



Miguel de Almeida

Reconstrução 3D e calibração de lasers no AtlasCar



Miguel de Almeida

Reconstrução 3D e calibração de lasers no AtlasCar

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Dr. Paulo Miguel de Jesus Dias, Professor auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e do Prof. Dr. Vítor Manuel Ferreira dos Santos, Professor associado do Departamento de Engenharia Mecânica.

o júri

presidente

Prof. Dra. Maria Beatriz Alves de Sousa Santos

Professora associada com agregação do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

Dr. António Fernando Cunha Castro Coelho

Professor auxiliar do Departamento de Engenharia Informática da Faculdade de Engenharia da Universidade do Porto

Prof. Dr. Paulo Miguel de Jesus Dias

Professor auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

Prof. Dr. Vítor Manuel Ferreira dos Santos

Professor associado do Departamento de Engenharia Mecânica da Universidade de Aveiro

agradecimentos

Desde já, quero agradecer ao Prof. Dr. Paulo Miguel de Jesus Dias na sua qualidade de orientador, cujo conhecimento na área da computação visual, competência científica e dedicação foram de extrema importância para a concretização desta tese.

Agradeço ao Prof. Dr. Vítor Manuel Ferreira dos Santos pela integração no projecto Atlas, sugestões e críticas ao trabalho que tenho desenvolvido, nomeadamente na área da computação visual.

Agradeço ao Miguel Oliveira, que pela sua disponibilidade e conhecimento na área da computação visual, geometria, álgebra e pela formação dada sobre as plataformas de programação do AtlasCar. A toda a equipa do LAR, que sempre se demonstrou prestável e dedicada.

Aos meus pais que sempre estiveram presentes e deram força e motivação para continuar e alcançar os objectivos.

Agradeço à minha namorada a paciência tida ao longo da evolução da tese, pelos momentos de *stress* que fiz passar mas que sempre esteve presente.

Obrigado a todos os meus colegas que deram sugestões e um pouco do seu conhecimento para a concretização da tese e a toda a minha família.

palavras-chave

Reconstrução 3D, Calibração, Triangulação, Laser 3D, Laser 2D, Textura.

resumo

O presente trabalho propõe a realização de reconstrução 3D a partir de lasers e câmaras do AtlasCar e a calibração dos lasers 2D relativamente ao laser 3D. A tese é composta por um estudo sobre a área da reconstrução 3D, algoritmos de triangulação e aplicação de textura. Foram reconstruídos modelos com textura obtida através de uma imagem digital de uma das câmaras do AtlasCar. Na área da calibração foi realizado um estudo sobre calibração de lasers 2D usando um laser 3D (métodos, algoritmos e procedimentos para a execução do processo). A calibração está dividida em validação teórica e aplicação em dados reais. A nível teórico o processo de calibração proposto satisfaz o objectivo. A nível prático, o ruído introduzido pelos lasers do AtlasCar influenciaram a precisão do resultado, resultando no entanto uma melhoria visual em comparação com a calibração já usada no AtlasCar, com algumas limitações e constitui um primeiro passo na calibração integrada de lasers e câmaras do veículo.

Índice

Índice de Tabelas.....	17
I. Introdução	19
I.1. Objectivos	20
I.2. Estado da arte	21
I.2.1. Reconstrução 3D	22
I.2.2. Calibração.....	26
I.3. Estrutura do documento	27
II. Descrição das plataformas.....	29
II.1. Lasers e câmaras.....	29
II.2. Comunicação entre módulos e <i>logging</i>	32
II.3. Plataforma de programação.....	33
II.4. Visualização do modelo	35
II.5. Ferramentas de interacção	36
III. Reconstrução 3D	39
III.1. Reconstrução de malhas 3D	39
III.2. Algoritmos de triangulação em estudo.....	40
III.3. Algoritmo Ball Pivoting	43
III.4. Integração com o VTK.....	45
III.5. Resultados	46
III.5.1. Nuvens de pontos pouco densas.....	47
III.5.2. Nuvens de pontos densa	50
III.5.3. Análise de desempenho	52
III.6. Conclusões	53
III.7. Aplicação de textura.....	53
III.8. Matrizes de Transformação de coordenadas	54
III.9. Resultados	56

III.10. Conclusões	57
IV. Calibração de lasers no AtlasCar	59
IV.1. Diagrama funcional da solução	62
IV.2. Objecto de calibração	64
IV.3. Simulador de nuvens de pontos.....	65
IV.3.1. Parâmetros do Simulador	66
IV.3.2. Aquisição de dados com um laser 2D	69
IV.3.3. Aquisição de dados com o laser 3D	70
IV.3.4. Resultados	71
IV.4. Processo de calibração do laser 2D	72
IV.4.1. Detecção do objecto de calibração no modelo 3D	73
IV.4.2. Processamento dos dados 2D	76
IV.4.3. Resultados	84
IV.4.4. Conclusões	89
IV.5. Aplicação em dados reais	90
IV.5.1. Objecto de calibração	90
IV.5.2. Aquisições de dados laser do AtlasCar	91
IV.5.3. Resultados	95
IV.5.4. Conclusões	99
IV.6. Conclusões	99
V. Conclusões e trabalho futuro.....	101
V.1. Conclusões	101
V.2. Trabalho futuro.....	102
A. Apêndice A – Alterações na Makefile	105
VI. Referências.....	107

