interagir com a programação das trajectórias através da fala (tirar a posição do robô, tempos de espera e tipos de movimento *MOVL*, *MOVJ*, entre outros), sendo exequível programar, de uma maneira simples e versátil, todo o trajecto do manipulador.

Quando se pretende mostrar a forma de um objecto, nada melhor do que tentar desenhá-lo, assim como era realizado antigamente pelos artesões. Neste contexto, Pires, J.N. (2007) et al [10] propõem o uso de canetas digitais (*da Anoto technology*) para a programação de tarefas. Juntamente com o uso destas canetas, são também usados ficheiros de CAD e interfaces para gerar código para os manipuladores, o que faz com que a programação destes manipuladores seja mais fácil e intuitiva.

O ensinamento por gestos tem começado a ganhar adeptos devido às vantagens já mencionadas na introdução. Pedro Neto et al (2009) [11] sugerem a programação por gestos, usando um sensor de movimento - acelerómetro de 3 eixos - embebido no comando da consola Wii.

## 1.2 Objectivos

Neste trabalho, é proposta mais uma estratégia à programação de robôs através da demonstração. É usada a visão artificial, mais especificamente um sistema de visão estereoscópico, para captar e observar os movimentos pretendidos. Concretamente, é utilizado um novo conceito, desenvolvido na FEUP, com a designação de Sincrovision. Trata-se de um sistema robusto de captura de movimento e posicionamento 3D. É composto por marcadores luminosos sincronizados com o sistema estereoscópico de visão artificial.

Inicialmente, o sistema permitia apenas o posicionamento 3D de um ponto. O objectivo é fazer com que o sistema consiga posicionar e capturar o movimento de uma ferramenta de trabalho que está a ser manejada por um operador. Após captura do movimento e análise do mesmo, é gerada a melhor trajectória para posteriormente o manipulador a imitar. Como área de teste, temos a tarefa de polimento, tomada como referência para a realização dos testes de alguns algoritmos.

Para atingir este objectivo final, foi efectuado um estudo do sistema estereoscópico já existente, calibração e melhoramento do mesmo, de forma a detectar a rotação da ferramenta. É também proposto um pequeno estudo sobre aproximação de pontos por rectas e arcos de circunferência.

### 1.3 Estrutura do documento

No capítulo número dois, sobre visão artificial, é feita uma introdução ao modelo *pinhole camera*, sendo apresentada alguma da geometria que o acompanha.

No capítulo três, faz-se uma introdução à visão estereoscópica, em que se explica a aplicação do modelo *pinhole camera* a um sistema estereoscópico. Na secção 3.4, é descrita a geometria epipolar sobre a qual um sistema estereoscópico assenta. Na secção 3.5, são apresentados dois algoritmos para a calibração de uma câmara e calibração de um sistema estereoscópico. No final do capítulo três, é exposto um algoritmo para a reconstrução 3D a partir de um sistema estereoscópico.

No capítulo número quatro é apresentado e descrito todo o *hardware* usado para este trabalho, assim como algumas das características relevantes sobre o mesmo. Muitas outras podem ser obtidas a partir das folhas de características fornecidas pelos fabricantes.

O capítulo número cinco descreve as aplicações desenvolvidas, e revela os resultados obtidos com os algoritmos implementados, para o cálculo da posição e rotação da ferramenta.

Para a geração de trajectórias, o capítulo seis descreve o pequeno estudo que se efectuou nesta área.

No capítulo sete é descrita a relação existente entre a força exercida pela peça a polir sobre o disco de polimento e a corrente eléctrica consumida pelo motor trifásico da máquina de polir.

As conclusões e trabalhos futuros estão presentes no capítulo oito.

O anexo A mostra os conceitos que foram utilizados ao longo de todo o documento.

### 1.4 Notação usada no documento

A notação usada neste documento para simplificar a sua leitura está definida da seguinte forma; As matrizes são representadas por letras maiúsculas e a negrito – **P**. Os vectores são representados por letras sozinhas e em maiúsculas – P.

A representação de escalares é feita em letra minúscula e em itálico – as coordenadas de um vector  $P$  tridimensional são  $(x,y,z)$ . A representação dos eixos de um referencial será feito em maiúsculas e os caracteres que os define, encontram-se todos juntos – as coordenadas do ponto  $P$  no referencial  $XYZ$  são  $(x,y,z)$ .

Palavras em inglês são identificadas em itálico – *pinhole camera*.

Ao longo dos capítulos de visão artificial vai ser usado um sistema de coordenadas homogéneas, para que se possam expressar certas operações através de multiplicação de matrizes – por exemplo a translação de um ponto, e que num sistema cartesiano não é possível.

O sistema de coordenadas homogéneas tem mais uma dimensão do que o espaço a que respeitam; Assim sendo, no espaço bidimensional têm a forma  $(x,y,w)$ , onde  $w$  representa um plano paralelo a  $XY$ . Ao longo do documento  $w$  será igual a 1. Então para um vector  $(x,y)$  no sistema cartesiano, este no sistema de coordenadas homogéneas é definido como  $(x,y,1)$ .

No caso das matrizes, para a passagem do sistema cartesiano, para o sistema de coordenadas homogéneas é adicionada uma última linha e/ou uma coluna (preenchida com zeros, excepto o último elemento é 1) à matriz original, de forma a que esta seja quadrada -  $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$  no sistema cartesiano; No sistema de coordenadas homogéneas a

matriz anterior passa a  $\begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & 1 \end{bmatrix}$ .

Quando se expressar alguma variável em coordenadas homogéneas esta é identificada com um til em cima da variável – ponto  $P_m$  em coordenadas homogéneas é  $\widetilde{P}_m$ .

Em qualquer altura em que se possa confundir qualquer tipo de letra/expressão representada em figuras, com a notação matemática aqui descrita, previamente é definido o que representa cada uma delas.

# Capítulo 2

## Visão Artificial

### 2.1 Introdução

Cada vez mais usada, a visão artificial é uma tecnologia que permite que as máquinas usem captação de imagens durante o seu funcionamento. Esta tecnologia baseia-se na teoria de construir sistemas artificiais que obtenham informações a partir de imagens capturadas através de câmaras fotográficas/vídeo. Esta captura pode ser efectuada em sequência de vídeo de uma só câmara, ou de múltiplas câmaras com diferentes perspectivas da mesma imagem.

Estes sistemas têm estado em constante entrada no mercado industrial, devido às suas características face à visão Humana. Este sistema não sofre de falsos julgamentos ou cansaço. No entanto, mesmo fora do uso industrial os sistemas de visão artificial têm sido usados como meio de interface humana para entretenimento ou facilidade de introdução de dados em alguns programas.

Este tipo de tecnologia tem sido introduzida em muitas áreas, entre as quais:

- Monitorização/controlo de processos – ex: contagem de pessoas, inspecção industrial, análise médica, modelação topográfica, Robôs industriais, veículos autónomos.
- Ajuda/Entretenimento – ex: como meio de dispositivo de entrada de alguns sistemas, no auxílio de pessoas com menos capacidades ou aumentar a facilidade de jogar em consolas de vídeo jogos.
- Física: ex: estudo da radiação electromagnética, em que esta é reflectida nas superfícies dos objectos e finalmente medida no CCD da câmara. São usados algoritmos de seguimento, reconhecimento de objectos, aprendizagem, estimação de movimento e restauração de imagem.

A implementação de um sistema de visão artificial não se apresenta como uma tarefa trivial pois a selecção dos componentes e a programação de todo o sistema reflecte as exigências particulares de cada aplicação. Adicionalmente as decisões que têm de ser tomadas devem reflectir aspectos mais abrangentes do que apenas os custos iniciais do investimento.

Em visão artificial podemos encontrar análise de imagem 2D (maioritariamente), nos últimos anos tem-se observado um crescente aumento na aplicação e estudo de imagens a 3D. Cada vez mais se tem tentado emular a visão biológica e a percepção visual dos Humanos (*Stereopsis*). Já há muitos anos se conhece o sistema de percepção visual humano e tem-se tentado passar este conhecimento para as máquinas.

Neste capítulo apenas vamos abordar o modelo de câmaras finitas e projectivas, tipo de câmaras usadas no trabalho.



Figura 2.1.1 Câmara de vídeo

As câmaras de vídeo são já um sensor muito usado na robótica, muito devido à grande quantidade de informação que é possível extrair da cada imagem (*frame*) que esta adquire. Para isso é necessário efectuar processamento da imagem obtida focado na informação que se pretende extrair dessa mesma imagem.

Modelo da câmara – O modelo *pinhole camera* descreve a relação matemática entre as coordenadas de um ponto 3D e a sua projecção na imagem do plano. Maior parte dos algoritmos desenvolvidos para visão computacional parte do princípio que o modelo da câmara usada segue o modelo *pinhole camera*.

## 2.2 Modelo *pinhole camera*

Este modelo não inclui distorções geométricas nem objectos desfocados provocados pelas lentes e apenas pode ser usado para uma primeira aproximação do mapeamento da cena 3D para a cena a 2D. Os erros do modelo da câmara diminuem com a qualidade da câmara e qualidade das lentes, o erro aumenta desde o centro da imagem até às margens desta, onde o efeito de barril<sup>3</sup> é mais acentuado nas margens [13]. Alguns destes erros podem ser compensados por algoritmos ou filtros.

Existe um plano  $\Omega$  fixo a uma distância  $f$  em frente a um plano de imagem  $\Pi$ . Um *pinhole* ideal é encontrado no plano  $\Omega$  - ver Figura 2.2.2.

Todos os raios de luz emitidos ou reflectidos por objectos passam através do *pinhole* e formam uma imagem invertida deste mesmo objecto na imagem do plano; ver Figura 2.2.1. Cada ponto do objecto constitui um ponto na imagem, formando-se uma linha por



cada ponto formado. A imagem seguinte mostra este fenómeno. A este fenómeno de projecção do espaço 3D para um plano dá-se o nome de projecção projectiva [15].

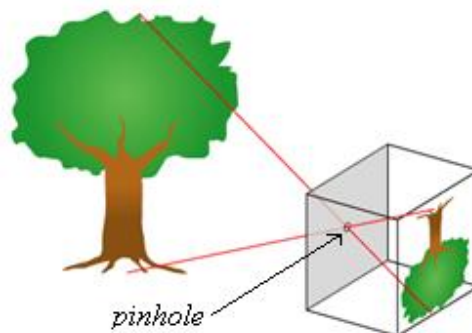


Figura 2.2.1 Representação típica do modelo *pinhole camera*

O modelo geométrico do modelo *pinhole camera* consiste num plano de imagem  $\Pi$  e um ponto  $C$  no plano  $\Omega$ . O ponto  $C$  é chamado de centro óptico ou ponto de focagem.

O plano  $\Omega$  paralelo a  $\Pi$  é chamado de plano principal. A distância entre o centro óptico e a imagem é chamado de distância focal e representado por  $f$ .

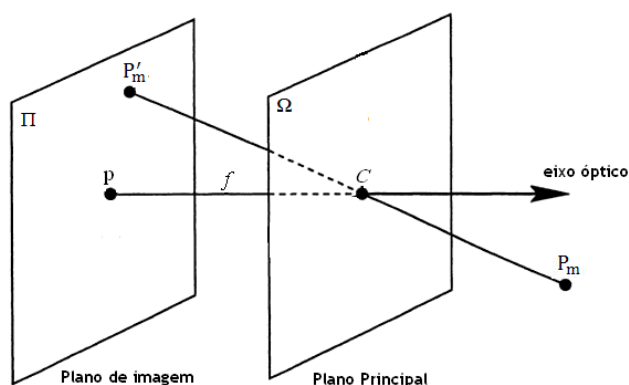


Figura 2.2.2: Modelo *pinhole camera*

A linha perpendicular à imagem do plano  $\Pi$  é chamado de eixo óptico e intercepta  $\Pi$  no ponto  $p$ , chamado ponto principal. Note-se que o plano focal é também perpendicular ao eixo óptico. Na Figura 2.2.2 temos representado o modelo *pinhole camera*.

Para a representação e explicação dos conceitos, vamos começar por definir um eixo de coordenadas no nosso modelo. O eixo  $Z$  aponta na direcção frontal da câmara sendo o eixo óptico ou eixo principal. O plano principal contém o referencial  $XYZ$  e é o lado da frente da câmara, a origem deste referencial é o ponto  $C$ . Estas definições podem ser observadas na Figura 2.2.3.

O plano de imagem onde toda a cena 3D é projectada através do diafragma da câmara é o plano de imagem e está paralelo aos eixos XY, está também a uma distância  $f$  da origem C. No modelo real da implementação do modelo o plano da imagem está localizada na coordenada  $-f$ .

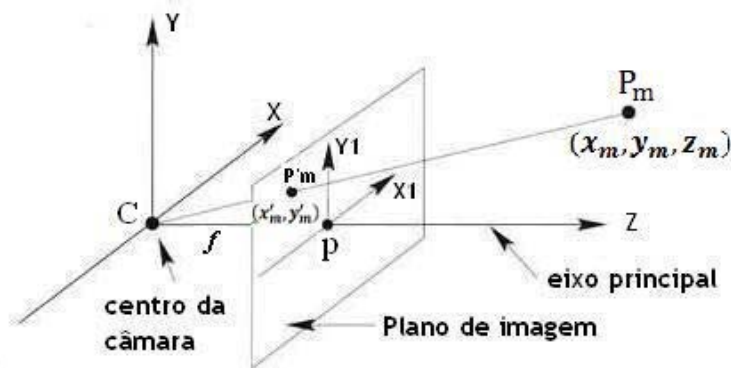


Figura 2.2.3: Representação dos eixos coordenados no nosso modelo *pinhole camera*.

O ponto  $P_m$  é um ponto no Mundo com as coordenadas  $(x_m, y_m, z_m)$  relativamente a XYZ.

A projecção do ponto  $P_m$  no plano de imagem é o ponto  $P'_m$ . Neste plano temos definido um referencial 2D  $X_1Y_1$  com origem p (centro da imagem) e nos quais as coordenadas do ponto  $P'_m$  deste sistema são  $(x'_m, y'_m)$ .

No modelo *pinhole camera* original a relação entre as coordenadas do ponto  $P_m$   $(x_m, y_m, z_m)$  e do ponto  $P'_m$   $(x'_m, y'_m)$  é dado por:

$$x'_m = -f \frac{x_m}{z_m} \quad (2.1)$$

$$y'_m = -f \frac{y_m}{z_m} \quad (2.2)$$

Esta relação geométrica de 3D para 2D descrita pelo *pinhole camera* apenas é uma projecção da imagem no plano de imagem seguido de uma rotação de  $180^\circ$ . Para evitar isto e simplificar a resolução, podemos colocar a imagem do plano na coordenada  $f$  em vez de  $-f$ . Assim obtemos:

$$x'_m = f \frac{x_m}{z_m} \quad (2.3)$$

$$y'_m = f \frac{y_m}{z_m} \quad (2.4)$$

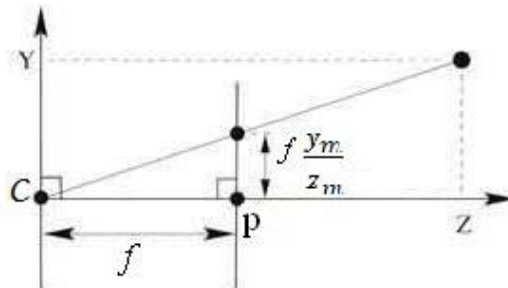


Figura 2.2.4 vista de lado do modelo *pinhole camera*

Salienta-se mais uma vez, que esta representação assume que a origem das coordenadas na imagem do plano é o ponto principal. Ter em conta que o passo anterior apenas é uma simplificação e uma maneira de tratar o problema real sem necessitarmos de fazer a rotação da imagem obtida.

A relação anterior entre o ponto  $P_m$  no mundo e o ponto  $P'_m$  no plano de imagem vem com a relação:

$$P'_m = \mathbf{P} P_m \quad (2.5)$$

Sendo  $\mathbf{P}$  a matriz de projecção da câmara. A matriz de projecção é uma matriz de dimensão 3x4 que contém os parâmetros intrínsecos e extrínsecos da câmara - distância focal, centro da imagem, factor de escala dos pixéis, rotação e translação da câmara. Em seguida explicamos onde se localiza cada um dos parâmetros na matriz de projecção.

O modelo *pinhole camera* assume que a origem das coordenadas na imagem do plano é feita no ponto principal,  $p$ . Tipicamente em processamento de imagem não usamos o ponto principal como origem do referencial neste plano, mas sim num dos cantos da imagem. Considerando que a origem do referencial é feita no canto inferior esquerdo do plano da imagem  $X_2Y_2$  e que as coordenadas do ponto principal neste novo referencial são  $(x_0, y_0)$ .

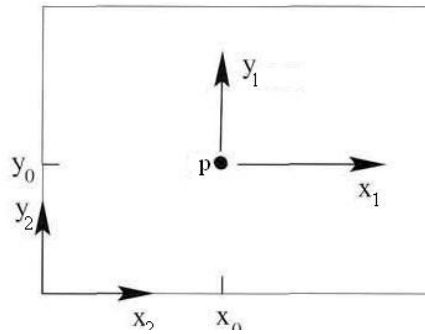


Figura 2.2.5: Eixo de coordenadas  $X_1Y_1$  no ponto principal e  $X_2Y_2$  num dos cantos da imagem.

As expressões que nos dão as coordenadas  $(x'_m, y'_m)$  no plano de imagem com origem no ponto principal:

$$x'_m = f \frac{x_m}{z_m} \quad (2.6)$$

$$y'_m = f \frac{y_m}{z_m} \quad (2.7)$$

Fazendo  $p_x = x_0$  e  $p_y = y_0$  as expressões (2.6) e (2.7), passam a:

$$x'_m = f \frac{x_m}{z_m} + p_x \quad (2.8)$$

$$y'_m = f \frac{y_m}{z_m} + p_y \quad (2.9)$$

E ficamos com a origem do nosso referencial num dos cantos da imagem.

Reescrevendo a expressão (2.5) que relaciona  $P'_m$  e  $P_m$  e apenas em função dos parâmetros intrínsecos da câmara, em coordenadas homogêneas fica-se com:

$$\widetilde{P}'_m = \mathbf{K} [1 \mid 0] \widetilde{P}_m \quad (2.10)$$

Em que o ponto  $\widetilde{P}_m$  é:

$$\widetilde{P}_m = \begin{bmatrix} P_m \\ 1 \end{bmatrix} \quad (2.11)$$

e o ponto  $\widetilde{P}'_m$  é a representação de  $P'_m$  em coordenadas homogêneas definido por:

$$\widetilde{P}'_m = \begin{bmatrix} P'_m \\ 1 \end{bmatrix} \quad (2.12)$$

E que  $\mathbf{K}$  é :

$$\mathbf{K} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.13)$$

$\mathbf{K}$  é definido como a matriz de calibração da câmara, contendo a distância focal e o centro da imagem. [13]

O modelo anterior considera que todo o espaço em frente câmara está representado segundo o referencial da câmara com origem em  $C$ , não sendo prático estar constantemente a representar o espaço envolvente no referencial da câmara. Para simplificar é adicionado mais um referencial tridimensional  $X_w Y_w Z_w$  do nosso mundo exterior (chamado referencial do mundo) e é nele onde o nosso espaço de trabalho ficará representado. As coordenadas 3D do nosso ponto  $P_m$  neste novo referencial do mundo serão  $(x_{mw}, y_{mw}, z_{mw})$  e definem o vector  $P_{mw}$ . O referencial da câmara e o referencial do mundo estão relacionados através de uma rotação e de uma translação. Na Figura 2.2.6 poderemos visualizar a relação entre os eixos coordenados. [14]

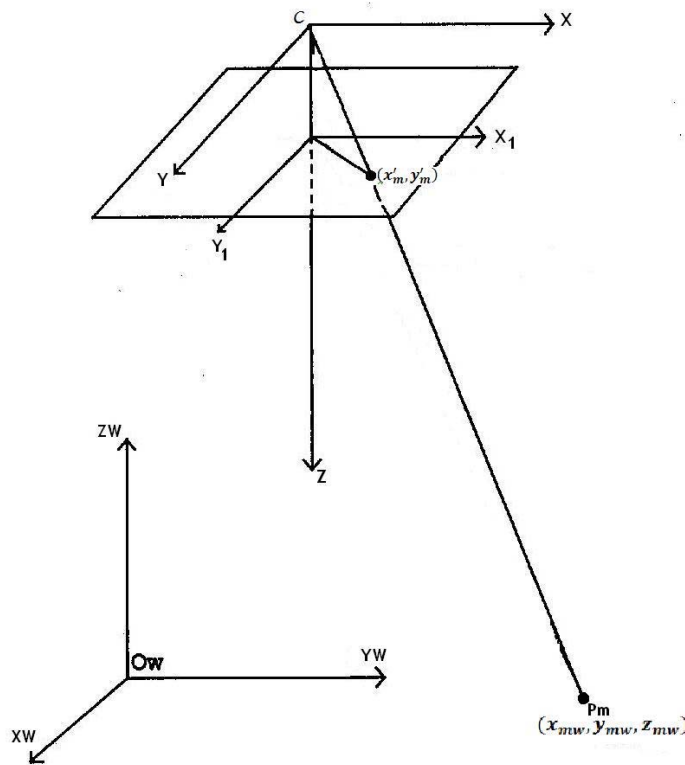


Figura 2.2.6 Referencial do mundo

A relação entre os pontos nos referenciais da câmara e do mundo é expressa por:

$$P_m = \mathbf{R}(P_{mw} - T) \quad (2.14)$$

e a relação do ponto no plano de imagem com o ponto no mundo em coordenadas homogêneas é expressa em:

$$\widetilde{P}'_m = \mathbf{KR} \begin{bmatrix} \mathbf{I} & | & -T \\ 0 & | & 1 \end{bmatrix} \widetilde{P}_{mw} \quad (2.15)$$

Na expressão (2.15)  $\widetilde{P}_{mw}$  é definido por:

$$\widetilde{P}_{mw} = \begin{bmatrix} P_{mw} \\ 1 \end{bmatrix} \quad (2.16)$$

A matriz  $\mathbf{R}$  é uma matriz 3x3 designada matriz de rotação sendo representada por:

$$\mathbf{R} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \quad (2.17)$$

Nesta matriz estão implícitos os 3 ângulos sobre cada um dos eixos do referencial, e que são denominados os ângulos de Euler, yaw  $\theta$ , pitch  $\varphi$ , e roll  $\psi$ . Estes podem ser obtidos da matriz de rotação por exemplo através das expressões:

$$\theta = \arctan2(r_9, \sqrt{1 - r_9^2}) \quad (2.18)$$

$$\varphi = \arctan2(r_3, r_6) \quad (2.19)$$

$$\psi = \arctan2(-r_7, r_8) \quad (2.20)$$

De salientar que estas expressões não são únicas, dependendo do sinal do ângulo que escolhermos para  $\theta$ . Mais pormenores podem ser consultados em [26].

O vector  $\mathbf{T}$  vector de translação representado por

$$\mathbf{T} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (2.21)$$

Podemos então definir a matriz de projecção  $\mathbf{P}$  da câmara:

$$\mathbf{P} = \mathbf{KR}[\mathbf{I} \quad | \quad -\mathbf{T}] \quad (2.22)$$

A matriz  $\mathbf{P}$  tem 9 graus de liberdade, 3 para a matriz de calibração  $\mathbf{K}$  (elementos  $f, p_x, p_y$ , 3 para  $\mathbf{R}$  (os ângulos  $\theta, \varphi$  e  $\psi$ ) e 3 para  $\mathbf{T}$ .

$\mathbf{K}$  contém os parâmetros internos ou intrínsecos da câmara e  $\mathbf{R}$  e  $\mathbf{T}$  relaciona a orientação e posição da câmara com o sistema de coordenadas do mundo. [13]

Reescrevendo a expressão (2.22) de forma a separar os parâmetros intrínsecos e extrínsecos da câmara fica-se com:

$$\widetilde{\mathbf{P}}'_m = \mathbf{K} \begin{bmatrix} \mathbf{R} & | & -\mathbf{RT} \\ 0 & | & 1 \end{bmatrix} \widetilde{\mathbf{P}}_{mw} \quad (2.23)$$

O modelo *pinhole camera* assume que os sistemas de eixos coordenados têm ambos as mesmas escalas. Nas câmaras actuais com o uso de CCD, a coordenada no plano de imagem costuma ser dado em pixéis. Com a possibilidade da não existência de pixéis completamente quadrados, temos um pequeno problema de factor de escala em cada direcção. Principalmente no número de pixéis por unidade de distância. [13]

Para continuarmos a usar o modelo *pinhole camera*, usando câmaras com CCD, apenas é necessário efectuar a conversão de pixéis para unidades de distância.

Então,  $\mathbf{K}$  virá definido como:

$$\mathbf{K} = \begin{bmatrix} a_x & 0 & x \\ 0 & a_y & y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.24)$$

Em que  $a_x = fm_x$  e  $a_y = fm_y$  representam a distância focal da câmara em termos das dimensões dos pixéis nas direcções dos eixos do referencial  $X_1Y_1$ . Analogamente  $x_0$  e  $y_0$ , é o ponto principal em termos de pixéis,  $x = m_x p_x$ ,  $y = m_y p_y$ . Com a adição destas duas novas variáveis  $m_x$  e  $m_y$  a matriz de projecção passa a ter 11 graus de liberdade.

É de nosso interesse através de uma imagem adquirida, conseguir retirar toda a informação referente ao ponto  $\mathbf{P}_{mw}$  no mundo; Para o efectuar necessitamos de parâmetros adicionais sobre o nosso sistema de visão que precisam de ser especificados e calibrados. [15]



Ao procedimento de determinação destes parâmetros dá-se o nome de calibração do sistema de visão. Alguns algoritmos serão analisados mais adiante.

A matriz de projecção  $\mathbf{P}$  pode ser definida:

$$\mathbf{P} = \mathbf{K}[\mathbf{R} \quad | \quad -\mathbf{RT}] = \mathbf{KR}[\mathbf{I} \quad | \quad -\mathbf{T}] \quad (2.25)$$

Podemos concluir que para recorrer à visão artificial é necessário conhecer à priori alguns parâmetros sobre o sistema. Em caso de não estarmos na posse destes parâmetros então é necessário determiná-los. No capítulo 3.5 serão abordados alguns algoritmos para obtenção dos parâmetros intrínsecos e extrínsecos das câmaras, calibração do sistema de visão.

Considerando as distorções introduzidas pelas lentes vamos introduzir um novo ponto de coordenadas  $(x_{md}, y_{md})$ , que são as coordenadas do ponto  $P'_m - (x'_m, y'_m)$ , no plano de imagem devido às distorções da lente. Mais tarde este efeito de distorção será abordado, ficando desde já definido.



# Capítulo 3

## Visão Estereoscópica

### 3.1 Introdução

Devido à evolução das tecnologias de áudio, quando ouvimos falar em *stéreo* é comum associarmos ao som. Colunas de som independentes fazem chegar aos nossos ouvidos sons ligeiramente diferentes. Estes sons recebidos são processados por nós no cérebro, dando a sensação de imersão ao ambiente de onde se originaram os sons. Sendo banal hoje em dia, em qualquer *home cinema* comprado para casa. Através deste sistema consegue-se transmitir ao espectador estar presente no meio da acção que se desenrola no plano de projecção do filme.

Esta tecnologia de som *stéreo* pode ser estendida ao visual [Lipton 1982]. Os animais, em específicos os seres humanos, são dotados de dois olhos que visualizam o mundo de diferentes pontos de vista, pois estão separados por uma distância. Podemos comprovar isto através de uma pequena experiência, alinhar o polegar de uma das mãos com um qualquer objecto (no caso da figura em baixo com uma bandeira) e o seu nariz. Se focar a sua visão para o dedo, com os dois olhos abertos, verá o objecto a duplicar, um para cada olho (feche um olho e abra o outro e em seguida inverta). [StereoGraphics 1997]. Ao focarmos, a bandeira e se mantiver o polegar na linha de visão, irá ter a visão de dois polegares. A Figura 3.1.1 e Figura 3.1.2 exemplificam os resultados das experiências.



Figura 3.1.1 Focando o polegar, obtemos a visão de duas bandeiras.  
Retirado de [15].

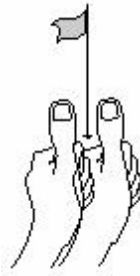


Figura 3.1.2 Focando a Bandeira obtêm-se a visão de dois polegares. Retirado de [15].

Estas diferenças entre imagens geradas pelo olho direito e pelo olho esquerdo são processadas pelo cérebro dando-nos uma noção e percepção de profundidade<sup>2</sup>.

Em computação gráfica, normalmente são geradas imagens mono (isto é, a partir de uma única câmara virtual) e estas imagens são visualizadas por um monitor ou projectadas em um plano. A estereoscopia visual, ao contrário, é feita a partir da geração de duas imagens a partir das localizações das câmaras virtuais separadas de uma determinada distância. [15]

## 3.2 Algumas noções

Numa foto tirada por uma simples câmara existem alguns efeitos que nos permitem retirar algumas características/informações tridimensionais da cena apresentada na imagem. A estes efeitos dá-se o nome de efeitos passivos. [15]

- **Perspectiva:** que certos objectos possam parecer “maiores” ou “menores”, ou estarem mais ao “fundo” ou mais à frente, temos que ter ideia que por mais que essa imagem possa parecer 3D ela é uma imagem planar.

Na perspectiva, objectos de mesmo tamanho devem aparecer maiores na foto quando mais estiverem próximos da câmara [15]. Olhando para Figura 3.2.1 parece-nos que a linha mais ao centro é maior, mas na realidade as duas têm o mesmo tamanho mas pelo efeito de perspectiva gerado a partir de outras linhas (definindo um ponto de fuga), concluímos que o traço mais ao centro da imagem é maior que o traço mais afastado do centro da imagem.

---

<sup>2</sup> Consultar Anexo A

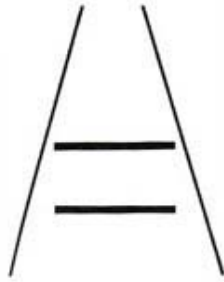


Figura 3.2.1 Efeito de Perspectiva

- Oclusão: a oclusão é responsável em esconder objectos em partes ou por inteiro. Assim conseguimos ter uma noção de distância dos objectos à câmara. Na figura em baixo podemos ver que na imagem da esquerda a esfera no mundo está “atrás” do cubo e na imagem da direita está à “frente” do cubo.

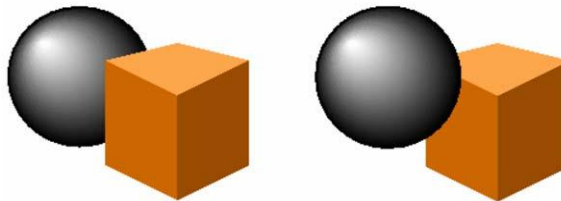


Figura 3.2.2: Efeito de Oclusão. Retirado de [15]

- Sombra: Numa dada imagem resultante de uma projecção 3D para 2D é-nos difícil ver se um dado objecto está colado a um plano de apoio. Isto porque o inverso dessa projecção, dado um ponto na imagem, é uma recta no espaço 3D, isto é, diversos pontos na cena 3D alinhados com a câmara projectam-se em um mesmo ponto. [15]

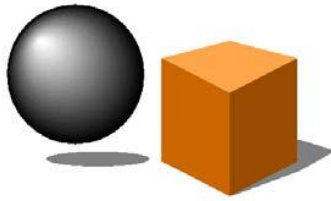


Figura 3.2.3 Efeito de Sombra. Retirado de [15].

A sombra auxilia a visualizar se um objecto está colado no plano de apoio, mas em algumas situações através dela não conseguimos ter 100% de certeza. Por exemplo, na Figura 3.2.3 podemos observar que a esfera está a uma certa distância do plano, pois conseguimos visualizar a toda a sombra por baixo da esfera gerada por ela, no entanto no caso do cubo temos a sensação de ele estar apoiado no plano de apoio, mas através de uma imagem projectada não nos é possível confirmar tal facto. [15]

Mais alguns efeitos passivos podem ser encontrados, gradiente de textura, efeito de iluminação, etc.

No sistema de visão humano uma consequência imediata da diferença das imagens capturadas pelos dois olhos é o espaçamento entre o mesmo ponto projectado nas duas retinas, chamado de *disparidade binocular*<sup>3</sup>.

Num sistema de visão binocular artificial, o qual usa duas fontes de captura de imagem para aquisição tridimensional do espaço envolvente, as suas relações geométricas são baseados no modelo *pinhole camera* e na geometria epipolar descrita na secção 3.4.

### 3.3 Modelo *pinhole camera* e o sistema estereoscópico

A partir do modelo descrito no capítulo anterior conseguimos perceber como efectuar o mapeamento 3D para 2D. Se juntarmos ao nosso sistema mais uma câmara localizada de uma posição diferente que a primeira, mas capturando no mesmo instante o mesmo cenário que a primeira, conseguimos através de triangulação obter informação 3D através de duas imagens 2D. Este conceito do uso de duas fontes de visão para captura de informação envolvente é denominado de *visão estereoscópica*. É um sistema análogo à visão humana, tendo dois pontos de vista da mesma cena através dos olhos e ter a percepção de profundidade.

Para certas tarefas industriais é muito importante ter a noção tridimensional do cenário envolvente o que faz, o uso de um sistema estereoscópico um sistema mais robusto na recolha da informação pretendida. A Figura 3.3.1 retrata um sistema de visão estereoscópica.

---

<sup>3</sup> Consultar Anexo A

<sup>6</sup>consultar Anexo A

Podemos facilmente concluir que o uso de um sistema destes é vantajoso e interessante.

Chama-se a atenção que, todos os conceitos explicados no capítulo anterior são usados no sistema de visão estereoscópica. Começamos por adicionar mais uma câmara ao nosso sistema e fazer o mesmo raciocínio efectuado.

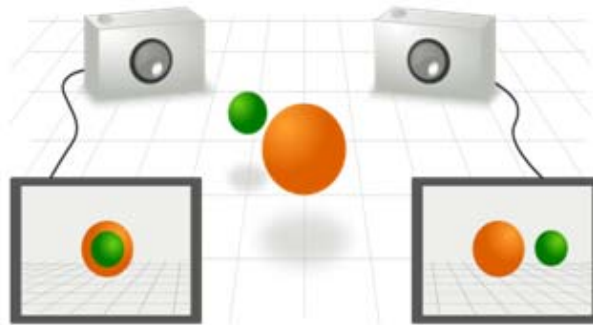


Figura 3.3.1 Sistema de visão estereoscópica

Tendo a câmara 1 (que daqui em diante é definida como a câmara que se encontra do lado esquerdo no caso de nos posicionarmos por trás do sistema de visão estereoscópico) que contém um referencial  $X_1Y_1Z_1$  com origem em  $C_1$ , e que contém um eixo óptico perpendicular ao seu plano de imagem com a referência  $X_1Y_1$ . Definindo para uma segunda câmara, um referencial  $X_2Y_2Z_2$  centrado em  $C_2$ , um centro óptico perpendicular ao seu plano de imagem com um referencial de coordenadas  $X_2Y_2$ .

As relações calculadas no capítulo anterior,

$$x'_m = f \frac{x_m}{z_m} \quad (3.1)$$

$$y'_m = f \frac{y_m}{z_m} \quad (3.2)$$

Continuam a ser válidas e temos para a segunda câmara:

$$x'_{2m} = f_2 \frac{x_m}{z_m} \quad (3.3)$$

$$y'_{2m} = f_2 \frac{y_m}{z_m} \quad (3.4)$$

Salienta-se que daqui em diante todas as variáveis com o índice 1 ou 2 associados referem-se ao sistema da primeira ou da segunda câmara respectivamente. O referencial 3D  $(x_w, y_w, z_w)$  e o nosso ponto  $P_{mw}$  com coordenadas  $(x_{mw}, y_{mw}, z_{mw})$  no mundo, são as únicas coisas que se mantêm constantes na passagem do sistema com uma câmara para o sistema estereoscópico.

Existe também uma matriz de rotação e de translação, devido ao facto da segunda câmara ter uma rotação e uma translação diferente relativamente ao referencial do mundo com origem em  $O_w$ .

$$\begin{bmatrix} x_{2m} \\ y_{2m} \\ z_{2m} \end{bmatrix} = \mathbf{R}_2 \begin{bmatrix} x_{mw} \\ y_{mw} \\ z_{mw} \end{bmatrix} + \mathbf{T}_2 \quad (3.5)$$

$$\mathbf{P}_2 = \mathbf{K}_2 [\mathbf{R}_2 \quad | \quad -\mathbf{R}_2 \mathbf{T}_2] \quad (3.6)$$

Em que  $\mathbf{P}_2$ ,  $\mathbf{K}_2$ ,  $\mathbf{R}_2$  e  $\mathbf{T}_2$  são as matrizes de projecção, calibração e rotação respectivamente e o vector de translação da câmara número 2.

No capítulo 3.5 são apresentados dois algoritmos para o cálculo dos parâmetros intrínsecos e extrínsecos de cada uma das câmaras pertencentes ao sistema estereoscópico.

### 3.4 Geometria epipolar

Geometria epipolar é a geometria projectiva intrínseca que existe entre duas vistas. Ela é independente da estrutura da cena e apenas é dependente dos parâmetros internos das câmaras e da posição relativa destas. Esta geometria intrínseca está encapsulada numa matriz, denominada matriz  $\mathbf{F}$ . A matriz  $\mathbf{F}$  é uma matriz de dimensão 3x3 e se um ponto  $P_m$  no espaço 3D é visto na primeira vista do sistema estereoscópico, sendo a sua projecção no plano de imagem  $P'_{1m}$ , na segunda vista  $P_m$  é visto como  $P'_{2m}$  então estes pontos nos dois planos de imagem satisfazem a relação [13]:



$$P_{2m}'^T \mathbf{F} P_{1m}' = 0 \quad (3.7)$$

A Figura 3.4.1 representa o modelo de um sistema estereoscópico e a sua terminologia segue descrita em seguida.

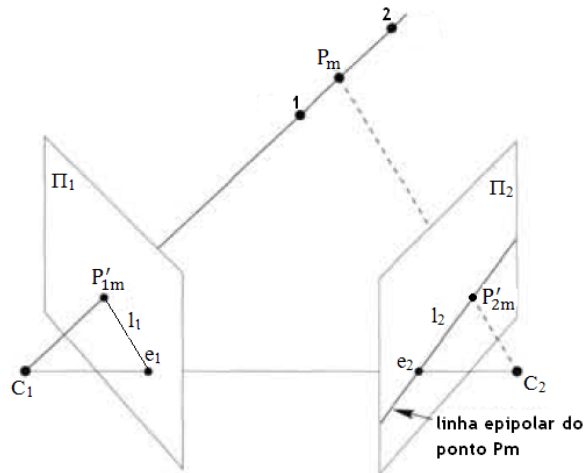


Figura 3.4.1 Modelo de um sistema estereoscópico

Os pontos  $e_1$  e  $e_2$  são denominados epipolos das imagens 1 e 2 respectivamente. Estes dois pontos são formados pela linha que une os dois centros das câmaras  $C_1$ ,  $C_2$  e esta denomina-se a *baseline* do sistema.

Na Figura 3.4.1 está também representado o plano epipolar, que é definido pelas linhas de projecção do ponto  $P_m$  no mundo ao centro de cada câmara juntamente com a *baseline*. A intercepção do plano epipolar com as imagens do plano gera em cada um destes uma linha, denominada de linha epipolar.