Índice de Figuras

Fig. I.1- Sensores e câmaras presentes no AtlasCar	19
Fig. I.2 - Triangulação de pontos 2D	21
Fig. I.3 - Reconhecimento do movimento e sua reconstrução	23
Fig. I.4 - Reconhecimento através da câmara <i>Pan-tilt</i>	24
Fig. I.5 - Laser 3D montado na estrutura rotativa	25
Fig. II.1 - Laser Planar 2D SICK	30
Fig. II.2 - Laser Planar 2D Hokuyo.....	30
Fig. II.3 - Laser 3D adaptado	30
Fig. II.4 - Câmara estéreo.....	31
Fig. II.5 - Câmara Pan-tilt	31
Fig. II.6 - <i>Player</i> utilizado para reproduzir os dados dos lasers e câmaras armazenados em Log. ..	33
Fig. II.7 - Visualização dos dados do AtlasCar da oficina do DEM.	36
Fig. III.1 - Ilustração da reconstrução de Poisson em 2D retirado de (Kazhdan et al., 2006).....	41
Fig. III.2 - União de dois discos T_p e T_q do algoritmo Layer Peeling.....	42
Fig. III.3 - Algoritmo Ball Pivoting em 2D.....	44
Fig. III.4 - Funcionamento do algoritmo Ball Pivoting.....	44
Fig. III.5 - BPA em presença de dados com ruído	45
Fig. III.6 - Imagem da Igreja de Vera Cruz.....	47
Fig. III.7- Triangulação do modelo da Igreja de Vera Cruz com nuvens de pontos pouco densas. .	47
Fig. III.8 - Imagem da Capela da N ^a Sra. das Febres	48
Fig. III.9 - Triangulação do modelo da Igreja N ^a Sra das Febres com nuvens de pontos pouco densas.	48
Fig. III.10 - Imagem da Oficina do Departamento de Engenharia Mecânica	49
Fig. III.11- Triangulação do modelo do departamento do DEM com nuvens de pontos pouco densas.	49
Fig. III.12 - Pormenores da oficina do DEM para o exterior	50
Fig. III.13 - Triangulação do modelo da Capela da N ^a Sra. das Febres com nuvens de pontos densas.	51
Fig. III.14 - Triangulação do modelo da Igreja de Vera Cruz com nuvens de pontos densas.....	51
Fig. III.15 - Triangulação do modelo da oficina do DEM com nuvens de pontos densas.	51
Fig. III.16 - Sistema de coordenadas de referência do AtlasCar.	54

Fig. III.17 - Aplicação de Textura ao modelo da Igreja de Vera Cruz.....	56
Fig. III.18 - Aplicação de Textura ao modelo da Capela da N ^a Sra. das Febres.	56
Fig. III.19 - Aplicação de Textura ao modelo da oficina do DEM.	57
Fig. IV.1 - Aquisição de dois lasers 2D não calibrados	59
Fig. IV.2 - Calibração de dois lasers planares.....	60
Fig. IV.3 - Diagrama de calibração	62
Fig. IV.4 - Diagrama funcional da aplicação	62
Fig. IV.5 - Nuvem de pontos resultante do modelo inicial	63
Fig. IV.6 - Objecto de calibração virtual.....	64
Fig. IV.7 - Casos de estudo da intercepção de um plano no cone.....	65
Fig. IV.8 - Processo de aquisição 2D de pontos no simulador.....	66
Fig. IV.9 - Parâmetros dos lasers	67
Fig. IV.10 - Exemplo de um ficheiro de configuração do objecto de calibração.....	68
Fig. IV.11 - Parâmetros de aquisição do laser 2D.....	69
Fig. IV.12 - Dados de aquisição do laser 3D simulado	71
Fig. IV.13 - Capturas simuladas em diversos ambientes com o objecto de calibração.....	72
Fig. IV.14 - Detecção do objecto de calibração	73
Fig. IV.15 - Dados do objecto de calibração filtrados.....	74
Fig. IV.16 - Resultado do fitting do objecto de calibração	76
Fig. IV.17 - Sequência de selecção de pontos das elipses.....	77
Fig. IV.18 - Esquema das elipses obtidas no laser 2D	78
Fig. IV.19 - Obtenção dos pontos das extremidades dos eixos da elipse	79
Fig. IV.20 - Corte do cone segundo o eixo maior	80
Fig. IV.21 - Elipse projectada em 2D.....	80
Fig. IV.22 - Encaixe da elipse ao cone no espaço tridimensional.....	82
Fig. IV.23 - Vista da posição do objecto de calibração na cena.....	85
Fig. IV.24 - Calibração usando um objecto de calibração	85
Fig. IV.25 - Aquisição com dois objectos de calibração.....	86
Fig. IV.26 - Calibração usando dois objectos de calibração	87
Fig. IV.27 - Aquisição usando três objectos de calibração	88
Fig. IV.28 - Calibração com três objectos de calibração.....	88
Fig. IV.29 - Protótipo do objecto de calibração.	90
Fig. IV.30 - Local das aquisições feitas com o AtlasCar com o objecto de calibração.....	91
Fig. IV.31 - Comparação da aplicação do ICP a diferentes densidades de pontos	92