Através destes conceitos observa-se que, se o ponto  $P_m$  mover sobre o raio de projecção definido para a câmara 1 (equivalente a mover-se para a posição 1 ou 2 na Figura 3.4.1), então a projecção deste ponto na imagem do plano da câmara 2, move-se sobre a linha epipolar  $l_2$ . Este conceito é muito importante para a realização de *match* de pontos de duas imagens, conhecendo um ponto  $P'_{1m}$  na primeira vista, na segunda vista não necessitamos de procurar em toda a imagem pelo correspondente par de  $P'_{1m}$ , mas sim apenas sobre a linha epipolar  $l_2$ . Esta definição é válida inversamente, ou seja, conhecido um ponto  $P'_{2m}$  na segunda vista, o seu par na primeira vista está restrito à linha epipolar  $l_1$ .

Se já tivermos acesso a  $\mathbf{F}$  e conhecendo um ponto  $P_m$  no plano de imagem da câmara 1, então o seu par encontra-se restringido à linha epipolar  $l_2$ . Em que  $l_2$  é obtido a partir de:

$$l_2 = \mathbf{F}P'_{1m} \quad (3.8)$$

Se pretendermos o inverso apenas temos que transpor a matriz  $\mathbf{F}$ , representado em (3.9)

$$l_1 = \mathbf{F}^T P'_{2m} \quad (3.9)$$

Se não conhecemos a matriz  $\mathbf{F}$ , é necessário determiná-la, existindo alguns caminhos para a sua determinação, como por exemplo:

- Processos lineares com e sem escalamento;
- Processos não lineares;
- Através do conceito de paralaxe virtual.

Em [16] são explicadas algumas vantagens e desvantagens entre cada um deles, entre as quais destacam-se que, os critérios não lineares serem melhores que os critérios lineares, mas ao mesmo tempo mais pesados computacionalmente.

Vamos abordar dois critérios, para o cálculo da matriz  $\mathbf{F}$ , a partir dos parâmetros intrínsecos das câmaras (obtidos pelo algoritmo de Tsai, ver secção 3.5) e através do *eight-point algorithm* – ver secção 3.5.2.

Rimon Elias e Robert Laganière [16] dizem que no caso de os parâmetros intrínsecos de ambas as câmaras já serem conhecidos então a matriz fundamental é a matriz essencial do sistema. Esta matriz é uma precursora da matriz fundamental e depende de 5 parâmetros descrevendo a disparidade binocular entre as duas vistas, em que os 3 primeiros parâmetros são devido à rotação 3D e os outros 2 definem a direcção da translação – apenas descreve os parâmetros extrínsecos da câmara; Enquanto a matriz  $\mathbf{F}$  contém os parâmetros intrínsecos e extrínsecos.

Se temos  $P'_{1m}$  e  $P'_{2m}$  de um ponto  $P_m$  nas duas imagens então segundo o modelo *pinhole*, tendo a matriz de rotação  $\mathbf{R}$  e o vector deslocamento  $T$  da segunda vista em relação à primeira podemos então determinar  $\mathbf{F}$  através da matriz  $\mathbf{E}$  - matriz essencial.  $\mathbf{R}$  e  $T$  passam a ser chamados parâmetros extrínsecos do sistema estereoscópico.

$$\mathbf{F} = \mathbf{K}_2^{-T} T_x \mathbf{R} \mathbf{K}_1^{-1} \quad (3.10)$$

Onde  $T_x$  é a matriz anti-simétrica definida pelo vector de translação  $T$ ,  $\mathbf{K}_2$  é a matriz de calibração da câmara 2,  $\mathbf{R}$  a matriz de rotação do sistema estereoscópico e  $\mathbf{K}_1$  a matriz de calibração da câmara 1.

A matriz essencial  $\mathbf{E}$  pode ser escrita como:

$$\mathbf{E} = \mathbf{T}_x \mathbf{R} \quad (3.11)$$

Com a relação (3.11) a expressão (3.7) passa a:

$$\mathbf{P}_{2m}'^T \mathbf{E} \mathbf{P}_{1m} = 0 \quad (3.12)$$

Ficando as matrizes  $\mathbf{F}$  e  $\mathbf{E}$  relacionadas por:

$$\mathbf{F} = \mathbf{K}_2^{-T} \mathbf{E} \mathbf{K}_1^{-1} \quad (3.13)$$

$$\mathbf{E} = \mathbf{K}_2^T \mathbf{F} \mathbf{K}_1 \quad (3.14)$$

## 3.5 Calibração do sistema de visão

### 3.5.1 O algoritmo de Tsai: versatile camera calibration technique

A calibração de um sistema de visão tenta calcular um largo número de parâmetros do sistema de visão. Classicamente são usados processos não lineares para percorrer todo o cenário de calibração. Para evitar estes processos não lineares, muitas vezes usam-se estratégias de sistemas de equações lineares, usadas por exemplo no modelo DLT – *Direct linear transformation* – em que é possível que se obtenha um número de variáveis superior ao número de graus de liberdade. O problema deste algoritmo é que ignora a distorção das lentes assim como a dependência entre parâmetros. [14]

O algoritmo de Tsai usa uma particularidade física do sistema de visão, esta particularidade trata-se do alinhamento radial. Esta particularidade é definida em função da rotação relativa e translação (excepto na componente  $Z$ ) entre a câmara e os pontos de calibração - ver Figura 3.5.1 e particularidades descritas a seguir:

- É assumido que a distorção é radial, ou seja, em nada afecta na profundidade do espaço envolvente. Não importa qual a magnitude da distorção, a direcção do vector  $O_i P_d$  (vector definido pela origem  $O_i$  e pelo ponto  $P_d$  com as coordenadas  $(x_d, y_d)$  na imagem do plano) mantém-se constante e radialmente alinhado com o vector  $P_{Oz} P$ , ( formado pela origem  $P_{Oz}$  - dada

pela coordenada  $z$  do ponto  $P$  - e pelo ponto  $P$ ). A coordenada  $z$  destes dois pontos é a mesma - ver Figura 3.5.1.

- A distância focal  $f$  não influencia a direção do vector  $O_iP_d$ .
- O ponto no mundo é transladado e rodado em  $X, Y$  logo  $O_iP_d$  será sempre paralelo a  $P_{Oz}P$  para todos os pontos. A componente  $Z$  de translação não altera a direção de  $O_iP_d$ . [14]

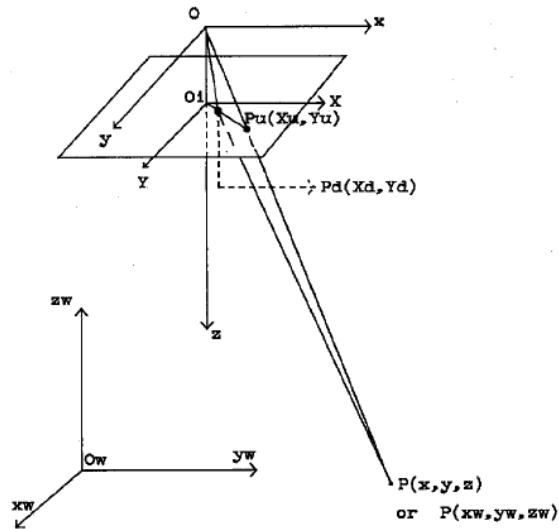


Figura 3.5.1 Particularidades assumidas no algoritmo de Tsai.  
Retirada de [14]

Para a calibração foi usado um conjunto de pontos coplanares com a câmara a calibrar – ver Figura 3.5.2.

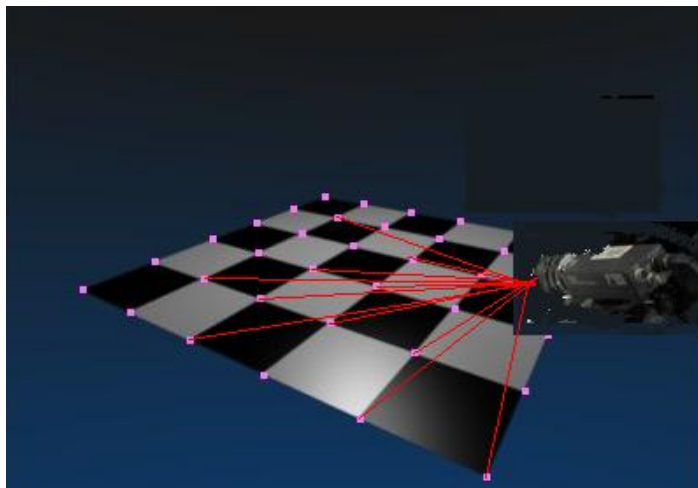


Figura 3.5.2 Setup Calibração Tsai

Esta calibração é feita através de um conjunto de pontos em que as suas posições no mundo  $(x_w, y_w, z_w)$  são conhecidas – bolinhas rosa na Figura 3.5.2. - e as coordenadas

$(x'_m, y'_m)$  medidas no plano de imagem – pontos formados pela intercepção das linhas vermelhas da Figura 3.5.2, com o plano de imagem da câmara a calibrar.

O conjunto de pontos no mundo e que a sua posição é conhecida (pontos de calibração) estão assentes num plano (chamado plano de calibração) – plano axadrezado na Figura 3.5.2.

Depois de calculadas as coordenadas no plano de imagem dos pontos do mundo conhecidos, todo o algoritmo se desenrola usando as relações anteriormente descritas. Através delas é possível calcular os parâmetros intrínsecos e extrínsecos apresentados a seguir.

Os parâmetros extrínsecos que pretendemos calcular são a transformação do sistema de coordenadas 3D do mundo para o sistema de coordenadas 3D da câmara centrado no centro óptico. Existem 6 parâmetros extrínsecos, 3 deles são os ângulos de Euler, yaw  $\theta$ , pitch  $\varphi$ , e roll  $\psi$  para a rotação e os três componentes de translação do vector T.

A matriz de rotação pode ser expressa como:

$$\mathbf{R} = \begin{bmatrix} \cos \psi \cos \theta & \sin \psi \cos \theta & -\sin \theta \\ -\sin \psi \cos \varphi + \cos \psi \sin \theta \cos \varphi & \cos \psi \cos \varphi + \sin \psi \sin \theta \sin \varphi & \cos \theta \sin \varphi \\ \sin \psi \sin \varphi + \cos \psi \sin \theta \cos \varphi & -\cos \psi \sin \varphi + \sin \psi \sin \theta \cos \varphi & \cos \theta \cos \varphi \end{bmatrix} \quad (3.15)$$

E que pode ser escrita numa forma mais compacta:

$$\mathbf{R} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \quad (3.16)$$

O vector T de translação é definido por:

$$\mathbf{T} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (3.17)$$

Através deste algoritmo é possível também calcular os parâmetros intrínsecos já referidos anteriormente no capítulo 2 e que se passam a citar:

- $f$  distância focal efectiva.
- Coeficiente de distorção da lente.
- $(C_x, C_y)$  Cálculo da posição da origem na imagem do plano.

Foi implementado o algoritmo descrito em [14] e pode ser consultado para mais pormenor. Nele é descrito o processo para o cálculo do verdadeiro centro da imagem  $(C_x, C_y)$  neste trabalho não foi efectuado este procedimento e foi assumido que o centro da imagem é o centro geométrico da imagem. No nosso caso as imagens adquiridas têm as dimensões (640 x 480) o centro da imagem será (320 x 240).

No documento original é apresentado um plano de calibração em que consiste numa sequência de quadrados com o mesmo tamanho e todos espaçados da mesma distância. No processamento de imagem são retiradas as coordenadas dos 4 cantos de cada quadrado e introduzidos no algoritmo. Para testar o algoritmo de detecção de clusters usado na detecção dos marcadores da ferramenta e para contornar a imprecisão do cálculo dos cantos dos quadrados, foi criado um plano de calibração diferente do referenciado no documento [14]. Ao invés do uso de quadrados como pontos de calibração foram usados círculos. O plano de calibração criado está na Figura 3.5.3

As coordenadas para o ponto  $(x_i, y_i)$  que precisamos para o processo de calibração, são as coordenadas do centro de cada círculo detectado.

Foram criados 2 planos de calibração mas com folhas de diferentes tamanhos - tamanhos A4, A3 e A2. Um dos planos continha 88 e outro com 368 pontos de calibração.

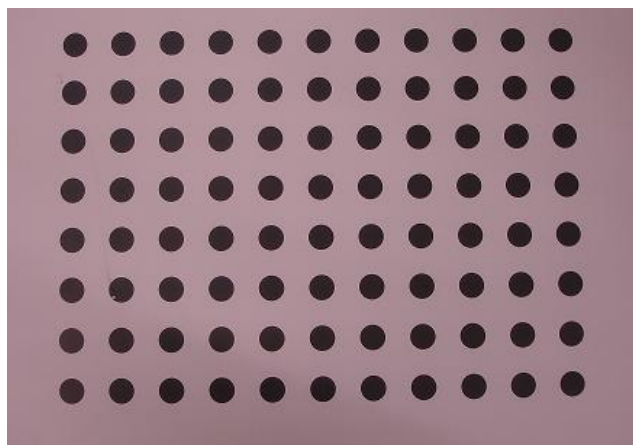


Figura 3.5.3 Plano de calibração de 88 pontos.

Este algoritmo serve para a calibração das câmaras individualmente, no caso de um sistema estereoscópico para o cálculo da matriz  $\mathbf{F}$ , apenas temos que calcular a rotação da segunda câmara em relação à primeira e usar as expressões (3.10) a (3.14) para o cálculo de  $\mathbf{F}$ .

Conhecidas as rotações de cada câmara ao plano de calibração, para obter a rotação e translação do sistema, ou seja a rotação e translação da câmara 2 em relação à câmara 1, colocando esta última como origem do sistema:

$$\mathbf{R} = \mathbf{R}_2 \mathbf{R}_1^T \quad (3.18)$$

Para obter a translação da câmara 2 em relação à câmara 1:

$$\mathbf{T} = \mathbf{T}_1 - \mathbf{R}^T \mathbf{T}_2 \quad (3.19)$$

São obtidos assim os parâmetros extrínsecos do sistema estereoscópico que são necessários para o cálculo da matriz  $\mathbf{F}$ .

### 3.5.2 Eight-point algorithm

O algoritmo de 8 pontos é um algoritmo linear para a estimação da matriz essencial ou da matriz fundamental de um sistema estereoscópico dependente de um factor de escala. Foi introduzido por Christopher Longuet-Higgins em 1981 para o cálculo da matriz essencial. Richard I. Hartley (1997) [17] descreve como este mesmo algoritmo pode ser usado para o cálculo da matriz fundamental, do seu trabalho [17] resultou o algoritmo de oito pontos normalizado. O algoritmo aqui descrito está rigorosamente com a mesma notação que no artigo original [16], para não haver confusões.

Neste algoritmo também é necessária a identificação entre pares de pontos nos dois planos de imagem. Usando-se a relação entre os dois planos e a matriz fundamental já descrita atrás:

$$\mathbf{P}'_m{}^T \mathbf{F} \mathbf{P}'_{1m} = 0 \quad (3.20)$$

Determina-se  $\mathbf{F}$ , em que o resultado obtido depende de um factor de escala, do resultado verdadeiro para um determinado setup do sistema estereoscópico.

Sabemos que  $\mathbf{F}$  é uma matriz de dimensão 3x3, podemos escrever os nove elementos da matriz  $\mathbf{F}$  num vector em que vamos chamar de vector  $\mathbf{F}$ :

$$\mathbf{F} = [f_{11} \ f_{12} \ f_{13} \ f_{21} \ f_{22} \ f_{23} \ f_{31} \ f_{32} \ f_{33}]^T \quad (3.21)$$

a expressão (3.7) pode ser reescrita como:

$$[x'_{2m}x'_{1m} \ y'_{2m}x'_{1m} \ x'_{1m} \ x'_{2m}y'_{1m} \ y'_{2m}y'_{1m} \ y'_{1m} \ x'_{2m} \ y'_{2m} \ 1]\mathbf{F} = 0 \quad (3.22)$$

Onde  $P'_{2m} = [x'_{2m} \ y'_{2m} \ 1]^T$  e  $P'_{1m} = [x'_{1m} \ y'_{1m} \ 1]^T$

Se tivermos mais que 8 correspondências obtemos um sistema sobredimensionado e é viável o uso do método dos mínimos quadrados para resolver:

$$\min_{\mathbf{F}} = \sum_i (P'_{2mi} \mathbf{F} P'_{1mi})^2 \quad (3.23)$$

Considerando n pares de correspondências, então pode ser escrito como:

$$\min_f = \|\mathbf{NF}\|^2 \quad (3.24)$$

Onde  $\mathbf{N}$  é uma matriz de  $n \times 9$  elementos.

A equação (3.7) pode ser escrita como

$$\mathbf{NF} = \mathbf{0} \quad (3.25)$$

e refere que a solução pode ser encontrada usando a decomposição em valores singulares – *Singular value decomposition (SVD)* - de  $\mathbf{N}^T \mathbf{N}$  que corresponde ao vector próprio mais pequeno da matriz  $\mathbf{U}$ , que corresponde à última coluna de  $\mathbf{U}$ .

Através deste algoritmo, o processo de calibração é rápido e fácil. Para gerar os pares de pontos necessários, movia-se aleatoriamente um marcador no campo de visão da câmara. Durante esta amostragem o sistema retira as posições dos pontos e faz a sua identificação como pares, após esta amostra os pares obtidos são usados no algoritmo para o cálculo de  $\mathbf{F}$ . Este algoritmo sofre do problema de reconstrução métrica. Como mencionado anteriormente o cálculo de  $\mathbf{F}$  é obtido e depende de um factor de escala, sendo necessário “afinar” os valores de  $\mathbf{F}$  para se obter uma reconstrução 3D correcta.

### 3.6 Reconstrução 3D

Depois de encontrados os pares de pontos nas duas imagens, é de nosso interesse efectuar a reconstrução tridimensional do ponto.

Para esta tarefa mais uma vez existem diferentes caminhos que se podem tomar para efectuar a reconstrução tridimensional a partir das matrizes de projecção. Na lista seguinte são apresentadas algumas delas: minimização da distância, intercepção dos raios das câmaras, triangulação através dos raios das câmaras que podem não se interceptar e usando decomposição de valores singulares.

Idealmente através de triangulação dos raios das câmaras, era possível obter a posição tridimensional do ponto  $P_m$  (ver Figura 3.6.1), mas devido a ruído raramente ocorre a intercepção das rectas de projecção.



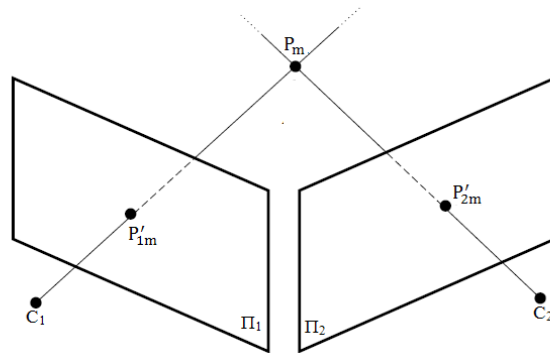


Figura 3.6.1 Triangulação ideal

No trabalho foi implementado o critério de triangulação que prevê que as rectas de projecção podem não se interceptar na realidade, retirando o ponto médio da perpendicular entre os dois raios, ver Figura 3.6.2.

A triangulação é a intercepção das rectas de projecção que saem dos centros das câmaras passando nos respectivos planos de imagem e no ponto comum a eles, ponto no mundo, as rectas intersectam-se. [18]

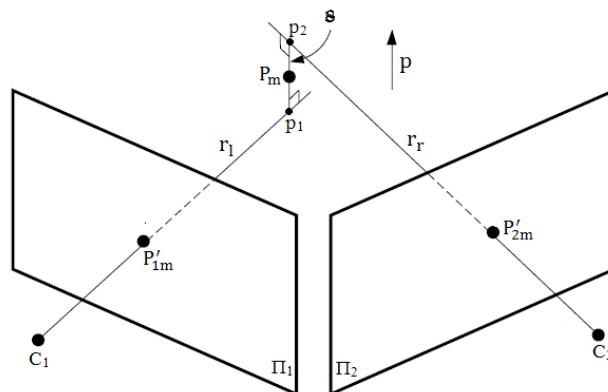


Figura 3.6.2 Triangulação sem intercepção dos raios

Rimon Elias and Robert Laganière [16] afirmam que a equação da linha de projecção da primeira vista pode ser representada por:

$$r_1 = aP'_{1m} \quad (3.26)$$

Onde  $a$  é um valor escalar. A linha de projecção da segunda vista, lado direito, referido ao plano de imagem esquerdo pode ser expresso como:

$$r_r = b\mathbf{R}^T P'_{2m} + T \quad (3.27)$$

Onde  $b$  é um valor escalar.

De salientar que conhecidos  $a$  e  $b$  as expressões anteriores no caso de triangulação ideal, dariam o mesmo ponto. Como estamos a considerar que a triangulação não é ideal então obtemos dois pontos  $p_1$  e  $p_2$  - ver Figura 3.6.2.

O vector perpendicular a  $r_l$  a  $r_r$  é:

$$p = P'_{1m} \times \mathbf{R}^T P'_{2m} \quad (3.28)$$

A linha  $s$  paralela a  $p$  e passando em  $aP'_{1m}$  é:

$$s = aP'_{1m} + cp \quad (3.29)$$

Onde  $c$  é um escalar, então a linha da esquerda mais o valor da linha perpendicular a  $r_l$  obtemos o outro extremo da linha do lado direito. Logo os pontos finais do segmento de recta  $s$  podem ser obtidos resolvendo:

$$aP'_{1m} + c(P'_{1m} \times \mathbf{R}^T P'_{2m}) = b\mathbf{R}^T P'_{2m} + T \quad (3.30)$$

Obtemos assim o ponto médio do erro entre os raios projectados. Uma boa medida do desvio que os raios estão a sofrer entre um e outro é analisar o escalar  $c$  - quanto mais próximo de zero for menos erro entre as rectas existe.

Voltando ao caso em que  $a$ ,  $b$  e  $c$  são conhecidos, na representação passamos a ter os pontos  $p_1$ ,  $p_2$  já referido anteriormente. Como estamos a considerar que o ponto  $P_m$  está a metade da distância entre  $p_1$  e  $p_2$  esta distância é  $c/2$ .

# Capítulo 4

## Hardware

Neste capítulo é descrito todo o hardware usado para a elaboração do trabalho apenas serão apresentadas as características mais relevantes. Para obter mais informações podem ser adquiridas através das folhas de características de cada componente.

### 4.1 Sistema estereoscópico

Sistema de visão, composto por duas câmaras com lentes iguais. As lentes usadas são de distância focal de 6 mm, e as câmaras fazem capturas de imagens com resolução de (640 x 480) pixéis e em modo automático a frequência máxima de captura está entre os 30-40ms. A comunicação destas câmaras é efectuada via USB e permitem o que o disparo do obturador funcione em modo manual ou em modo automático. Este funcionamento é configurado por software e em modo de funcionamento manual é necessário fornecer o sinal de disparo do obturador na entrada de trigger que possui. Esta característica das câmaras é importante para o nosso trabalho - é imperativo que as imagens capturadas pelo nosso sistema estereoscópico sejam do mesmo instante. Assim sendo apenas se usa o modo manual de disparo e à frequência de 10 Hz. As Figura 4.1.1 e Figura 4.1.2 mostram uma câmara do sistema estereoscópico.



Figura 4.1.1 Câmara do sistema estereoscópico



Figura 4.1.2 Entradas USB e *Trigger* das câmaras usadas

As câmaras foram assentes num tripé e colocadas numa barra de suporte, separadas de uma distância de 78 cm. A Figura 4.1.3 ilustra a disposição das câmaras, ficando o sistema estereoscópico assim definido.



Figura 4.1.3 Sistema estereoscópico montado

## 4.2 Marcadores Luminosos

Para obter a posição e rotação da ferramenta nela será acoplada um conjunto de leds de iluminação de alta potência, que daqui em diante serão referidos como marcadores ou marcadores luminosos.

Para os testes foi criado um suporte modular em forma de cruz (Figura 4.2.1) que permitisse facilmente, quando pretendido, alterar a posição dos marcadores.

Para a detecção da posição da ferramenta apenas é necessário um marcador, mas para retirar a rotação e orientação dela, é necessário ter a ideia do plano em que ela está “assente”. Para se conseguir definir este plano são precisos no mínimo 3 marcadores, tendo-se optado o uso de quatro marcadores para aumentar a precisão e robustez do sistema. Desta forma, se mais tarde se concluir que é vantajoso introduzir mais marcadores, as funções implementadas já estão preparadas para processar um maior número de marcadores.



Figura 4.2.1 Suporte construído para colocar os marcadores

Para se detectar o sincronismo entre a aquisição de imagens por parte do sistema estereoscópico, os nossos marcadores não fornecem uma emissão contínua de luz, estando apenas a emitir num curto espaço de tempo para que possa ser capturado pelo sistema estereoscópico. Desta forma é possível fornecer uma potência muito mais elevada durante um curto espaço de tempo, o que facilita a detecção destes nas imagens. Permite também diminuir/anular o incómodo provocado pela intensidade de luz nos utilizadores do sistema.

O disparo dos *leds* está sincronizado com o sistema de trigger das câmaras, se o nosso sistema estiver bem sincronizado, os marcadores luminosos apareceram em emissão contínua no filme capturado pelas imagens; Caso contrário, conseguimos visualizar um “piscar” de marcadores como na realidade se verifica a olho nu.

A Figura 4.2.2 mostra a disposição dos marcadores na ferramenta real de trabalho - notar que os marcadores não têm que ter a disposição em cruz, apresentada no modelo montado em cima.

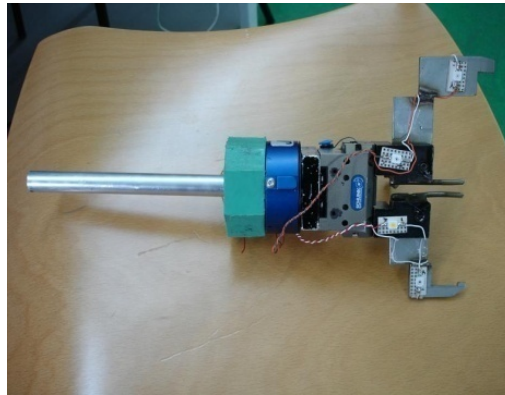


Figura 4.2.2 Disposição dos marcadores luminosos na ferramenta

### 4.3 Sistema de Sincronização

Para o sistema de sincronização, disparo de *trigger* e de marcadores foi usada uma placa genérica que contém um microprocessador Atmega164P. O micro controlador através de dois *mosfets* de potência actua sobre os marcadores e entrada de *trigger* durante o tempo previamente predefinido.

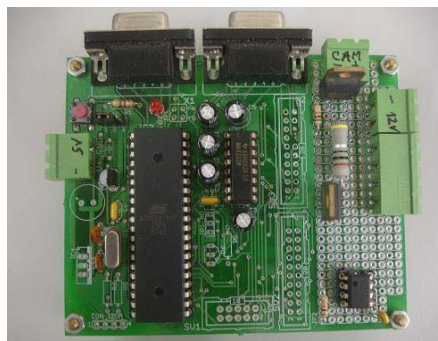


Figura 4.3.1 Placa genérica Atmega164P

Este circuito estava em funcionamento no sistema Sincrovision já construído anteriormente, sendo que as suas características e funcionalidades já estavam bem testadas.

## 4.4 Manipulador industrial

Para um trabalho sobre programação rápida de manipuladores industriais nada melhor que ter acesso a um manipulador para testes. Assim temos a noção de alguns problemas reais, sobre comunicações, limitações de movimentos, configurações etc.

A empresa MOTOMAN cedeu gentilmente um manipulador robótico que tem a capacidade de 20 kg de carga no punho.



Figura 4.4.1 Manipulador Motoman

Estudou-se o modo de programar e configurar este tipo de manipulador. Como este tipo de manipuladores tem muita precisão, até à centésima de milímetro, foram escritos pequenos programas para *debug* do sistema estereoscópico desenvolvido – desde simples movimentos com translação e rotação, como percursos mais elaborados. Na foto pode-se ver o manipulador em acção com os marcadores.

O manipulador foi usado para verificar se o sistema estereoscópico estava a medir bem translações e rotações dos marcadores bem como para a parte de geração de trajectórias. Usado nos testes de força versus corrente consumida pelo motor da máquina de polimento e será utilizado futuramente no processo de polimento robotizado.

## 4.5 Sistema Pneumático

O sistema pneumático serve para que o manipulador possa segurar as peças a polir durante esta actividade. O sistema montado é composto por um compressor, um regulador de pressão para definir a pressão necessária ao funcionamento do *gripper* e um bloco de válvulas pneumáticas. As características do material são apresentadas a seguir. No *gripper* foram efectuadas alterações para que fosse possível segurar as peças apresentadas na secção 4.9.



Figura 4.5.1 Gripper usado

Características:

### ***Gripper* PGN 64 -2 370 150**

- Força aos 6 bars: 420 N;
- Peso máximo da peça a trabalhar: 2.5 kg;

Tempo de abertura e de fecho: 0.02 s.

### **Regulador de pressão:**

- Pressão máxima – 15 bars.

### **Bloco de válvulas:**

- Válvula de uma bobine;
- Pressão máxima: 8 bars;
- Tensão de fecho: 12 Volts DC.



## 4.6 Sensor de força

Na actividade de polimento é necessário ter uma boa estimativa da força exercida sobre a peça que estamos a polir. Para se ter uma ideia do valor da força foi usado um sensor JR3 analógico de 6 graus de liberdade de forma a procedermos a algumas experiências. Este sensor já vem preparado para ser acoplado ao punho de um manipulador industrial e a sua aparência pode ser visualizada na Figura 4.6.1. O sensor que dispomos é ligado a uma placa PCI e através dela podemos obter os valores medidos pelo JR3. Este sensor tem a capacidade de medir forças e binários segundo 6 eixos, 3 como mostrados na Figura 4.6.2, mais 3 de torção, ou seja o binário exercido sobre cada um dos eixos ilustrados.

A saída do sensor é em Newton, sendo que é necessário ajustar escalas para que se consiga obter este valor correctamente. Não foi necessário este ajuste, pois o sensor já estava configurado com as escalas correctas.



Figura 4.6.1 Sensor de força JR3

Surgiu a motivação, em particular para a actividade de polimento, da substituição do sensor JR3 por um sensor mais barato. Foram efectuados testes de forma a se encontrar uma relação válida entre a força exercida pelo manipulador sobre o disco de polimento e a corrente consumida pelo motor eléctrico que o faz girar. O sistema de monitorização de corrente está descrito na secção seguinte.

Ao início deparou-se com algumas dificuldades em colocar este sensor a funcionar na plataforma Linux. Encontrou-se um driver para Linux num local Web de um outro projecto muito interessante de Mário Prats [29], mas que não funcionava nas novas versões de Linux.

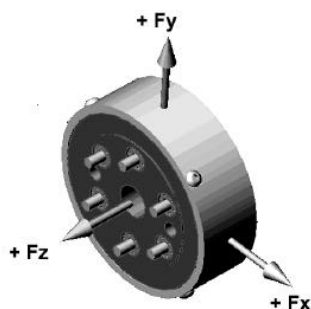


Figura 4.6.2 Representação de 3 dos 6 graus de liberdade mensuráveis pelo JR3. Imagem retirada da folha de características do sensor.

O driver para Linux foi actualizado, nas bibliotecas de que depende para a sua instalação. Como se sabe o software *Open Source* está constantemente a ser actualizado, e por vezes é necessário fazer uma revisão ao software já desenvolvido, principalmente quando são necessárias bibliotecas de sistema. Ao autor do driver Linux Mário Prats [29] foi-lhe comunicado esta alteração tendo sido criado um script que escolhe automaticamente a biblioteca a ser usada consoante a versão de kernel instalada. Foram implementados também alguns procedimentos de forma a aplicações desenvolvidas em Lazarus poderem efectuar a leitura dos valores do sensor de força - Figura 4.6.3. Futuramente, procedimentos para a escrita de escalas e escolha de filtros serão implementadas. Não tendo sido usada a plataforma Windows, mesmo assim fica a referência onde poderá ser encontrado o driver para este sistema operativo - página pessoal de Norberto Pires [30].

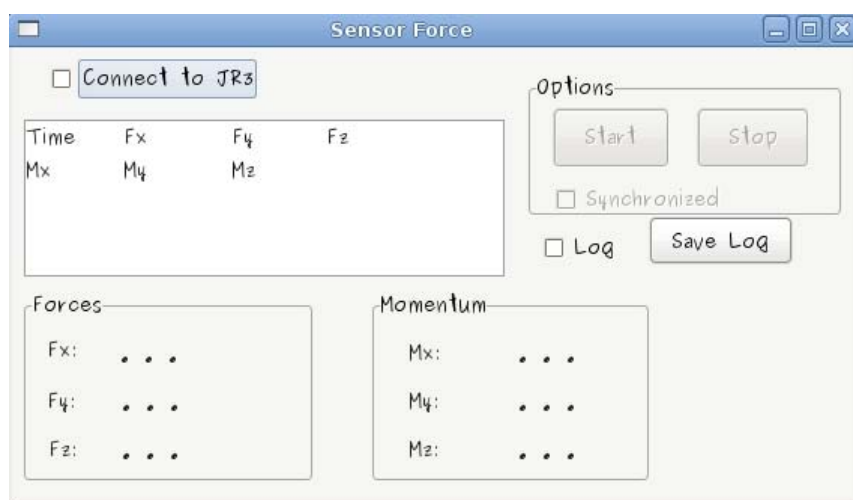


Figura 4.6.3 Aplicação JR3

## 4.7 Máquina de Polimento

Para o estudo da relação da força aplicada sobre as peças a polir versus corrente consumida pelo motor trifásico, a NS-Máquinas cedeu uma máquina de polimento.

Esta máquina é constituída por um motor trifásico e um disco de polimento, com a possibilidade de se poder duplicar as características anteriores - 2 motores e dois discos, ver Figura 4.7.1



Figura 4.7.1 Máquina de polir

Obteve-se uma ideia do funcionamento do conjunto do sistema – manipulador, sistema pneumático, máquina de polimento.

## 4.8 Sistema de monitorização de Corrente

Com a motivação apresentada na secção 4.6 era necessário ter um sistema a monitorizar a corrente do motor trifásico que faz girar o disco de polimento. Para este efeito usou-se o analisador de energia Mach Smart da Ducati, que permite a leitura e monitorização dos gastos energéticos de um motor trifásico (ex: Potência média consumida, Correntes em cada fase, Tensões em cada fase, tensões compostas, entre outras).

Este dispositivo tem a possibilidade de se efectuarem leituras dos valores medidos através do protocolo RS-485. Foi desenvolvida uma aplicação modelo de leitura dos valores medidos pelo analisador de energia da DUCATI -Figura 4.8.2.



Figura 4.8.1 Analisador de energia Mach Smart da Ducati

Com o início dos testes e *logs* sincronizados efectuados, foi detectado que o aparelho de medida Mach Smart apenas coloca disponíveis os valores medidos à frequência de 1 Hz - embora ele internamente trabalhe a uma frequência de amostragem de 1800 Hz. Abandonou-se a ideia do uso deste sensor, sendo apenas usado para confirmar os valores obtidos no circuito apresentado a seguir.

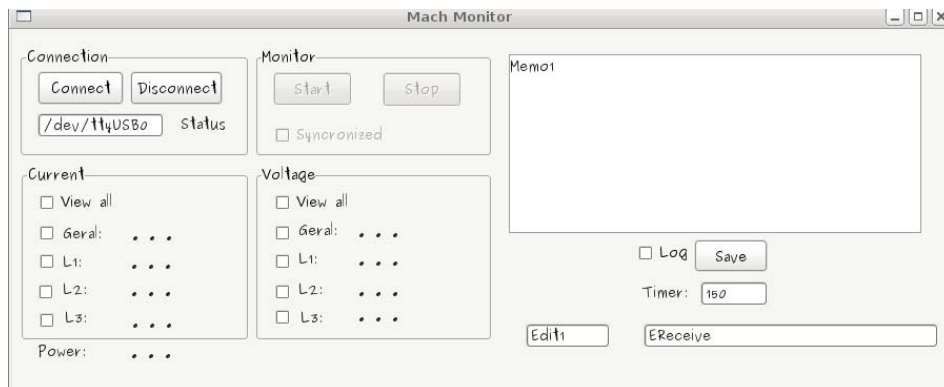


Figura 4.8.2 Aplicação Mach Monitor

Foi desenvolvido um circuito de leitura da corrente de uma fase do motor trifásico, que funcionasse à frequência de 1 kHz. Este circuito desenvolvido é composto por um circuito de condicionamento de sinal (em que faz um ajuste de escala e amplifica o sinal aplicado à sua entrada) e um LEM LTS 15 NP (a sua saída é em tensão e varia linearmente com os valores e sentido das correntes medidas que o atravessam) - Figura 4.8.3 e Figura 4.8.4. Para leitura da tensão de saída do LEM foi usada uma placa de circuito impresso que contém uma montagem genérica de um micro controlador atmega8. O micro controlador efectua leituras através do seu ADC, processa a informação adquirida e envia os dados por RS-232 para a aplicação desenvolvida para o efeito.

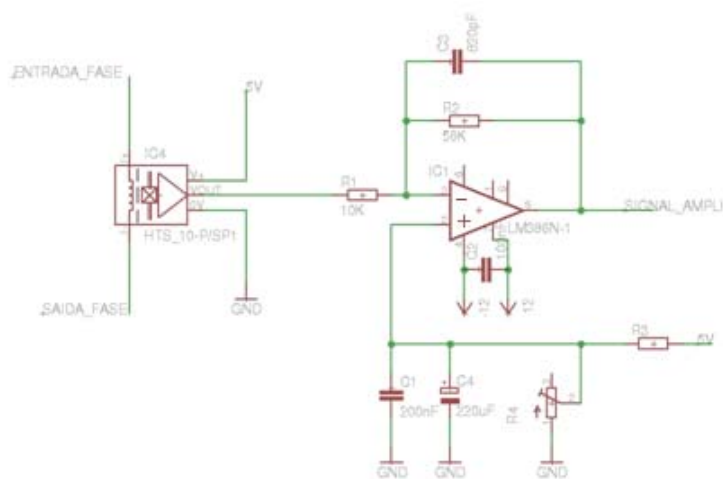


Figura 4.8.3 Circuito de Condicionamento de sinal

Este circuito está a funcionar com uma frequência de amostragem de 1kHz o que é suficiente para se conseguir a reconstrução da onda a medir - 50Hz.

A saída do nosso sensor esta representada na Figura 4.8.4 e é linearmente dependente com a corrente e direcção que o atravessa.

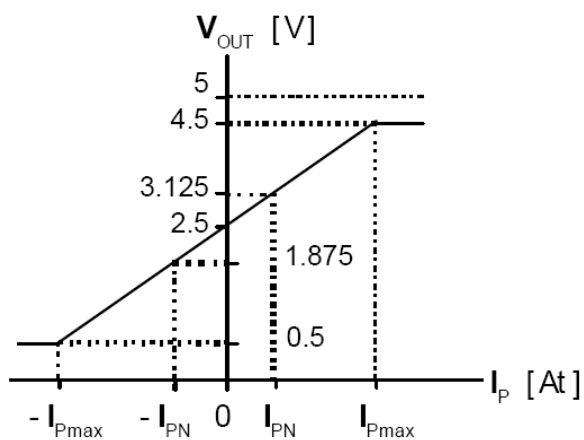


Figura 4.8.4 Saída em tensão do LEM

$-I_{Pmax}/ I_{Pmax}$  - Corrente máxima no primário, neste sensor +/- 45A.

$-I_{PN}/I_{PN}$  – Corrente nominal no primário, neste sensor +/- 15 A.

O conversor analógico digital – *analog to digital converter ADC* - do micro controlador tem 8 *bits* e o seu valor de fim de escala usado foi de 5V obtendo uma resolução de 19.6 mV.