Fig. IV.32 - Dados do laser 2D e 3D do AtlasCar.....	93
Fig. IV.33 - Sobreposição de dados 3D com vários objectos de calibração	94
Fig. IV.34 - Resultado do ICP em dados reais.	95
Fig. IV.35 - Reconstrução de elipses em dados laser 2D reais.....	96
Fig. IV.36 - Resultado da acumulação de duas aquisições do laser 2D	96
Fig. IV.37 - Resultado da calibração em dados reais usando três objectos de calibração.....	97
Fig. IV.38 - Resultado da calibração em dados reais usando quatro objectos de calibração.	98
Fig. V.1 - Definição do caminho da biblioteca VTK	105
Fig. V.2 - Inclusão das bibliotecas necessárias à compilação	105
Fig. V.3 - Definição dos ficheiro binário para <i>linkagem</i>	106

Índice de Tabelas

Tabela III.1- Tempos de processamento para 1 mensagem (~15 000pontos).....	52
Tabela III.2 - Tempos de processamento para 5 mensagens (~75 000 pontos).....	52
Tabela IV.1 - Valores predefinidos dos parâmetros dos lasers	68
Tabela IV.2 - Parâmetros fixos do objecto de calibração.....	84
Tabela IV.3 - Parâmetros do laser 2D usando 1 objecto de calibração	84
Tabela IV.4 - Parâmetros do objecto de calibração.....	84
Tabela IV.5 - Parâmetros do laser 2D usando dois objectos de calibração.....	86
Tabela IV.6 - Parâmetros dos dois objectos de calibração utilizados	86
Tabela IV.7 - Parâmetros do laser 2D usando três objectos de calibração.....	87
Tabela IV.8 - Parâmetros dos três objectos de calibração utilizados	87
Tabela IV.9 - Distância média em função do número de objectos de calibração.....	89
Tabela IV.10 - Dados do objecto de calibração real	90

I. Introdução

Este trabalho insere-se no projecto Atlas, desenvolvido pelo Grupo de Automação e Robótica do Departamento de Engenharia Mecânica (DEM) da Universidade de Aveiro. O principal objectivo deste projecto é contribuir no avanço no campo de aquisição de dados sensorial avançados e sistemas activos para implementação na área automóvel e plataformas móveis. Este grupo possui uma vasta experiência em navegação autónoma em ambientes controlados e evoluiu para cenários reais em estrada. Para fazer face ao desafio foi desenvolvido um protótipo em tamanho real, o AtlasCar (Fig. I.1).

Este veículo está equipado com diversos sensores, lasers e câmaras. No entanto apenas serão usados os seguintes no âmbito deste trabalho:

- Laser 3D;
- Câmaras *point grey* com unidade pan-tilt;
- Câmara estéreo;
- Lasers planares.

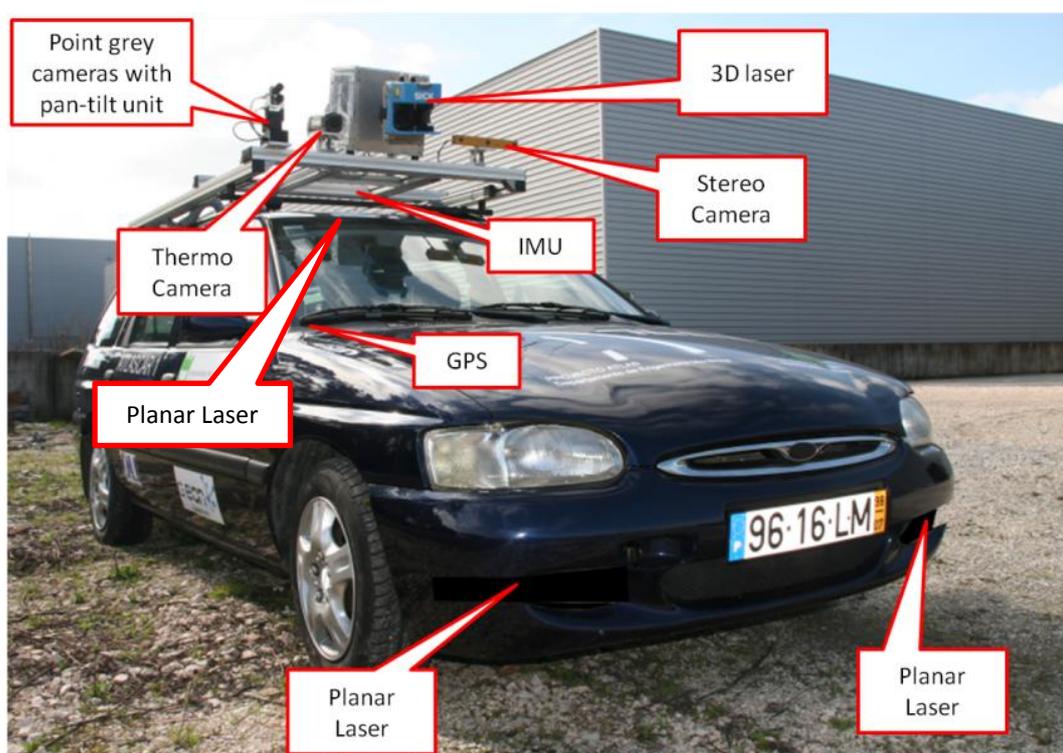


Fig. I.1- Sensores e câmaras presentes no AtlasCar

Para o suporte de toda a estrutura montada no veículo, foi necessário efectuar intervenções eléctricas e mecânicas para suporte ao consumo de energia por parte dos sensores, dos

computadores montados a bordo de suporte e processamento de dados e dos dispositivos de regulação e armazenamento de energia, descrito em detalhe em (Robótica, 2010).