Foram analisados os consumos do motor trifásico através do analisador da Mach, as correntes em vazio, ou seja, sem nenhuma peça no disco de polimento, rondam em

média os 4.10 A, em carga estimou-se por observação de resultados obtidos em testes, que a corrente máxima que o motor poderia consumir, seria 6.5 A. Projectou-se então um circuito para leituras até 7.5 A de valor eficaz – 10.6 A de valor de pico, valor medido pelo LEM. A tensão na saída do LEM varia segundo a expressão:

$$V_{out} = 2.5 \pm 0.625 \left( \frac{I_p}{I_{PN}} \right) \text{ Volts} \quad (4.1)$$

Em que  $I_p$  é a corrente que está a ser medida no LEM. Podemos notar que para  $I_p = 0$ ,  $V_{out} = 2.5 \text{ V}$ . Para os nossos valores de correntes máximos esperados a tensão vai variar da referência 2.5V de - Figura 4.8.5.

Corrente de pico (A)	Saída do LEM (V)	Diferença visualizada na saída
-10.6	2.05	450 mV
- 5.8	2.25	250 mv
5.8	2.74	240 mv
10.6	2.94	440 mv

Figura 4.8.5 Valores Máximos e mínimos do circuito de leitura de corrente

Note-se que a saída do LEM é linear com a intensidade e sentido da corrente, como estamos a trabalhar com ondas em AC sinusoidais, iremos obter valores positivos e negativos.

Então para uma corrente eficaz de 7.5 A apenas iríamos ter na saída do LEM uma diferença de meio volt, foi então projectado um circuito amplificador e de ajustamento de escala.

Na Figura 4.8.3 está representado o circuito de condicionamento de sinal. É composto por um amplificador operacional em que na sua entrada negativa temos a saída em tensão do LEM seguido de uma montagem amplificadora de ganho 5.6, como se pode deduzir pelas resistências  $R_1$  e  $R_2$ . Na entrada positiva temos um divisor resistivo de forma a efectuar um ajustamento do ponto de funcionamento, para assim obter uma maior resolução de leitura de corrente.

Através deste circuito de condicionamento de sinal obtemos à saída 0 V para -10.6 A e 5.04V para 10.6A de pico.

Para o cálculo do valor eficaz da corrente é usada a expressão:

$$v_{eff} = \sqrt{\frac{1}{T} \int_0^T v^2(t) dt} \quad (4.2)$$

Sendo o sinal amostrado é aproximada por:

$$V_{\text{eff}} = \sqrt{\frac{1}{N} \sum_{i=1}^N V_i^2} \quad (4.3)$$

Para o cálculo do valor eficaz, o somatório do quadrado dos valores medidos está a ser efectuado pelo próprio micro controlador, sendo este valor enviado a cada 100 ms via porta série para a aplicação Sincrovision. O restante cálculo é efectuado dentro da aplicação, para ser processado o seu valor correcto.

As medidas efectuadas por este circuito foram comparadas com as medidas efectuadas pelo Mach, sendo que a diferença de valores obtidos não eram muito diferente e eram fiáveis. Conseguimos assim obter valores eficazes da corrente com uma resolução de 80 mA de 100 em 100 ms, tempos impostos pela captura do sistema estereoscópico.

Na Figura 4.8.6 podemos ver o aspecto da nova aplicação desenhada para a leitura da corrente através do sensor LEM.



Figura 4.8.6 Aspecto da aplicação de monitorização de corrente

## 4.9 Peças para polimento

Para a actividade do polimento a empresa NS-Máquinas cedeu um conjunto de peças de teste para polimento. Com estas peças foram efectuados os *logs* e testes de monitorização de corrente e força.

Foi enviada uma peça modelo, já polida. Podemos ver na Figura 4.9.1 a peça no topo da imagem é a peça modelo, e a peça no fundo da imagem é uma das peças enviadas antes de entrar na máquina de polimento.



Figura 4.9.1 Peças para polimento

Nesta fase do trabalho não era objectivo proceder ao polimento das peças, elas foram usadas para testes e aquisição de medidas.



## Capítulo 5

# Sistema Sincrovision

Neste capítulo são descritas as alterações efectuadas à aplicação já existente a funcionar com o sistema Sincrovision. Até aqui apenas era processado e determinado a posição de um marcador no mundo. Após este período de trabalho, novas funcionalidades foram adicionadas ao programa, entre as quais:

- ✓ Identificação e determinação da posição e da rotação de 4 marcadores colocados na ferramenta do operador.
- ✓ Monitorização/armazenamento de trajectórias efectuadas pela ferramenta.
- ✓ Aquisição dos valores das correntes do motor trifásico da máquina de polimento, com a opção de efectuar a medida em modo sincronizado com o sistema de visão ou em modo livre.
- ✓ Aquisição dos valores do sensor de força, com a opção de efectuar as medidas em modo sincronizado com o sistema de visão ou em modo livre.

São também abordados os passos efectuados para a implementação e testes dos algoritmos escolhidos.

### 5.1 Calibração Tsai

Para obter os parâmetros intrínsecos e extrínsecos da configuração do sistema estereoscópico montado, foi efectuada a calibração do mesmo. Foram utilizados os algoritmos Tsai e *eight point*. No entanto, neste último é necessário ajustar a escala da reconstrução, o que ainda não se conseguiu implementar de forma automática. O Tsai é o algoritmo mais usado pois permite uma reconstrução 3D métrica directa, após a obtenção dos parâmetros das câmaras.

Começamos por definir um referencial tridimensional no plano de calibração, para explicar as rotações efectuadas.

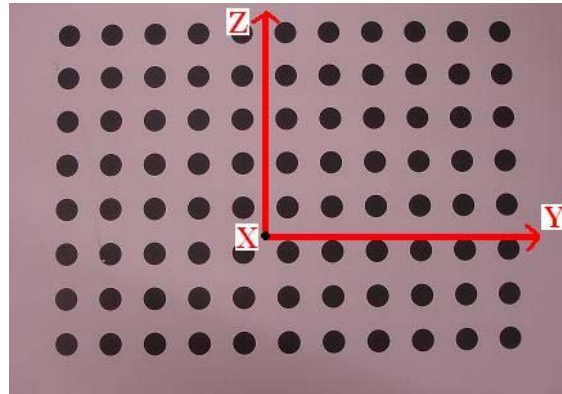


Figura 5.1.1 Eixos de rotação

Como já referido na secção 3.5.1, o plano de calibração não pode estar perpendicular aos planos de imagem das câmaras. Para um sistema de uma só câmara, é com alguma facilidade que se coloca o plano de calibração numa posição propícia à mesma. Num sistema estereoscópico, com a configuração escolhida, é difícil o posicionamento do plano de calibração segundo a rotação sobre o eixo Z - Figura 5.1.1. Por exemplo, quando se coloca o plano de calibração não perpendicular para a câmara número 1 - ver Figura 5.1.2 a) - na câmara 2 o plano de calibração fica quase na perpendicular - ver Figura 5.1.2 b).

Efectuaram-se alguns testes e verificou-se que, a decisão de rotação sobre este eixo Z não era a mais adequada, devido a terem-se obtido valores incorrectos e não fiáveis.

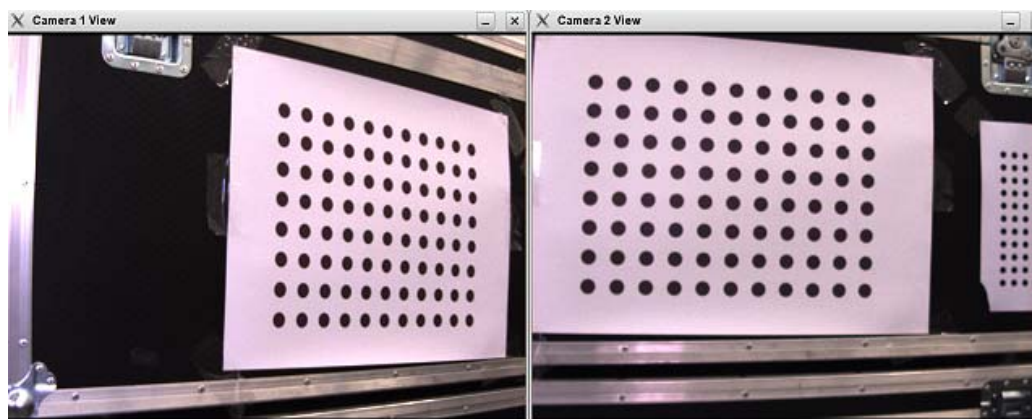


Figura 5.1.2 Rotação segundo eixo Z- a) - vista para a câmara 1, b) vista para a câmara 2

Optou-se pela rotação segundo outro eixo, rotação sobre o eixo Y, em que permitia a variação do ângulo do plano de calibração, garantindo a não perpendicularidade do plano às câmaras. A Figura 5.1.3 exemplifica uma parte de uma sequência de imagens para calibração do algoritmo Tsai.

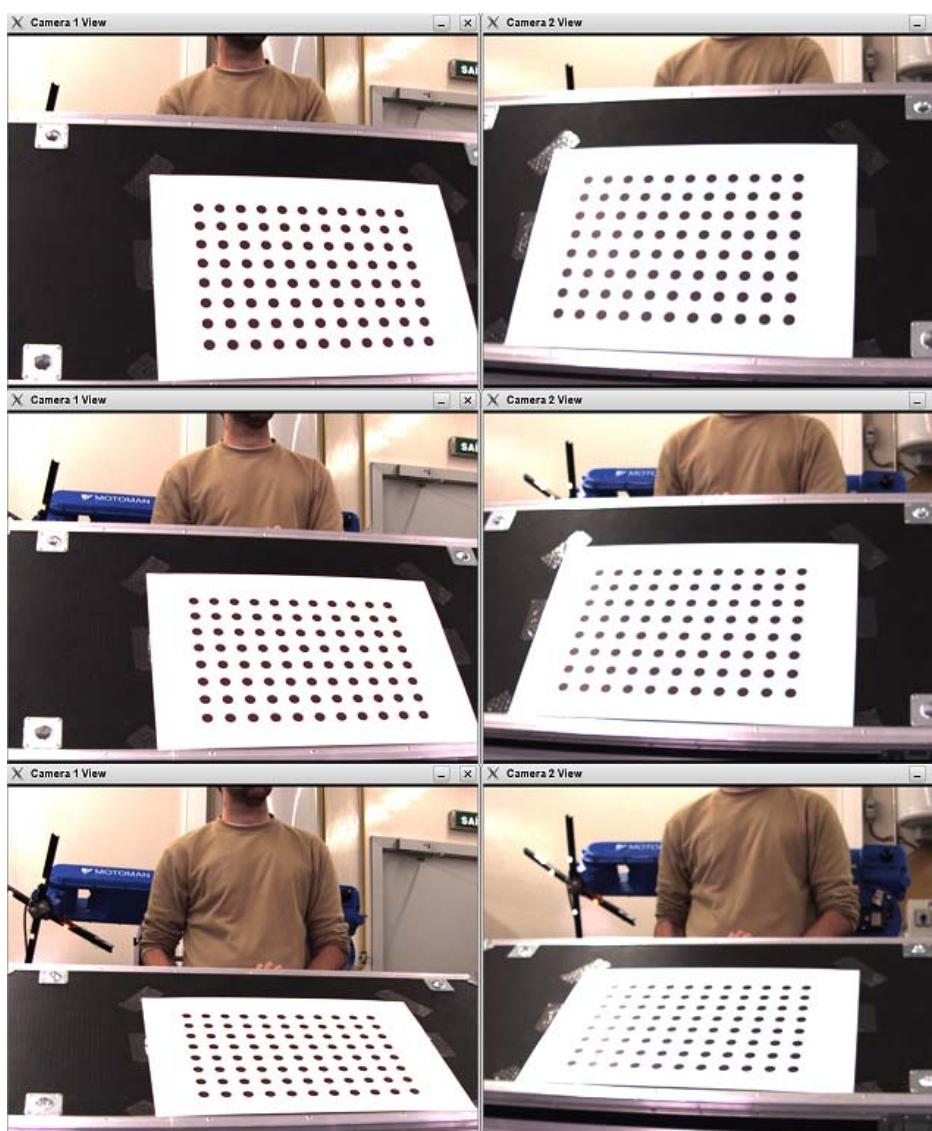


Figura 5.1.3 Começando a referir por cima - Teste 2, 4 e 6 efectuados para a calibração Tsai.

Exemplo da sequência efectuada do processo de calibração Tsai:

- i) Captura das imagens - Figura 5.1.4;
- ii) *Threshold* das imagens - Figura 5.1.5;
- iii) Localização dos pontos de calibração e extracção das suas posições - Figura 5.1.6;

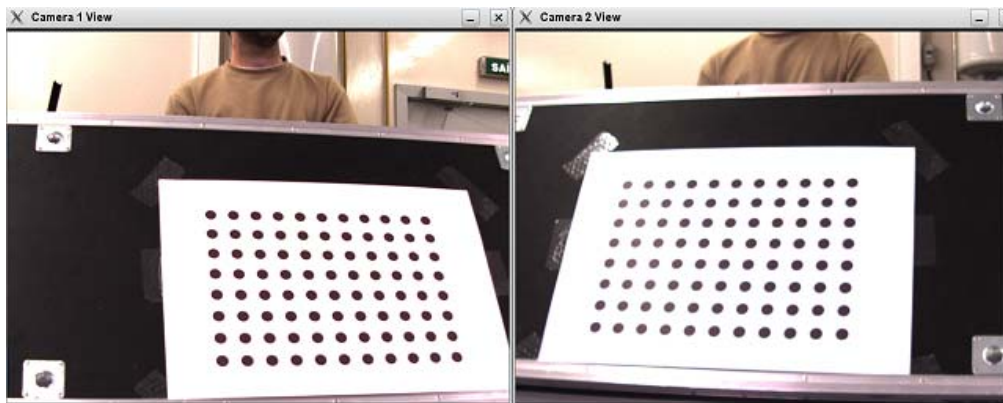


Figura 5.1.4 Captura dos planos - Teste 2

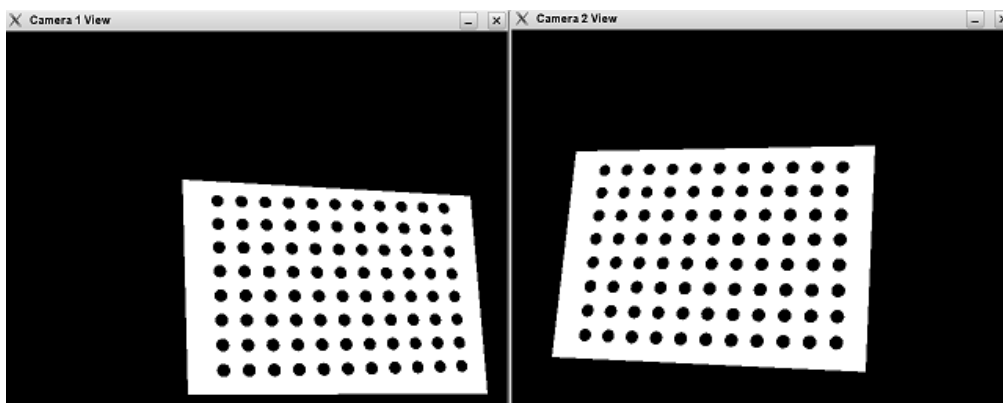
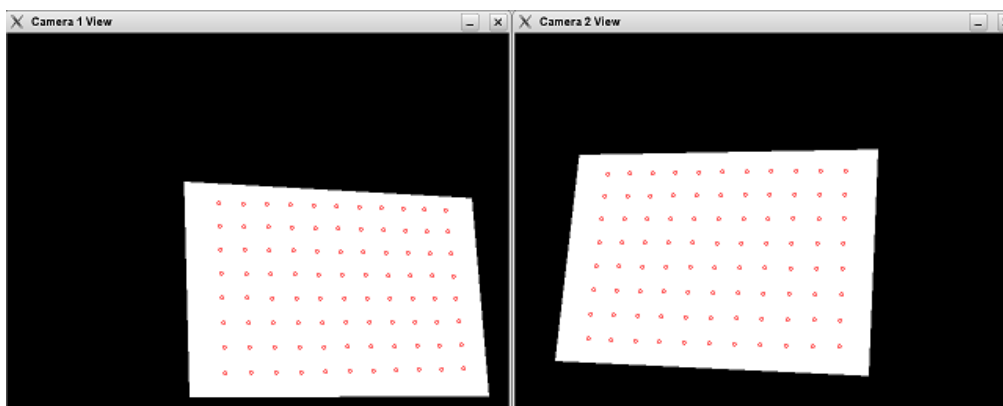
Figura 5.1.5 Aplicação de *thresh*

Figura 5.1.6 Localização de Pontos de calibração

Alguns resultados à medida que se inclinava o plano de calibração estão representados na tabela 5.1.1. Podemos visualizar a variação de alguns dos parâmetros intrínsecos e extrínsecos com a posição do plano de calibração.

5.1.1 Alguns Parâmetros intrínsecos calculados

Parâmetros		f	Tx	Ty	Tz
Ensaio					
1	Câmara 1	0.004875	-0.049413	0.241497	1.030214
	Câmara 2	0.005719	-0.284669	0.196879	1.257953
2	Câmara 1	0.005274	-0.052935	0.226462	1.137745
	Câmara 2	0.005968	-0.281696	0.180703	1.327709
3	Câmara 1	0.005576	-0.056073	0.223467	1.221038
	Câmara 2	0.006008	-0.278884	0.176837	1.351057
4	Câmara 1	0.005935	-0.060027	0.222717	1.322871
	Câmara 2	0.006035	-0.275045	0.175178	1.375974
5	Câmara 1	0.00615	-0.063986	0.225072	1.393956
	Câmara 2	0.006116	-0.271115	0.176864	1.413817

Os valores desta tabela são os resultados da calibração da sequência efectuada e mostrada na Figura 5.1.3 – na figura apenas estão representados três passos da sequência. Nesta sequência, o plano de calibração estava a aproximadamente a noventa graus com a horizontal e foi-se diminuindo o ângulo do plano até o sistema de visão não ser capaz de detectar os pontos de calibração. Os valores estão apresentados de forma decrescente do ângulo do plano de calibração. Do teste número 6 da Figura 5.1.3 não foi possível extrair valores fiáveis.

Salienta-se que o ângulo do plano de calibração não foi medido nesta experiência. Para trabalho futuro, poder-se-á calcular o valor ideal deste parâmetro para este tipo de calibração.

É observada na tabela a variação da distância focal  $f$  e os parâmetros de translação de cada câmara ao plano de calibração. Observa-se um constante aumento da distância focal consoante se diminui o ângulo com a horizontal. O valor que melhor se aproxima do correcto é o ensaio número 4 e número 5, pois são os únicos em que para as duas câmaras as distâncias focais estão perto de 6 mm – este é o valor esperado, pois são usadas lentes de 6 mm no sistema de visão. Usaram-se estes dois valores para alguns ensaios que são apresentados nas seguintes secções.

Pode-se então concluir que é necessário ter em atenção o processo de calibração, para se obter correctamente os parâmetros pretendidos.

## 5.2 Teste da Geometria epipolar

Após extracção dos parâmetros intrínsecos e extrínsecos do sistema estereoscópico, foram efectuados grupos de testes para verificação dos valores obtidos. O teste da geometria epipolar corresponde ao primeiro passo num total de 3, para a verificação da fiabilidade dos valores calculados. Recorreu-se à geometria epipolar para confirmação dos cálculos e dos parâmetros intrínsecos calculados no passo anterior.

Através da geometria epipolar sabemos que se conhecemos um ponto na primeira vista, o par correspondente na imagem 2, encontra-se sobre a linha epipolar. A Figura 5.2.1 mostra a linha epipolar obtida, recorrendo ao teste da geometria epipolar.



Figura 5.2.1 Linha epipolar calculada através do ponto seleccionado na imagem 1

Quando dois pontos estão sobre a mesma linha horizontal na primeira imagem, existe ambiguidade na determinação do seu par; isto porque existem 2 pontos na segunda imagem que estão sobre a mesma linha epipolar. Neste caso, utiliza-se outro critério para fazer a correcta identificação de pares de pontos. No sistema Sincrovision, tal facto ocorre quando dois marcadores de cores iguais ficam sobre a mesma linha. Existem várias formas de resolver esta ambiguidade. A solução implementada é calculada com base na distância de cada um dos pontos à linha epipolar – aquele que se encontrar a uma distância menor da linha é o que lhe pertence. Caso ocorra que a distância à linha epipolar seja a mesma, baseamo-nos na geometria do sistema e faz-se a correspondência directa de pontos – o ponto mais à esquerda/direita na imagem número dois corresponde ao ponto à esquerda/direita da imagem número 1.

## 5.3 Teste reconstrução 3D

Para a segunda validação do algoritmo de calibração, efectuou-se a reconstrução 3D dos pontos do plano de calibração. Na Figura 5.3.1, é possível observar, em plano de fundo, a imagem captada pelo sistema estereoscópico e, no canto superior direito, a

representação gráfica da reconstrução dos pontos do plano de calibração usados neste ensaio.

Graficamente podemos ver que a reconstrução 3D está a ser executada, já que os planos de calibração mais pequenos aparentam estar perpendiculares entre si, como estavam na realidade.

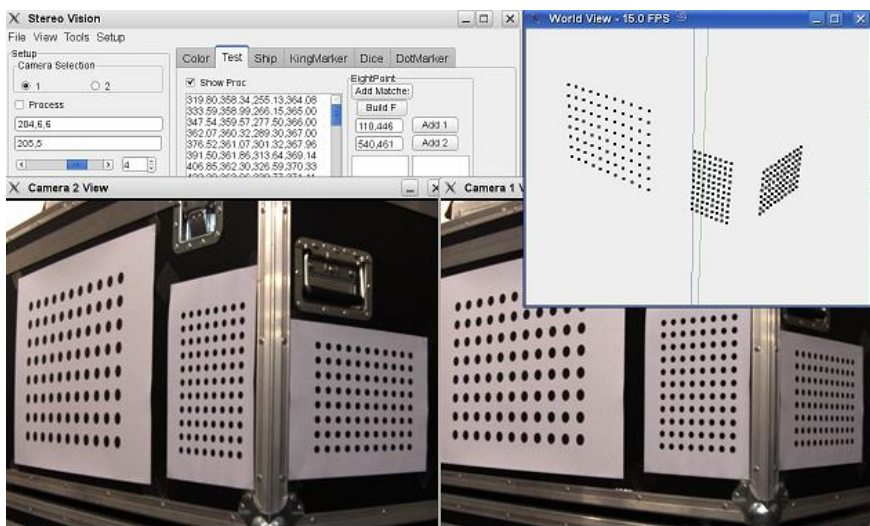


Figura 5.3.1 Teste reconstrução 3D

Podemos observar na Figura 5.3.2, na figura da esquerda a reconstrução de um plano de calibração. Embora os resultados das calibrações sejam satisfatórias, vemos graficamente na imagem da direita, após rotação do plano de calibração que as lentes podem introduzir alguns erros, como demonstrado na Figura 5.3.2, denominado efeito de barril.

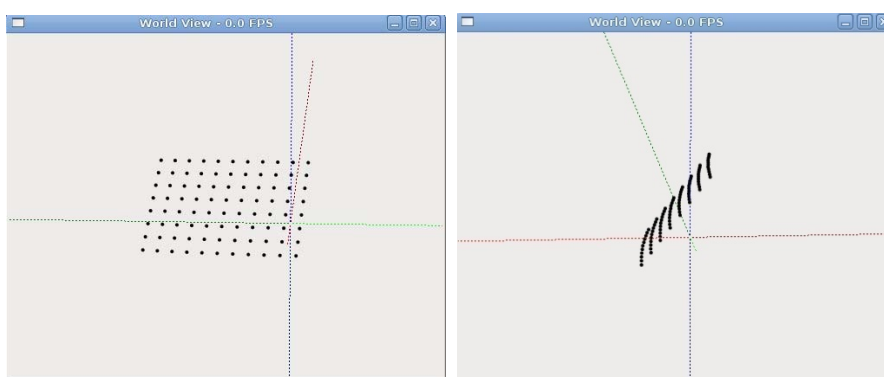


Figura 5.3.2 Efeito de Barril

O efeito de barril é detectado na reconstrução de linhas rectas, estas deixam de ser rectas e passam a linhas distorcidas.

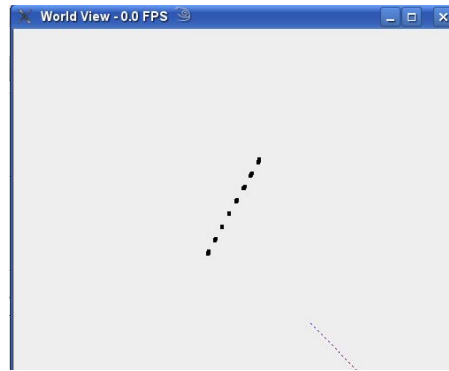


Figura 5.3.3 Correção do efeito de barril

Para compensar este fenómeno tem que ser aplicado um filtro sobre os pontos calculados, usando funções próprias para este efeito, foi então corrigido este efeito. Pode-se observar na Figura 5.3.3, a reconstrução do mesmo plano, aplicando a correção do efeito de barril.

## 5.4 Posição e rotação da ferramenta

O plano onde assenta a ferramenta é determinado através de mínimos quadrados de forma a se obter a melhor aproximação aos quatro marcadores. A rotação da ferramenta sobre o eixo do plano será dada através da posição de um dos marcadores, que tem cor diferente dos restantes três.

São necessárias, no mínimo, duas cores diferentes. Inicialmente, foram utilizadas as cores branca e vermelha. Após alguns testes, concluiu-se que o sistema por vezes confundia reflexos de luz com o marcador de cor branca. Cada vez que isto acontecia era necessário executar nova calibração de cores ou alterar a abertura dos diafragmas das câmaras.

Optou-se por substituir a cor branca pela cor amarela, para assim aumentar a robustez do sistema – de modo a não ser afectado pelas repetidas alterações de luminosidade no ambiente de trabalho. Após esta permuta, notou-se uma significativa melhoria.



O modelo idealizado para o cálculo da posição e rotação da ferramenta é feito através dos 4 marcadores luminosos, consoante o seguinte procedimento:

- i) Cálculo do ponto central, a que chamamos *centroid* – ponto verde na Figura 5.4.1.
- ii) Cálculo da equação do plano definido pelos 4 marcadores e cálculo do seu vector normal – vector  $n$ ;
- iii) Cálculo do vector  $R_M$  que vai entre o *centroid* e o marcador vermelho.
- iv) Cálculo do vector  $B$ , vector normal ao vector  $n$  e ao vector  $R_M$ . Fica assim definido automaticamente nosso terceiro eixo do referencial da ferramenta.

Para ajudar a compreender a geometria do modelo idealizado a Figura 5.4.1 mostra graficamente o pretendido.

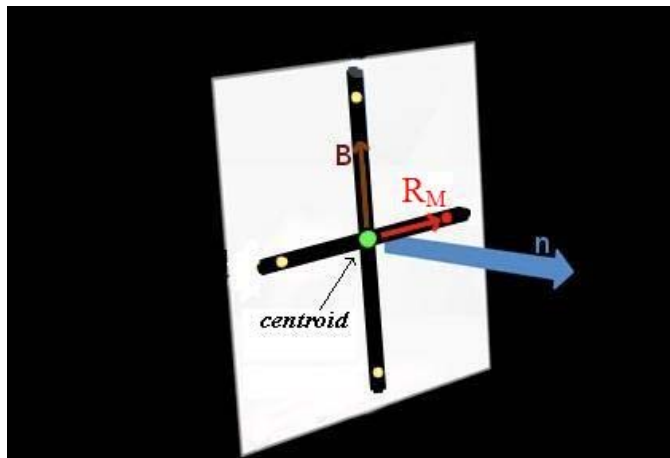


Figura 5.4.1 Modelo da ferramenta com marcadores

Durante o cálculo da equação do plano é obtido o vector normal a ele, sendo este vector a componente que é necessária para a orientação da ferramenta.

É feita uma aproximação por um plano aos quatro marcadores usando o critério dos mínimos quadrados. Eberly, David (2008) [19] propõe uma aproximação hiper planar de  $n$  pontos, usando uma regressão ortogonal, utilizada também neste trabalho, através das posições dos marcadores e calculando a matriz  $M$  definida a seguir.

$$\mathbf{M} = \begin{bmatrix} \sum_{i=1}^m (x_i - a)^2 & \sum_{i=1}^m (x_i - a) - (y_i - b) & \sum_{i=1}^m (x_i - a) - (z_i - c) \\ \sum_{i=1}^m (x_i - a) - (y_i - b) & \sum_{i=1}^m (y_i - b)^2 & \sum_{i=1}^m (y_i - b) - (z_i - c) \\ \sum_{i=1}^m (x_i - a) - (z_i - c) & \sum_{i=1}^m (y_i - b) - (z_i - c) & \sum_{i=1}^m (z_i - c)^2 \end{bmatrix}$$

Em que os valores  $a$ ,  $b$  e  $c$  são as coordenadas do centro dos 4 pontos – obtida pela posição média das suas posições.

Após definida esta matriz Eberly, David (2008) [19] comenta que é possível obter o vector normal, ao plano definido pelos 4 pontos, através da decomposição de valores singulares (SVD) a partir do produto  $A = M^T M$ .

O resultado obtido será  $A = USV^T$ , em que  $S$  é a matriz diagonal contendo os valores próprios de  $M$ , as colunas de  $V$  os vectores próprios, e  $U$  é uma matriz ortogonal. Na última coluna da matriz  $U$ , está o vector normal ao plano, associado ao menor valor próprio da matriz  $M$ .

O vector  $R_M$  é calculado a partir das posições do *centroid* e do marcador vermelho – ver Figura 5.4.1.

O vector  $B$  é calculado a partir do produto externo entre  $n$  e  $R_M$ .

É assim criado um referencial com origem no ponto *centroid*, com o eixo  $X$  segundo o vector  $n$ , o eixo  $Y$  segundo o vector  $R_M$  e o eixo  $Z$  segundo o vector  $B$  – ver Figura 5.4.2.

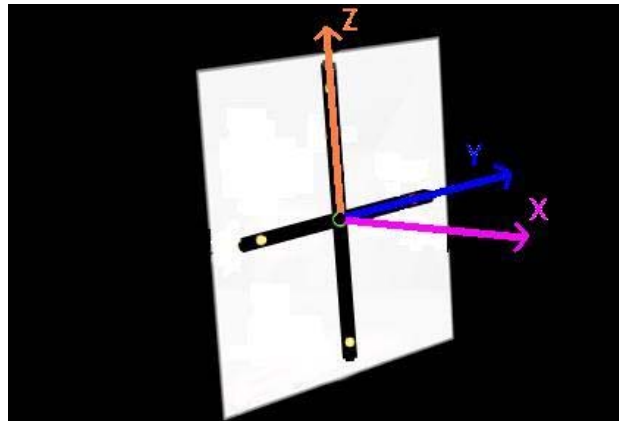


Figura 5.4.2 Eixos coordenados

É de realçar que a face do plano onde assentam os marcadores, é a face para a qual a componente  $x$  do vector  $n$  é positiva - ver Figura 5.4.2. Durante o cálculo do vector normal  $n$ , pode acontecer, para o mesmo plano, obter o mesmo vector mas com direcção oposta, coordenada  $x$  do vector  $n$ , negativa. Este resultado não faz sentido, pois significa que a ferramenta teria rodado  $180^\circ$  sobre o eixo  $Y$ , e se isto acontecesse na realidade os marcadores não estavam visíveis no sistema de visão artificial. Constatando-se tal facto, se durante os cálculos obtemos o vector da face oposta do nosso plano (componente  $x$  do vector  $n$  é negativa) então invertamos a direcção deste vector – corresponde a multiplicar o vector  $n$  pela constante  $-1$ .

Ficando assim definidos os eixos coordenados do manipulador – ver Figura 5.4.3.

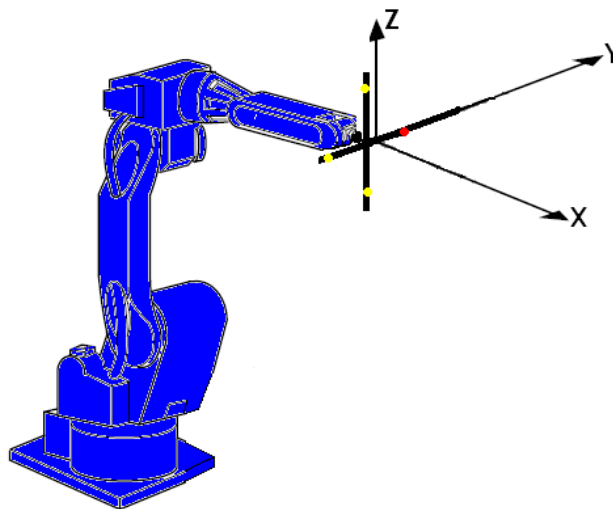


Figura 5.4.3 Eixos coordenados Manipulador

Após determinação do eixo de coordenadas da ferramenta, é possível determinar a sua rotação.

Usando os vectores calculados anteriormente retira-se a rotação sobre cada um dos eixos do referencial XYZ.

Definindo os seguintes vectores:

$$R_M = \begin{bmatrix} r_{m_x} \\ r_{m_y} \\ r_{m_z} \end{bmatrix} \quad (5.1)$$

$$B = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \quad (5.2)$$

$$n = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} \quad (5.3)$$

As rotações segundas os eixos X,Y e Z são  $R_x$ ,  $R_y$ ,  $R_z$  respectivamente podem ser calculadas através das seguintes expressões:

$$R_x = -\arctan2(b_y, b_z) * 180/\pi \quad (5.4)$$

$$R_y = -\arctan2(n_z, n_x) * 180/\pi \quad (5.5)$$

$$R_z = \arctan2(n_y, n_x) * 180/\pi \quad (5.6)$$

Através das expressões em cima são obtidas os valores dos ângulos em graus das rotações, sobre os respectivos eixos coordenados. Nas expressões (5.4) e (5.5) observam-se dois sinais negativos, em que para o cálculo dos ângulos por estas expressões são necessários, para se obter o resultado correcto conforme os ângulos do manipulador.

## 5.5 Validação de resultados

Este é o terceiro passo para a validação do sistema de calibração. Devido à grande precisão que este manipulador industrial oferece, decidiu-se fixar a cruz com os marcadores no manipulador para validação de todas as funções e modelos definidos.

O manipulador passou a mover-se com a cruz, de forma a existir uma forma fácil de efectuar rotações e translações com grande precisão dentro de uma área de trabalho apresentada na Tabela 5.5.1. Na Tabela 5.5.1 estão também representadas configurações de *setup* do sistema Sincrovision.

Em complemento, recorreu-se à aplicação ComMotoman que permite enviar alguns comandos via TCP para o manipulador, posicionando o manipulador nas posições desejadas. Em cada uma delas retiraram-se as medidas pretendidas através da aplicação Sincrovision. Alguns resultados são apresentados na tabela Tabela 5.5.2.

Tabela 5.5.1 Propriedades do setup

<b>Propriedades do setup</b>	<b>Valores</b>
Distância entre câmaras (mm)	780
Ângulo entre câmaras (°)	46.4
Distância dos marcadores às câmaras (mm) (No ponto origem e sem nenhuma rotação)	1900
Distância entre marcadores (mm)	200
Área de trabalho em X (mm)	700
Área de trabalho em Y (mm)	500
Área de trabalho em Z (mm)	400

Tabela 5.5.2 Medidas efectuadas pelo Sincrovision

Ponto Sincrovision	X (mm)	Y (mm)	Z (mm)	Rx (°)	RY (°)	RZ (°)
Ponto Motoman (x, y, z, Rx, Ry, Rz)						
1 (0, 0, 0, 0, 0, 0)	0.712	0.167	0.025	-0.324	2.00	1.155
2 (0, 0, 0, 0, 30°, 0)	-2.678	0.109	-6.227	-0.051	31.801	-0.581

2 (0,0,0,30°, -30°, 0)	-2.801	1.246	7.834	30.497	-29.188	-0.636
3 (200,0,0,30,0,0)	191.062	-9.04	-5.401	29.722	1.446	-0.327
4 (-300,300,300,0,0,0)	-285.383	315.127	308.406	-0.141	2.579	0.014

Tabela 5.5.3 Erros máximos posição e rotação

Ponto Motoman (x,y,z,Rx,Ry,Rz)	Erro Máximo obtido Posição (mm)	Eixo maior erro na posição	Erro Máximo Obtido Rotação (°)	Eixo maior erro na rotação
1 (0,0,0,0,0,0)	0.721	Em X	2	Em Y
2 (0,0,0,0,30°,0)	6.222	Em Z	1.8	Em Y
3 (0,0,0,30°, -30°, 0)	7.834	Em Z	0.812	Em Y
4 (200,0,0,30,0,0)	9.04	Em Y	1.446	Em Y
5 (-300,300,300,0,0,0)	14.617	Em X	2.579	Em Y

Na Tabela 5.5.3 estão alguns dos valores dos erros máximos cometidos pelo Sincrovision nas medidas efectuadas nos pontos presentes na Tabela 5.5.2. Estas tabelas apenas servem para dar uma ideia da precisão do sistema desenvolvido.

Nota-se um pequeno erro na rotação sobre o eixo Y, ainda não se conseguiu determinar a origem para este erro, mas com mais testes será possível perceber de onde surge este erro. Mas provavelmente, provém da fixação da cruz ao manipulador.

Tabela 5.5.4 Erros (%) em função da área total de trabalho

Ponto Motoman (x,y,z,Rx,Ry,Rz)	Erro em X (%)	Erro em Y (%)	Erro em Z (%)
1 (0,0,0,0,0,0)	0.102	0.0334	0.006
2 (0,0,0,0,30°,0)	0.38	0.0218	1.56
2 (0,0,0,30°, -30°, 0)	0.4	0.25	1.96
3 (200,0,0,30,0,0)	1.28	1.8	1.35
4 (-300,300,300,0,0,0)	2.088	3.02	2.10

A avaliação exhaustiva da precisão do sistema é deixada para trabalho futuro. Pelas medidas das tabelas em cima, conclui-se que para actividades que não necessitem de muita precisão este o sistema é suficiente – são obtidos erros máximos de 3 % ver Tabela 5.5.4.

Durante a operação, visualiza-se que à medida que se os marcadores se afastam do sistema de visão, deslocamento no sentido negativo do eixo X, o ruído nas medidas aumenta. Este efeito constata-se devido à maior dificuldade de detecção dos marcadores à medida que estes se afastam do sistema de visão.

Outra ideia que surgiu, para a validação de resultados, consistiu em fazer com que o manipulador percorresse um trajecto bem definido/conhecido, e colocar o sistema a medir automaticamente. Foram criados dois trajectos, um em forma de cubo e outro em forma de cubo deformado - ver Figura 5.5.1e Figura 5.5.2. Num primeiro teste, o manipulador percorria os vértices das figuras geométricas. No segundo teste, o manipulador executava o teste anterior e em cada vértice exercia rotações sobre os 3 eixos de coordenadas. Para uma rigorosa avaliação é também proposto que se repitam os testes anteriores, alterando as distâncias entre marcadores e assim tentar encontrar os limites do sistema - Distância máxima, mínima e ideal entre marcadores.