Os sensores descritos neste documento permitem a aquisição de dados 3D, imagens e vídeo, que incide essencialmente no campo de visão do condutor, sendo que alguns lasers cujo alcance incide sobre zonas laterais e posteriores do veículo.

Os dados recolhidos são enviados pelos diferentes módulos para a unidade central que regista o instante relativo à mensagem e os dados associados. Tendo o instante em que foi enviada a mensagem e os dados relativos à mesma, é possível reproduzir a missão efectuada pelo veículo, explicado em detalhe na secção II.2.

I.1. Objectivos

Um dos objectivos deste trabalho é permitir a integração dos diferentes dados recolhidos dos lasers, sensores e/ou câmaras do AtlasCar para permitir a reconstrução de modelos 3D com textura do ambiente do campo de visão do condutor. Estes modelos poderão ser usados para demonstração, para verificação das calibrações e até para auxílio à condução.

No entanto, a integração de dados de diversos sensores numa só plataforma sem qualquer pré-processamento produz modelos desalinhados pois cada sensor possui o seu próprio referencial e não segundo um referencial comum entre todos eles.

Para resolver este problema é necessário proceder a uma calibração. A abordagem nesta tese consiste em usar o laser 3D (mais rico em informação) como referência e alinhar os outros lasers aplicando uma transformação nos dados para que estes coincidam com o sistema de coordenadas do laser de referência.

Devido à crescente evolução do projecto AtlasCar com a inclusão de mais sensores, algoritmos de processamento de dados, etc., todos estes dados terão que ser suportados por uma plataforma que permita a interacção com os dados de forma simples, integrando estruturas e algoritmos de forma modular.

I.2. Estado da arte

Por definição, a reconstrução 3D é a criação de modelos 3D (foto-realistas ou não) a partir de dados sensoriais. Nesta área, um vasto trabalho tem vindo a ser desenvolvido com a finalidade de obter modelos mais realistas e na interpretação do modelo 3D construído ao nível da identificação de padrões.

Para obter modelos 3D a partir dos dados sensoriais, é necessário ter em conta diversos factores, começando pelo método de aquisição dos dados 3D, de seguida o seu processamento para criação de superfícies e por fim a aplicação de textura a partir de imagens digitais.

A obtenção de dados tem origem em sensores, que fornecem dados não tratados (geralmente nuvens de pontos). Para a criação de uma malha poligonal é necessário realizar um processamento aos dados fornecidos pelos sensores (triangulação). A triangulação consiste em criar uma malha (geralmente de triângulos) que une triplos de pontos em que o conjunto final de triângulos corresponde a uma superfície geometricamente equivalente ao modelo real. Está ilustrado na Fig. I.2 um exemplo de uma triangulação em 2D. No final obtém-se um conjunto de superfícies que caracteriza o ambiente capturado pelo laser.



Fig. I.2 - Triangulação de pontos 2D

(a) – Conjunto de pontos fornecidos por um laser. (b) – Exemplo de uma triangulação em 2D dos pontos obtidos por um laser

Para concluir o processo de reconstrução é aplicada textura ao modelo triangulado a partir de imagens digitais, de acordo com um mapeamento de cada *pixel*/zona da imagem a um dado triângulo (descrito em detalhe na secção III.7). Esse mapeamento exige também uma calibração da câmara que fornece as imagens para aplicar na textura. Existem diversos trabalhos realizados na área da reconstrução 3D, descritos na secção I.2.1.

Para a aplicação de textura é necessário que a câmara esteja calibrada. Este é um problema estudado e que foi resolvido recorrendo a um plano com padrão de xadrez (Zhengyou, 1999) (Zhang, 2000). O padrão tem de ser captado pela câmara em pelo menos duas posições diferentes

para calcular a calibração. Essa calibração é obtida sobre a forma de matriz extrínseca, descrito em mais detalhe na secção III.8.

Existem métodos fornecidos por bibliotecas de programação que realizam a calibração de forma automática das câmaras (Bradski and Kaehler, 2008) (Bouguet, 2010), pelo que serão tomadas em conta apenas as calibrações de lasers.

Na calibração de lasers existem diversas formas de o fazer, recorrendo a outros sensores, lasers, câmaras, reconhecimento de padrões, etc. Qualquer que seja o método, o objectivo é permitir o alinhamento correcto dos diversos lasers de forma simples e com o maior grau de precisão possível. A questão da calibração apenas surge quando é necessária a fusão de dados de diversos lasers num só modelo. São mencionadas na secção I.2.2 diversos trabalhos realizados nesta área.

I.2.1. Reconstrução 3D

A reconstrução 3D de ambientes e objectos é alvo de estudo em diversas áreas, como por exemplo os videojogos, arquitectura, engenharia mecânica, realidade virtual, biomedicina e condução autónoma.

Os dados para a reconstrução 3D podem ser obtidos através de sensores activos: projectam a informação necessária para obter os dados (infravermelhos, ultra-sons, etc.); ou passivos: fazem uso da luz emitida pelos objectos.