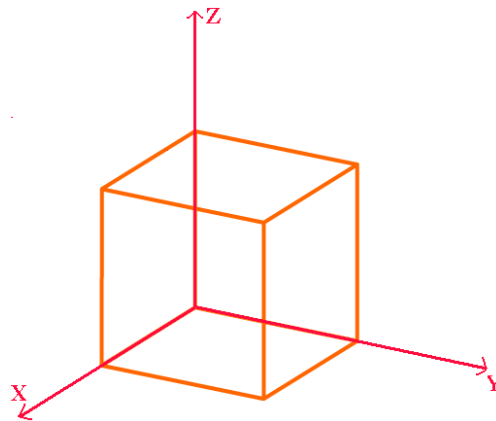


Figura 5.5.1 Cubo de teste

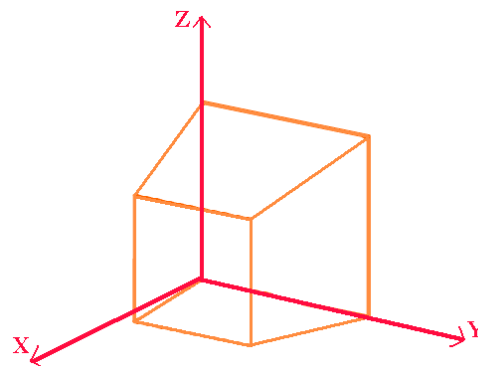


Figura 5.5.2 Cubo deformado

Foram efectuados alguns vídeos com as propostas anteriores, mas devido a alterações de *setup*, posições e modelos das câmaras, estes vídeos de nada servem para avaliação do novo sistema montado. Não foi possível a repetição dos testes e a aquisição das medidas pretendidas, por carência de tempo. No entanto, fica proposto para trabalho futuro a execução destes testes para avaliar a precisão do sistema.

Após a verificação da veracidade de todos os valores, estabeleceu-se a comunicação entre as aplicações Sincrovision e ComMotoman para colocar em prática os algoritmos testados. Os valores que são enviados pela aplicação Sincrovision são os da posição da ferramenta e rotação da mesma. No programa está definida uma área de trabalho limite para a ferramenta. Caso esta se localize fora dos limites, ou não seja possível a visualização dos quatro marcadores, o envio de comandos para o manipulador cessa e este pára. Esta é uma medida de segurança, que tem por objectivo evitar que o manipulador invada áreas não permitidas à sua volta, pois a segurança é primordial e factor a ter em conta neste tipo de trabalhos. Existe também um botão de emergência na consola do manipulador, que vem já embutido de fábrica, sendo possível, em caso de emergência, anular todas as manobras que se estejam a executar.

Conseguimos assim controlar o manipulador em tempo real. Algo de semelhante ao trabalho feito por Neto, Pedro et al (2009) [11].

Efectua-se então o passo seguinte, guardar os pontos do movimento da ferramenta capturados através do sistema de visão, para posterior análise numa outra aplicação.

O *setup* para o funcionamento do manipulador em tempo real é mostrado na Figura 5.5.3.

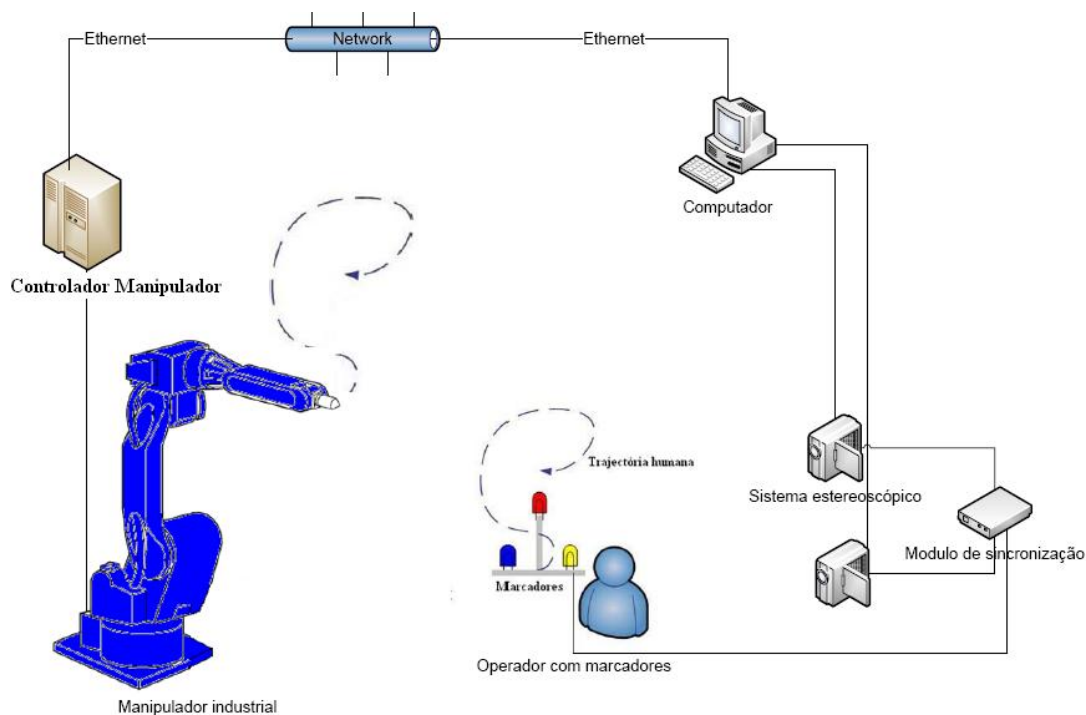


Figura 5.5.3 Configuração do sistema Sincrovision





# Capítulo 6

## Geração de trajectórias

### 6.1 Exposição do problema

Após executada e guardada uma trajectória em ficheiro, esta pode ser carregada para uma outra aplicação. Esta aplicação não funcionará em tempo real e nela é gerada a melhor trajectória para o manipulador.

Aleotti J e Caselli S. (2006) [2] no seu trabalho propõem a génese de trajectórias, recorrendo ao critério de NURBS (*Non Uniform Rational Basis Spline*). Os pontos do trajecto são introduzidos no computador através de uma luva antropomórfica e guardados para posterior análise utilizando o critério NURBS.

Este critério muito flexível e muito usado, principalmente na área de design, permite de uma forma simples a representação de superfícies livres. Pierre Bézier e Paul de Casteljaou, criadores do critério NURBS, trabalhavam ambos na indústria automóvel e necessitavam de uma ferramenta flexível e eficaz para a representação dos contornos dos automóveis e representação dos chassis.

O seu uso permite a representação de formas geométricas de uma forma compacta dependendo de dois parâmetros mapeados, na superfície tridimensional pretendida e a que se pretende aproximar. Estes dois parâmetros são denominados pontos de controlo e podem ser colocados em posições estratégicas de forma a manipular a forma da curva – ver Figura 6.1.1.

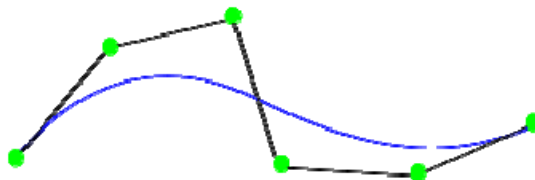


Figura 6.1.1 Caminho gerado pelo NURBS

## 6.2 Abordagem adoptada

Foi feito um estudo mais aprofundado deste algoritmo com o uso de *Splines*. No entanto, a aplicação deste critério não foi efectuada, devido ao facto de o manipulador ter predefinido duas funções e o objectivo ser o seu uso - instruções *MOVL* e *MOVC*. Para a utilização destas instruções, as expressões dos percursos calculadas seriam inúteis. Apesar disso, é deixada a referência do livro onde pode ser encontrada informação adicional sobre este assunto - Foley, James D. (1990) [21]. Para se poderem definir a sequência de operações a efectuar ao conjunto de pontos guardados, começou-se por analisar as limitações e condições que cada instrução acarretava, de forma a otimizar os algoritmos - estas condições são explicadas ao longo das próximas secções.

Decidiu-se começar por analisar o ruído que afecta a amostra de pontos guardada. Após filtragem da amostra, agrupam-se os pontos por velocidades iguais e tenta-se efectuar uma aproximação por rectas ou circunferências.

De seguida, são descritas cada uma das instruções.

## 6.3 Instruções a usar no manipulador

### 6.3.1 Instrução *MOVL*

Nesta instrução, é necessário definir o ponto final e a velocidade e opcionalmente o *position level*. Chama-se a atenção para o facto de o ponto inicial das instruções é o ponto actual onde se encontra o manipulador.

A instrução *MOVL*, para este manipulador, é escrita na forma:

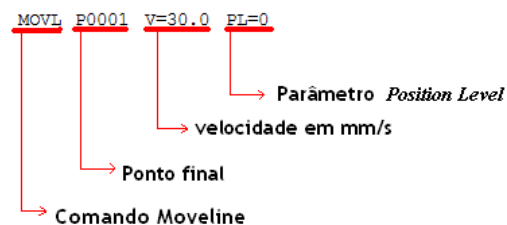


Figura 6.3.1 Instrução *MOVL*

O último parâmetro opcional denominado *PL* (*position level*) define a aproximação do manipulador ao ponto final entre instruções. Este parâmetro tem 8 níveis possíveis e a Figura 6.3.2 mostra quais os seus efeitos.

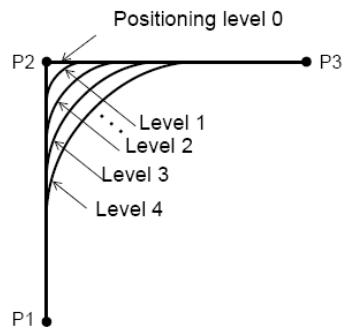


Figura 6.3.2 *Position Level* - retirada dos manuais da MOTOMAN

Se este parâmetro não for definido, ele variará consoante a operação a executar e a velocidade do manipulador. [20]

### 6.3.2 Instrução *MOVC*

Com a instrução *MOVC* o manipulador move-se sobre o arco definido por três pontos - ver Figura 6.3.3.

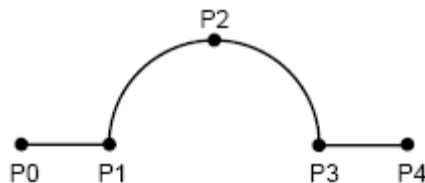


Figura 6.3.3 Resultado da instrução *MOVC*

Dados três pontos ao manipulador P1, P2 e P3 o manipulador move-se segundo um arco de circunferência. Notar que se é programada uma interpolação linear até P0, o manipulador mover-se-á de P0 até P1 em linha recta. [20]

Se pretendermos ter dois arcos de circunferência seguidos como apresentado na Figura 6.3.4.

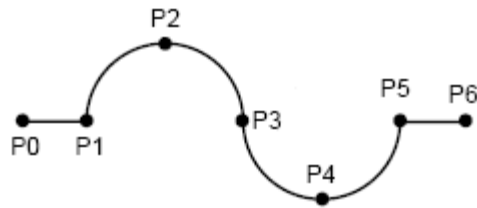


Figura 6.3.4 Circunferências consecutivas

Transmitindo ao manipulador os pontos P1 ... até P5, este não executa o movimento pretendido [20]; Uma explicação deste facto é dada seguidamente.

Para se conseguir efectuar a trajetória definida pelos pontos P1 a P5, entre as circunferências, precisamos de introduzir uma instrução de movimento de junção *MOVJ* ou um *MOVL*. Esta deve ser inserida no ponto idêntico às duas circunferências – ver Figura 6.3.5.

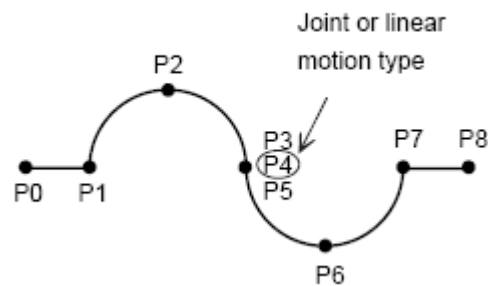


Figura 6.3.5 Circunferências consecutivas de forma correcta - figura retirada do manual de programação do manipulador

Assim sendo o ponto P3 é replicado 3 vezes, um é o ponto final da primeira circunferência, outro - o P4 - para inserir uma instrução *MOVL* ou *MOVJ* e o ponto P5 para o início do segundo círculo. Para efectuar os dois arcos de círculo correctamente, precisamos de dar os pontos P1,P2 e P3 com a instrução *MOVC* o ponto P4 com a instrução *MOVJ* ou *MOVL* e os pontos P5,P6 e P7 com a instrução *MOVC*. Na Figura 6.3.6 está descrita a forma de declarar a instrução *MOVC*.

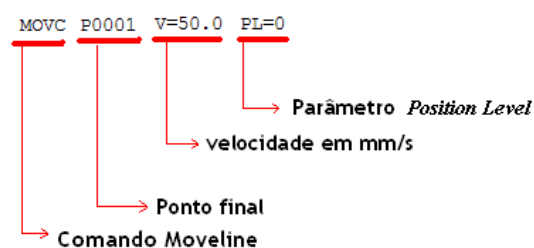


Figura 6.3.6 Instrução *MOVC*

Mais uma vez, nesta instrução, é possível utilizar a opção *position level* (parâmetro *PL*), já descrita na instrução *MOVL*.

O código para a execução de uma circunferência nos pontos P1,P2,P3 à velocidade de 50.0 mm/s e PL=0 tem a seguinte estrutura:

```
MOVC P1 V=50.0 PL=0  
MOVC P2 V=50.0 PL=0  
MOVC P3 V=50.0 PL=0
```

Para ambas as instruções *MOVC* e *MOVL*, a velocidade pode ser definida de várias formas, indicadas a seguir:

- **V** -> Velocidade do centro da ferramenta. Valores de 0.1mm até 1500.0 mm/s.
- **VR** -> Velocidade angular. Valores de 0.1 até 180 graus/s.
- **VE** -> Velocidade dos eixos externos. Valores de 0.01% até 100.00%.

## 6.4 Detecção de zonas de paragem

Como existe ruído na recolha de valores, devido à detecção dos marcadores luminosos, começa-se por efectuar uma filtragem de todos os pontos da trajectória.

É feita uma análise às distâncias entre cada ponto e os seus vizinhos. Esta análise é feita criando uma esfera imaginária em redor do ponto em análise. No caso do ponto seguinte estar dentro desta esfera imaginária ele será encarado como sendo o mesmo ponto. A posição deste novo ponto é actualizada para a posição média dos outros dois pontos. Considere-se um ponto I, representado na Figura 6.4.1 a partir deste ponto, é criada uma esfera de raio D. Se os pontos seguintes da trajectória estiverem dentro desta esfera, ou seja numa distância inferior a D, (representada a vermelho), então esses pontos serão etiquetados como sendo o mesmo ponto I - ver Figura 6.4.2. O próximo ponto em que se situe a uma distância superior a D (representado a cor azul clara), será o próximo ponto distinto da trajectória, sendo atribuída a etiqueta II. É feita, novamente, uma esfera de raio D em torno deste novo ponto II e assim sucessivamente, até ao final da trajectória.

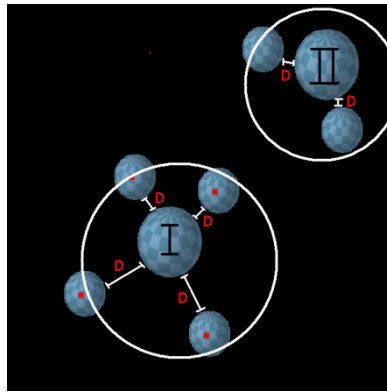


Figura 6.4.1 Esferas em torno dos pontos de análise

Caso exista um número de pontos superior a um dado limite Delta, conclui-se que ocorreu uma paragem da ferramenta, sendo adicionada uma instrução de espera, para que o manipulador fique parado nessa mesma posição. Todos estes parâmetros Delta e distância D podem ser definidos antes de mandar analisar a trajectória.

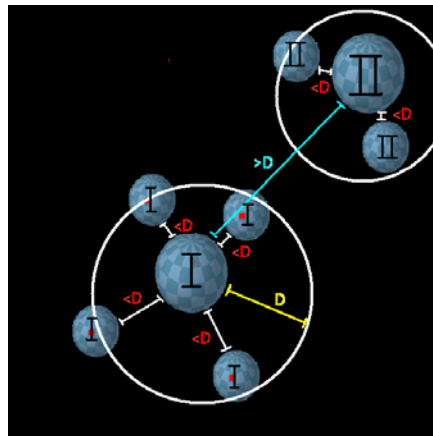


Figura 6.4.2 Filtragem de pontos

## 6.5 Detecção de zonas de velocidade constante

O principal objectivo é criar uma trajectória aproximada por segmentos de recta ou arcos de circunferência. Começando pela aproximação a segmentos de recta e analisando as capacidades do manipulador, sabemos à partida que o manipulador executará a instrução *MOVL* a velocidade constante. Por exemplo: considerando o segmento de recta da Figura 6.5.1, supondo que esta é percorrida por diferentes velocidades, e para que o manipulador consiga executá-la, será necessário dividi-la em rectas de igual velocidade - neste caso, 3 rectas.

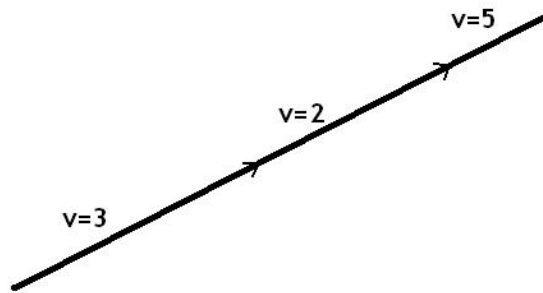


Figura 6.5.1 Trajetória em recta

Embora se trate apenas de uma recta, é necessário dividi-la em 3 segmentos de recta distintos - Figura 6.5.2.

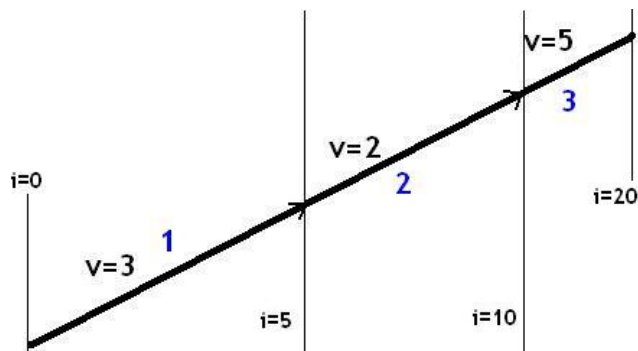


Figura 6.5.2 Diferentes segmentos da mesma recta com velocidades diferentes

Assim tem-se que:

```
Movline i=0 até i=5 com v=3;
Movline i=5 até i=10 com v=2;
Movline i=10 até i=20 com v=5;
```

Chama-se a atenção que o código apresentado, não está na linguagem do manipulador. É apenas uma maneira de expressar o pretendido.

## 6.6 Detecção de rectas

Após a obtenção de um conjunto de pontos num espaço tridimensional, designado de trajectória, pretende-se detectar subconjuntos de pontos que possam ser aproximados por rectas.

Para este efeito, o algoritmo usado é um algoritmo iterativo que percorrerá toda a trajectória, testando se um dado conjunto de pontos pode ser aproximado por uma dada recta definida pelos próprios pontos do conjunto. Considerando uma qualquer sequência de pontos, por exemplo o da Figura 6.6.1, e numerando cada um deles, para fácil identificação por parte do leitor, passa-se a explicar o algoritmo de aproximação de rectas a um determinado conjunto de pontos. Inicialmente, utilizam-se os 3 primeiros pontos da trajectória e, com estes, definir uma recta - ver Figura 6.6.1. Posteriormente, é calculado o erro quadrático médio de cada ponto que define a recta.

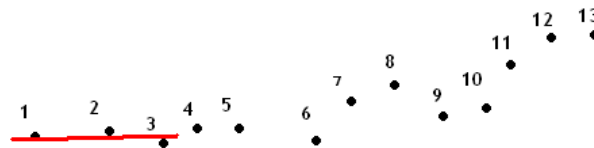


Figura 6.6.1 Recta dos 3 primeiros pontos

Caso este erro quadrático médio seja mais pequeno que um dado valor  $v$ , aumenta-se o número de pontos que passam a definir a nova recta - neste caso, os primeiro 4 pontos da recta e assim sucessivamente, até ao ponto em que a média do erro quadrático seja superior a  $v$ . Desta forma, a recta fica só definida pelos primeiros seis pontos, ver Figura 6.6.2, sendo que a partir do ponto seis, é considerada uma nova recta. No final, é obtida uma trajectória aproximada por pequenos segmentos de recta - ver Figura 6.6.3.

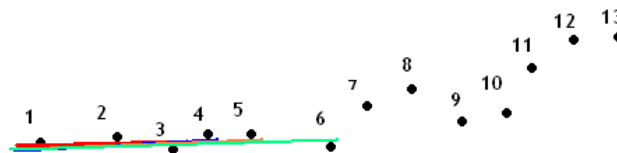


Figura 6.6.2 Rectas calculadas dos 6 primeiros pontos



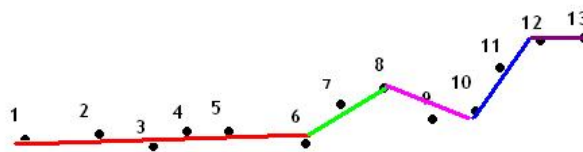


Figura 6.6.3 Trajectória final aproximada por rectas

Juntando todos estes passos, foi implementada uma aplicação que analisa os pontos da ferramenta. A sequência de operações efectuada é:

- i) Fazer uma filtragem do ruído e detecção de paragens;
- ii) Entre paragens realizar o agrupamento dos pontos por velocidades iguais;
- iii) Dentro dos conjuntos de velocidades iguais, analisar os pontos e verificar se pertencem a uma recta ou um arco de circunferência, dependendo de um critério de ajuste.



# Capítulo 7

## Relação Força vs Corrente eléctrica

Para se tentar determinar a relação entre a força, exercida durante a actividade de polimento, em função da corrente eléctrica consumida pelo um motor trifásico da máquina de polimento, foram efectuados *logs* sincronizados destas duas grandezas.

### 7.1 Hardware

A lista de hardware para estes testes é:

- ✓ Manipulador
- ✓ *Gripper*
- ✓ Sensor JR3
- ✓ Analisador da Ducati
- ✓ Placa de monitorização de corrente
- ✓ Peças de polimento
- ✓ Máquina de Polir

O hardware e software usados, foram já descritos no capítulo 4.

#### 7.1.1 Abordagem

Existem dois tipos de peça para teste, se bem que apenas um pode ser processado, devido ao *gripper* apenas conseguir segurar um dos dois tipos de peças. Para a recolha destas amostras foram definidas 3 zonas de teste na peça processada. Esta repartição de diferentes zonas, efectuou-se devido ao facto da força exercida variar consoante a zona a polir.

Divisão da peça representada na Figura 7.1.1.



Figura 7.1.1 Peça número 2

Em cada teste foram retiradas em média 800 amostras de cada valor. Usaram-se duas peças e estas foram escolhidas aleatoriamente. Este número foi escolhido pois em média ao fim de 5 leituras consecutivas a peça tinha que ser retirada da máquina, devido ao sobreaquecimento do disco de polimento e da peça a polir.

Para evitar relações de atrito/ aquecimento, as peças antes de serem usadas no teste, eram limpas, para eliminar detritos originados pelo aquecimento do disco e a troca das peças efectuava-se de 5 em 5 leituras.

O manipulador segura a peça de teste através do *gripper*, avançava segundo o referencial mostrado na Figura 7.1.2. Este referencial está assente na extremidade da ferramenta, neste caso o *gripper*. Efectuou-se um avanço uniforme segundo o eixo X de meio em meio milímetro, numa distância total verdadeira de 7 mm. É definido como distância total verdadeira, a distância avançada com a peça em contacto com o disco. Entre leituras era necessário ir actualizando a posição de contacto com o disco, devido ao desgaste deste.

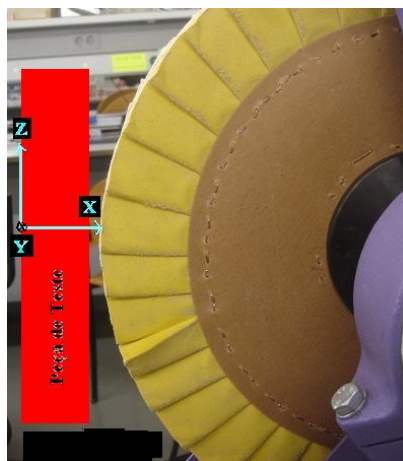


Figura 7.1.2 Referencial para o teste

Durante a actividade de polimento todas as peças envolvidas, principalmente o disco de polimento e a superfície das peças vão sofrendo desgaste. No início de cada teste houve o cuidado de actualizar a posição de contacto com o disco de polimento. Para se ter uma noção de desgaste por parte deste, no final de 23 testes observou-se um desgaste de 7 mm do disco. À primeira vista aparenta ser um desgaste muito pequeno, mas que afecta em muito toda a experiência, considerando que o avanço da peça foi feito com um incremento de meio milímetro na maioria dos testes.

É muito importante actualizar sempre que possível a coordenada do instante de contacto entre a peça e o disco, para obtermos resultado da relação entre a força e a corrente mais fiáveis.

Notou-se que as leituras do sensor JR3 com a instrução *SERVO ON* (que coloca os motores do manipulador em funcionamento), ficavam afectadas de ruído devido às correntes consumidas pelos motores do manipulador. Compreende-se este fenómeno, visto que estamos a trabalhar com um sensor analógico e que o cabo de extensão até à placa PCI não é blindado.

Para reduzir o efeito do ruído, posicionava-se a peça através do manipulador na posição pretendida, e os servos do manipulador eram desligados.

Os testes feitos basearam-se em encostar a peça ao disco e retirar a força e corrente. Entre testes havia um incremento de 0,5mm segundo o eixo X.

Devido a mudanças de *setup* e falta de tempo, apenas se realizaram testes com a peça número 2 e na zona 1 e 2.

Durante a análise dos *logs*, observou-se que o circuito de leitura de corrente estava a saturar, na fase final dos testes – Zonas onde se exercia mais força sobre o disco de polimento e o consumo de corrente era maior. Este fenómeno, deve-se ao facto de uma má projecção do limite máximo mensurável da corrente por parte do circuito implementado – valor eficaz de 7.5 A.

O valor máximo apresentado pelo analisador de corrente da Ducati nestas alturas, era de 6.5A enquanto, o nosso circuito tinha leituras muito próximas de 7.4 A no início do teste e no final do teste, ao fim de 400 medidas o valor da corrente eficaz medido era de 6.5A – valor correcto. Este disparo abrupto de corrente é normal visualizar-se no circuito implementado e não no analisador da Ducati, já que este apenas apresenta valores de 1 em 1 segundo e durante este tempo, efectua filtragens, não sendo possível medir-se disparos/oscilações abruptas de corrente.

Para evitar confusões, chama-se a atenção para o facto de serem usados dois referenciais, o referencial do robô, representado Figura 7.1.2 e o referencial usado no sensor de força, que está representado na secção 4 na Figura 4.6.2. No gráfico em cima é representada a força  $F_z$  do sensor Jr3.

Mesmo assim analisaram-se os dados obtidos para valores mais baixos de corrente e de força, estando estes representados na Figura 7.1.3.

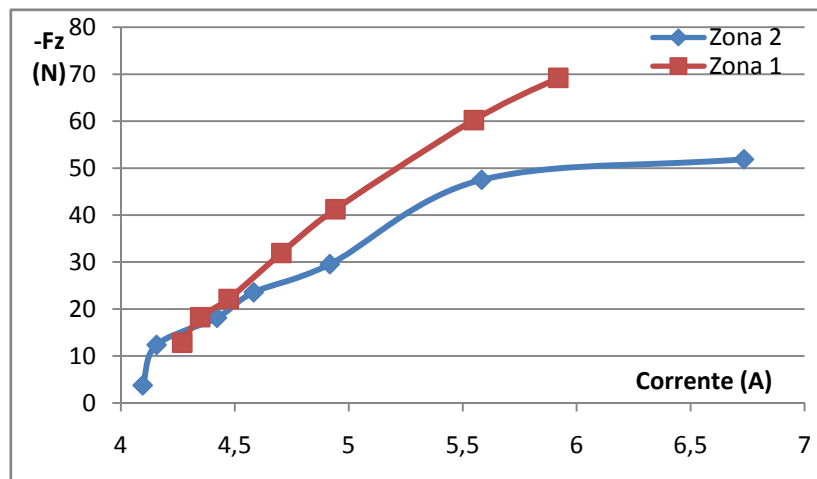


Figura 7.1.3 Gráficos Força/Corrente eléctrica

Das duas curvas tiradas a que se pode ter mais confiança é na curva calculada na zona 1. A curva calculada para a zona 2, foi a que se verificou onde existiu mais saturação do valor da corrente e portanto deve ser ignorada.

Observando a curva da zona 1, aparenta que a relação entre as duas grandezas é linear, conforme era esperado. No entanto deve-se fazer mais ensaios para confirmar este facto.

Para testes futuros, pode-se projectar um novo circuito para medida de valores eficazes da corrente, ou tentar arranjar um produto comercial que consiga cumprir as especificações necessárias - no mínimo funcionar à frequência de 10 Hz, frequência de funcionamento do sistema actual.

# Capítulo 8

## Conclusões

Foram abordados todos os campos previstos nos objectivos principais para o trabalho.

Verificou-se que através do algoritmo de calibração de Tsai é possível extrair os parâmetros do sistema de visão de uma maneira fácil e versátil. O algoritmo *eight point* é mais flexível em termos de calibração mas a forma como está implementado não é possível uma reconstrução 3D métrica directa, sendo necessário introduzir este passo para se poder comparar com o algoritmo Tsai.

O cálculo da posição e rotação da ferramenta está a ser calculado devidamente e é possível o controlo em tempo real do manipulador.

Juntamente com a determinação da orientação da ferramenta é possível efectuar *logs* sincronizados de imagem, força e corrente. A recolha sincronizada destas grandezas, será futuramente muito importante para o relacionamento entre elas no contexto da actividade de polimento.

A aproximação de pontos por segmentos de rectas, apenas metade do algoritmo está implementado - detecção de ruído, e agrupar pontos por velocidade igual e a génese de trajectórias por arcos de circunferência não foi possível ainda a sua implementação devido à falta de tempo.

Este trabalho foi a continuação de um estudo sobre visão estereoscópica. Penso que em redor do sistema Sincrovision se continuará a estudar novas opções e estratégias para tornar o sistema mais intuitivo e robusto.

Neste trabalho tentou-se gerar trajectórias para o manipulador, mas através do uso deste sistema é possível também manusear o manipulador industrial em tempo real. Este modo de operação é importante para certas actividades como por exemplo: manusear substâncias perigosas ou objectos de grande porte que de forma manual era completamente impossível - sendo apenas necessário 4 marcadores, um sistema de

visão e um computador para manobrar com grande facilidade e agilidade um manipulador industrial.

É uma alternativa às soluções de programação de manipuladores oferecidas hoje em dia, tendo sido o que mais me motivou para aceitar este desafio. Os resultados são animadores embora não perfeitos, mas devido ao tempo que se dispunha para a sua elaboração, apenas foi possível fazer o exposto nesta tese.

## 8.1 Trabalho futuro

Para trabalhos futuros sugiro as seguintes ideias para as áreas de:

- a) Caracterização do sistema Sincrovision:
  - i. Análise qualitativa e quantitativa da precisão do sistema Sincrovision. Dependendo do método de calibração e técnicas usadas a precisão do sistema pode ser aumentada.
  - ii. Avaliar pontos de *stress* do sistema. Qual o espaçamento mínimo/máximo/ideal e posicionamento dos marcadores para os quais ainda se obtêm valores correctos. Visto que se pretende colocar os marcadores nas ferramentas dos operadores, para guardar os seus movimentos, esta etapa é muito importante, para se ter uma noção sobre o tipo de actividades a que se pode aplicar. Alguns dados foram já recolhidos, sobre a distância entre marcadores, sendo necessário tempo para os processar e tirar conclusões.
  
- b) Aumento da robustez do sistema:
  - i. Determinação da posição e rotação da ferramenta. De momento a aplicação para o cálculo do estado da ferramenta precisa de estar a visualizar os quatro marcadores. O que seria possível fazê-lo apenas com 3 marcadores, o marcador de referência e 2 amarelos, caso algum deles, saísse fora do alcance do sistema de visão. Um procedimento automático poderia detectar em caso de apenas existirem 3 marcadores visíveis, efectuar o cálculo do plano e *centroid* ser efectuada através dos 3 pontos e assim determinar o estado da ferramenta.
  - ii. Aplicar algoritmos preditivos baseados no estado anterior da ferramenta para o cálculo do estado seguinte caso o marcador vermelho saísse fora do alcance do sistema de visão.



## c) Geração de trajectórias:

- i. Aperfeiçoar os algoritmos já existentes para a geração de trajectórias para o manipulador e/ou até mesmo implementar outros.
- ii. O uso da instrução *MOVS* que faz mover o manipulador entre 3 pontos segundo uma *Spline*.

## d) Actividade de polimento:

- i. Analisar dados retirados sobre a relação Força vs corrente e tirar a sua relação.
- ii. Efectuar vídeos de uma sessão de polimento, e tentar adaptar as trajectórias geradas a esta actividade.



## Referências

- [1] Maeda,Y, Ushioda,T and Makita S. (2008), Yokohama National University “*Easy Robot Programming for Industrial Manipulators by Manual Volume Sweeping*” in 2008 IEEE International Conference on Robotics and Automation Pasadena, CA, USA, May 19-23.
- [2] Aleotti J, Caselli S, (2006) (Parma University, Italy) “*Robust Trajectory learning and approximation for robot programming by demonstration.*” in Robotics and Autonomous Systems.
- [3] Ekvall,S. Kragic,D. (2005): “*Grasp Recognition for Programming by Demonstration*”; in: Robotics and Automation, 2005. ICRA.
- [4] Dillmann R. (2004), “*Teaching and learning of robot tasks via observation of human performance,*” in Robotics and Autonomous Systems, vol. 47, no.2-3, pp. 109-116.
- [5] Ehrenmann,M, Rogalla,O, Zöllner,R and Dillmann, R “*Teaching Service Robots Comple Tasks: Programming by Demonstration for workshop and household environments*”,University of Karlsruhe (TH), Germany
- [6] Lopes,M and Bernardino,A and Santos-Victor,J (2005) “*A Developmental Roadmap for Task Learning by Imitation in Humanoid Robots. AISB*” - Third International Symposium on Imitation in Animals and Artefacts. Hatfield, England.
- [7] Nagata, F, Watanabe, K, Kiguchi, K, Tsuda, K, Kawaguchi, S, Noda, Y, and Komino,M, (2001) “*Joystick teaching system for polishing robots using fuzzy compliance control,*” Proceedings 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation, pp. 362-367.
- [8] Jayawardena, C. Watanabe, K, Izumi, K, Dept. of Adv. Syst. Control Eng., Saga Univ., Japan (2005), “*Teaching a tele-robot using natural language commands in Micro-NanoMechatronics and Human Science*”, IEEE International Symposium on.
- [9] Pires, J.N.(2007): “*The Industrial robot as a human coworker: the role of the speech interfaces*”. In Proceedings of the International Conference on Software Development for Enhancing Accessibility and Fighting Info-exclusion Portugal.

- [10] Pires, J.N. (2007), “*Demonstration videos: programming robots with digital pens*”, Industrial Robotics Laboratory, Mechanical Engineering Department, Coimbra, available at: <http://robotics.dem.uc.pt/robot-pen>
- [11] Neto, P, Pires, JN, Moreira, AP, (2009) "*High-level programming and control for industrial robotics: using a hand-held accelerometer-based input device for gesture and posture recognition*", Emerald Industrial Robot, 2009 (accepted - April 2009). Internal Code: IR-09-339
- [12] Aleotti, J, Skoglund, A, and Duckett, T, (2004), “*Position teaching of a robot arm by demonstration with a wearable input device*” International Conference on Intelligent Manipulation and Grasping (IMG04), Genoa, Italy, July 1-2.
- [13] Hartley, R, and Zisserman, A, (2003). “*Multiple View Geometry in computer vision*”. Cambridge University Press.
- [14] TSAI, R, (1987) – “*A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses*”, in IEEE JOURNAL OF ROBOTICS AND AUTOMATION, VOL.RA-3, NO.4.
- [15] Raposo, AB, Szenberg, F, Gattass, M, Celes, W : “*Visão Estereoscópica, Realidade Virtual, Realidade Aumentada e Colaboração*”.
- [16] Elias, R, and Laganière, R, - “*Projective Geometry for Three-Dimensional Computer Vision*”.
- [17] Hartley,R, (1997). "*In Defense of the Eight-Point Algorithm*". IEEE Transaction on Pattern Recognition and Machine Intelligence.
- [18] Hartley, R and Sturm, P – “*Triangulation*”.
- [19] Eberly,D (2008), “*Least Squares Fitting of Data*”, in: Geometric Tools, LLC
- [20] MOTOMAN NX100 “*Basic Programming*”.
- [21] Foley, J (1990) : “*Computer Graphics*”.
- [22] Qian, N.(1997), “*Binocular Disparity and the Perception of Depth*”, Neuron, 18, 359-368.
- [23] Xu,G and Zhang, Z (1996). “*Epipolar geometry in Stereo, Motion and Object Recognition*”. Kluwer Academic Publishers.
- [24] <http://www.robot.uji.es/lab/plone/Members/mprats>, Maio 2009.
- [25] <http://robotics.dem.uc.pt/norberto/> , Maio 2009.
- [26] Mark W. Spong, Hutchinson, S, Vidyasagar, M. (2005) “*Robot Modeling and Control*” .

# Anexo A

## Conceitos

**Epipolos ou pontos epipolares:** ponto de intersecção da linha que une os centros das câmaras com cada plano de imagem. [13]

**Plano epipolar:** Plano que contem a linha que une os epipolos. [13]

**Linha epipolar:** Linha resultante da intersecção do plano epipolar com os planos de imagem. Passa entre os pontos projectados no plano de imagem e nos epipolos (ex:  $e_R - x_R$ , exemplo de linha epipolar da imagem do lado direito). [13]

**Disparidade binocular:** Refere-se à diferença existente na localização de um objecto numa imagem vista no lado esquerdo ou direito do sistema de captura (olho esquerdo ou olho direito, mas neste caso denominado disparidade da retina) devido à distância entre as duas fontes de captura da mesma cena.

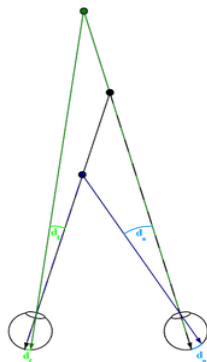


Figura 8.1.1 Disparidade binocular

Com uma disposição de câmaras parecida à disposição dos olhos humanos, a disparidade apenas aparece na direcção horizontal, e não existe disparidade na

coordenada  $y$  da imagem. Usualmente a disparidade é calculada pelo deslocamento para a esquerda de uma cena vista numa imagem observada do lado direito. Por exemplo, um ponto  $K$  no mundo aparece na coordenada  $x$  (medida em pixéis) na imagem do lado esquerdo; O mesmo ponto  $K$  na imagem do lado direito tem a coordenada  $x - 3$  nesta situação a disparidade é de 3 pixéis. No caso de a disparidade ser 0, os pontos nas duas imagens aparecem exactamente no mesmo sítio na imagem do lado direito.

**Percepção de profundidade** – capacidade visual para percepção tridimensional do Mundo, até agora o conceito de percepção é reservado aos humanos, embora qualquer animal consiga mover-se no seu ambiente envolvente, apenas nós humanos conseguimos transmitir a noção de distância uns aos outros.

**Homografia** - Conceito matemático em que consiste numa transformação de um plano de projecção para outro plano de projecção, também conhecido como transformação projectiva.

**Geometria epipolar** – é a geometria projectiva intrínseca entre dois pontos de vista distintos. Uso de duas câmaras para visualizar um cenário a 3D a partir de duas posições distintas. Este tipo de geometria é independente da cena visualizada apenas dependendo dos parâmetros internos e da posição relativa das câmaras usadas. Todas estas relações são obtidas com base no pressuposto de que as câmaras podem ser aproximadas pelo modelo de *pinhole camera*. Esta geometria é normalmente motivada pela procura de pontos de correspondência nas duas imagens. [13]

**Matriz anti-simétrica** - Uma matriz anti-simétrica é aquela com a qual sua matriz transposta coincide com sua matriz oposta:

$$A^t = -A$$