Os sensores activos são mais usados em medições precisas e em soluções industriais, sendo no entanto dispendiosos dependendo do tipo de dispositivo e resolução dos mesmos. Os sensores baseados em métodos passivos (e.g. fotometria ou visão computacional) são hoje em dia, com o aumento da tecnologia, cada vez mais usados na aquisição de dados geométricos. Estes sensores não emitem energia para obtenção de dados mas capturam a energia emitida pelos objectos, fornecendo medições 3D em imagens em 2D, podendo combinar ou não as diversas imagens.

Quanto aos métodos de reconstrução 3D, podem de um modo geral ser divididos de acordo com o apresentado nas duas secções seguintes.

I.2.1.1. Câmaras 2D

A reconstrução 3D pode ser efectuada exclusivamente através da análise de imagens 2D a partir da qual é retirada informação para recriar o modelo 3D. Estas imagens 2D são obtidas através de sensores passivos. Devido ao baixo custo de alguns destes sensores (como o caso de câmaras de

vídeo), o seu uso é mais generalizado até em projectos pessoais. No entanto, grande parte destes sensores não possui ferramentas de modelação 3D, obrigando ao posterior processamento da imagem. Existe também a problemática da profundidade dos objectos que não é obtida através de uma imagem 2D, obrigando a combinação de imagens ou recorrendo a uma base de dados com diversos objectos 3D e através da imagem 2D, realizar a correspondência ao objecto. Além disso é necessário ter em conta questões de iluminação e ângulo de visualização para identificação do objecto. Todos estes aspectos devem ser tidos em conta na modelação 3D. Estão apresentados nos parágrafos seguintes alguns trabalhos na área de reconhecimento e reconstrução 3D usando sensores passivos.

Um exemplo usando dados vídeo de uma câmara é apresentado em (Howe et al., 1999). O objectivo é capturar movimento 3D de humanos através de vídeo, proporcionando aplicações em “*grafismo computacional industrial, realidade virtual e interacção humano-computador melhorada*”. É usada apenas uma câmara para obtenção de vídeo e o sistema efectua um reconhecimento de zonas do corpo (braços, pernas e tronco) em 2D, combinando dados anteriores para melhor estimar o movimento do corpo em 3D. Alguns *frames* estão exemplificados na Fig. I.3.

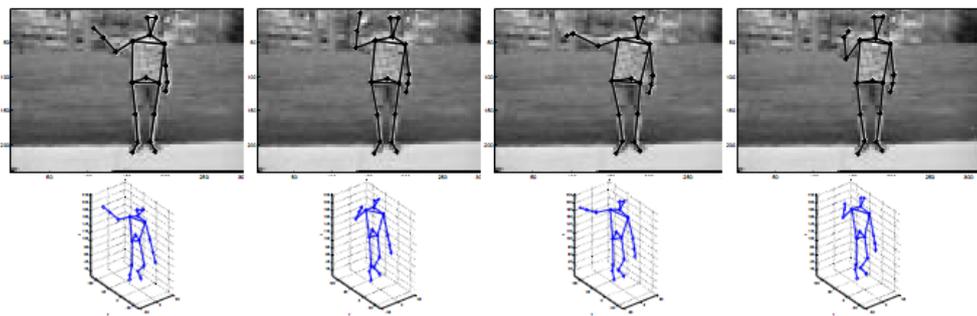
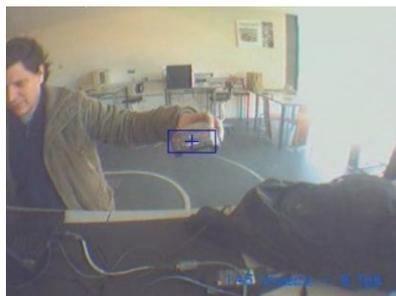


Fig. I.3 - Reconhecimento do movimento e sua reconstrução

Este método não faz uso de marcadores no corpo para identificar os diversos membros, o que o torna possível de utilizar recorrendo às câmaras instaladas no AtlasCar. Este método permite o estudo na área de reconhecimento e do movimento humano, uma problemática já em estudo pela equipa do AtlasCar, apresentado em (Santos et al., 2010), onde é realizado uma análise de imagens digitais para reconhecimento de objectos ou obstáculos na condução autónoma. As câmaras *Pan-tilt* providenciam essa capacidade de percepção activa, virando a sua atenção para um objecto em particular. Esse evento é gerado normalmente pela câmara periférica que possui um ângulo de visão mais abrangente (Fig. I.4).



(a)



(b)

Fig. 1.4 - Reconhecimento através da câmara *Pan-tilt*

(a) Cena visualizada pela câmara periférica e (b) pela câmara *foveada*. Imagem retirada de (Santos et al., 2010).

No entanto, não é realizada uma reconstrução 3D como proposto por *Howe*. A integração de reconstrução 3D para reconhecimento e estudo de movimento no AtlasCar é um trabalho futuro nesta tese, que já fornece uma plataforma sobre a qual podem ser desenvolvidos algoritmos, módulos de reconhecimento e reconstrução 3D.

I.2.1.2. Dados a partir de lasers

Os lasers são sensores activos. Estes podem de forma geral ser divididos em dois tipos:

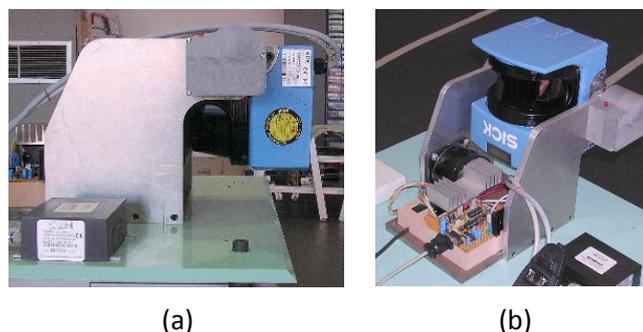
- Baseados em triangulações: tais como sistemas de luz estruturada. Este método tem por base a projecção uma série de linhas ou malhas de luz (laser) que ao incidirem numa estrutura 3D aparece distorcida de outras perspectivas do que a do projector. Analisando a distorção dos feixes de luz é possível extrair a geometria da superfície. Surgem problemas na precisão deste método que dependem da superfície em termos de reflectividade ou transparência. A reflexão da luz faz com que esta seja reflectida para fora ou directamente na objectiva da câmara que captura os feixes de luz do projector. As reflexões duplas podem causar sobreposições de padrões de luz, pondo em causa a qualidade dos resultados. Este tipo de sensores são adequados para curtas distâncias (tipicamente 3 ou 4m).
- *Laser Range Finders (LRF)*: este tipo de sensores emite um impulso laser para determinar a distância a um objecto. A forma mais comum de operação destes sensores baseia-se no tempo que um impulso de luz demora desde o seu envio, reflexão no objecto até ao retorno ao emissor. Com o conhecimento da velocidade da luz e uma medição correcta do tempo a distância é facilmente calculada. Um outro tipo de operação destes sensores é baseado na diferença de fase entre o sinal emitido e o recebido. Este tipo de sensores permitem alcances superiores que dependem da potência do feixe de luz emitida, que podem atingir vários

quilómetros. Devido a isto, os LRF são o tipo de sensores mais usados para aquisições no mundo real.

Alguns destes sensores fornecem ferramentas de modelação 3D ou ferramentas de cálculo de áreas ou volumes mas outros apenas fornecem uma nuvem de pontos tridimensional do ambiente capturado. Existem alguns estudos sobre a reconstrução de ambientes 3D combinando dados de LRF com dados sensores passivos como imagens digitais ou imagens de intensidade, apresentados nos parágrafos seguintes.

Um método é proposto em (Sequeira and Goncalves, 2002) de um sistema de reconstrução 3D usando lasers e imagens digitais para criação de modelos tridimensionais realistas. É usado um LRF para aquisição de dados 3D e uma câmara para aquisição de uma imagem digital. O modelo é analisado para detecção de oclusões (zonas não abrangidas pela câmara). A melhor posição para colocar a câmara e o laser para correcção do problema da oclusão é calculada e estes são movidos para essa nova posição. É efectuada uma segunda captura dos dois sensores. Os dados são fundidos com as aquisições anteriores e se forem detectadas novas oclusões ou oclusões ainda por resolver, são solicitadas novas aquisições. Quando o processo é terminado, o modelo final é triangulado e aplicada a textura a partir dos dados da câmara.

Um primeiro estudo realizado com um protótipo do laser 3D usado no AtlasCar está descrito em (Dias et al., 2006). Este protótipo é diferente do que está actualmente instalado no veículo na forma de aquisição dos dados 3D. É proposto uma reconstrução 3D de um ambiente real usando um LRF 2D combinando com imagens digitais. A filosofia deste tipo de adaptação é na aquisição 3D de ambientes reais através de um LRF mais económico. A estrutura usa um LRF comercial SICK LMS 200 montada sobre uma estrutura rotativa (Fig. I.5).



**Fig. I.5 - Laser 3D montado na estrutura rotativa
(a) Laser apontando para baixo e (b) apontando para a retaguarda.**

O resultado obtido resulta num conjunto de pontos em 3D. É realizada a triangulação recorrendo ao algoritmo 2D de Delaunay com pré-processamento na detecção de arestas e discontinuidades para

obter melhores resultados. A textura aplicada ao modelo pode ser obtida através de diversas imagens digitais em que o processo de calibração das imagens ao modelo é através da interação do utilizador. O resultado final é um modelo 3D reconstruído com diversas imagens digitais que cobrem toda a área abrangida pelo laser.

Todos os métodos apresentados fazem uso de combinação de LRF com aplicação de textura. O último método apresentado é o mais próximo daquele estudado nesta tese, cujas fases para atingir o modelo final em tudo se assemelha a este, variando na metodologia adoptada.

I.2.2. Calibração

Uma outra questão nos sensores que devolvem dados espaciais é a questão da sua calibração. Este ponto é relevante quando são agrupados diversos sensores e os dados combinados para análise ou processamento. Para calibrar um sensor existem diversas formas de o fazer, podendo combinar dois ou mais sensores. A calibração destes sensores pode ser decomposta em parâmetros internos e externos. Os parâmetros de calibração externa têm a ver com a posição e orientação do sensor relativamente a um referencial. Os parâmetros internos como a de uma câmara, têm a ver com a forma de como é adquirida a cena (distância focal, distorção, etc.). Grande parte das câmaras fornece os parâmetros internos de calibração, restando o cálculo dos parâmetros externos.

Na reconstrução 3D, a calibração dos sensores (passivos ou activos) é essencial para atingir um modelo realista. Não foram encontrados muitos métodos que realizem calibração de dois LRF sem recurso a uma câmara. A solução mais comum de calibração é de uma câmara relativamente a um LRF. É proposto em (Qilong and Pless, 2004) um método automático de calibração dos parâmetros externos de uma câmara relativamente a um laser 2D. É usado um padrão de xadrez colocado em diversas posições e ângulos relativamente ao laser e câmara. A forma de identificação do padrão de xadrez por parte da câmara é uma problemática já discutida no início da secção I.2. Algebricamente é calculada a calibração da câmara com base no tamanho do padrão de todas as capturas realizadas.

Em (Núñez et al.) é proposta a calibração de uma câmara relativamente a um laser 3D com ajuda de uma Unidade de Medição Inercial (IMU). Este método propõe um plano com padrão xadrez estacionário. A homografia que define a matriz de transformação entre o LRF e a câmara é conseguida através do movimento da plataforma que contém os sensores, tendo em conta os valores devolvidos por estes durante o movimento. É usado um método de mínimos erros quadrados no alinhamento dos cantos presentes no padrão usado capturado pela câmara e laser, obtendo no final a matriz de transformação que minimiza o erro entre os dois sensores.

Um método próximo da solução estudada nesta tese encontra-se em (Lisca et al., 2010) em que é efectuada uma calibração de um *scanner* laser multi-linha em relação a uma câmara estéreo. Com ajuda de um objecto de calibração, é obtida uma aquisição com os lasers e são usadas as nuvens de pontos e as imagens devolvidas pela câmara estéreo. Aplicando um processo de detecção de arestas a esta câmara e ao laser, os parâmetros extrínsecos do laser multi-linha são calculados. O processo é automático e não necessita de várias aquisições ou variações na posição e orientação quer dos sensores ou do objecto de calibração.

Todos os processos de calibração mencionados fazem uso de um objecto comum aos sensores para que possa ser feito um alinhamento dos mesmos, chamado de objecto de calibração. Além disso as calibrações são realizadas através da aplicação de uma matriz de transformação, que transforma o referencial do sensor que se pretende calibrar para o referencial do outro sensor, formando um só referencial.

I.3. Estrutura do documento

A primeira etapa do trabalho consistiu em estudar a plataforma usada no AtlasCar para aceder aos vários dados sensoriais para a seguir desenvolver uma plataforma de visualização e manipulação dos dados dos diversos lasers/câmaras. No capítulo II está descrita a estrutura da plataforma desenvolvida assim como a interacção e comunicação com a plataforma do AtlasCar.

De seguida é descrito no capítulo III a reconstrução 3D do ambiente obtido através do laser 3D. Partindo de uma nuvem de pontos 3D, é efectuada a triangulação desses dados obtidos pelo AtlasCar. Devido aos diversos métodos existentes para atingir esse objectivo, torna-se necessário efectuar um estudo dos vários métodos, as suas vantagens e as suas limitações de acordo com a estrutura de dados existente. Após a triangulação é aplicada textura ao modelo com imagens digitais provenientes das câmaras instaladas no AtlasCar.

A reconstrução 3D é efectuada com base apenas nos dados fornecidos pelo laser 3D, descartando os dados devolvidos pelos lasers 2D e da câmara estéreo por esta possuir demasiado ruído nos dados devolvidos. O laser 3D incide principalmente sobre a área de visão do condutor mas os lasers 2D possuem maior alcance, inclusive a zonas laterais e na retaguarda do veículo. A combinação de todos estes lasers pode fornecer mais dados quanto a obstáculos ou ambiente circundante, podendo auxiliar em manobras de pouca visibilidade, como é o caso de inversão de marcha ou estacionamento paralelo. No entanto é necessária a calibração dos lasers 2D relativamente ao laser

3D para que estes possam ser tidos em conta. Cada laser 2D está no seu referencial, o que não permite a sua integração directa com dados do laser 3D. É necessário que estejam todos no mesmo referencial, que neste caso é o do laser 3D. Esta tese constitui o primeiro passo para a reconstrução 3D com integração de lasers. É proposto no capítulo IV um método de calibração de um laser 2D relativamente a um laser 3D com uso de um objecto de calibração. Este capítulo divide-se em duas abordagens. Numa primeira fase é realizado um estudo e validação teórica do método proposto; e a aplicação do método em dados reais. Foi desenvolvida uma aplicação que realiza a calibração que com o recurso à interacção do utilizador, realiza a calibração de um laser 2D relativamente a um laser 3D.

Devido à dificuldade em obter disponibilidade do AtlasCar para realizar aquisições com os sensores e à periódica manutenção do veículo exigida, foi desenvolvido um módulo de simulação na aplicação de calibração que simula uma nuvem de pontos de ambos os lasers de acordo com um modelo de superfície dado e parâmetros dos lasers (posição, orientação). Na realização dos testes a dados reais, o módulo de simulação não é usado.

O resultado da aplicação de calibração é uma matriz de transformação a aplicar ao laser 2D para que o seu referencial passe a ser o referencial do laser 3D. Usando o módulo de simulação é possível quantificar a precisão do método visto o resultado esperado ser conhecido. No entanto em dados reais, apenas é possível nesta fase classificar a precisão do método através da análise visual dos dados por não existirem formas de a quantificar.

Está presente um anexo nesta tese. O apêndice A apresenta um exemplo das alterações necessárias à compilação da aplicação desenvolvida na secção II.4.

II. Descrição das plataformas

Os dados recolhidos dos diferentes lasers e das câmaras são todos enviados a uma unidade central que processa todos os dados recolhidos pelos sensores. A plataforma de comunicação está descrita na secção II.2.

O ambiente de programação usado e as ferramentas de desenvolvimento utilizadas estão descritos na secção II.3. As bibliotecas de apoio ao projecto e a integração de módulos do AtlasCar levam a uma escolha de uma plataforma de programação que permita a leitura e integração de dados e permitir a sua futura alteração para suportar novas funcionalidades.

A plataforma que irá suportar todas as ferramentas de interacção e a forma como ela é integrada com os dados dos sensores é descrito na secção II.4. Todos os lasers usados são agrupados na mesma janela de visualização e as imagens das câmaras são também mostradas na interface desenvolvida. Os dados recolhidos são guardados em ficheiro (*logging*) e através de uma aplicação, permite simular a missão realizada com o AtlasCar. Desta forma, é possível reproduzir a aquisição realizada pelo AtlasCar as vezes que forem necessárias sem necessidade do hardware físico presente no veículo.

II.1. Lasers e câmaras

Nesta secção iremos apresentar os vários lasers e câmaras do AtlasCar utilizados no âmbito deste trabalho.

Laser planar (2D)

O laser 2D (Fig.II.1) é um SICK LMS-100 localizado ao nível do pára-choques do AtlasCar conforme descrito na Fig.I.1. Existem dois exemplares no veículo, em lados opostos do pára-choques. O varrimento é realizado por um espelho rotativo a uma velocidade de 50 Hz resultando num campo de visão de 270°. A rede de distâncias é de até 50m. O erro associado à medição é de +/-20mm.



Fig. II.1 - Laser Planar 2D SICK

Um outro laser planar 2D está presente no AtlasCar por cima do pára-brisas orientado para obter dados a 3 ou 4 metros de distância do veículo. Este laser é um Hokuyo UTM-30LX com um campo de visão de 270° e uma rede de distâncias até 30m. Tem associado um erro de 30mm até 10m e um erro de 50mm até 30m.



Fig. II.2 - Laser Planar 2D Hokuyo

Laser 3D

Trata-se de um sensor adaptado proposto originalmente por (Dias et al., 2006), baseado numa solução comercial 2D de um laser planar SICK LMS 200. A estrutura actual do laser 3D instalado no AtlasCar está ilustrada na Fig. II.3 e foi uma estrutura proposta por (Dias, 2009). O laser 2D pode ser configurado para fornecer dados a diferentes resoluções e aberturas angulares. O laser é depois rodado em torno de um eixo externo para formar uma nuvem de pontos 3D. Com as configurações descritas, o laser 2D fornece dados a 18Hz com um alcance nominal de 20m.

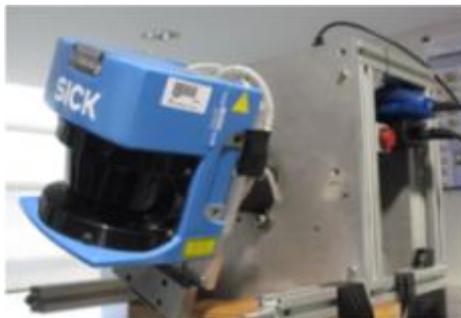


Fig. II.3 - Laser 3D adaptado

Câmara Estéreo

Este sensor (Fig.II.3) usa duas câmaras separadas por uma base conhecida com calibrações ópticas. Essa base pode ser definida como sendo uma base mais curta ou mais longa, dependendo da distância dos dados a adquirir, que neste caso concreto está fixa. Este é um sensor comercial *bumblebee xb3 stereo camera*. Disponibiliza uma resolução de 1280 por 960 pixels a um *framerate* de 15fps.



Fig. II.4 - Câmara estéreo

Câmara Pan-Tilt

O AtlasCar é equipado com duas câmaras similares (Fig.II.4), montadas uma por cima da outra em cima de uma unidade *Pan & Tilt*. Cada câmara possui uma óptica diferente, permitindo diferentes ângulos de visão e distâncias focais, sendo que uma câmara possui uma visão panorâmica e a outra possui uma visão para uma zona específica para obter maior detalhe. Este sensor é usado para visão de zonas de interesse e por essa razão está montada numa plataforma Pan-tilt permitindo modificar o foco das mesmas em função do ambiente. Ambas fornecem imagens com uma resolução de 1032 por 776 pixels a um *framerate* de 30fps.

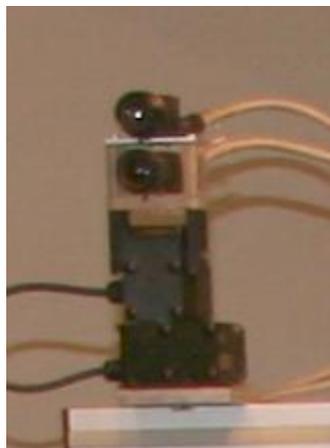


Fig. II.5 - Câmara Pan-tilt

II.2. Comunicação entre módulos e *logging*

A bordo do AtlasCar está implementada uma arquitectura modular baseada em CARMEN/IPC(CARMEN-Team), que é uma colecção de módulos *open source* para robótica móvel que permite adicionar ou modificar módulos de acordo com as necessidades.

A comunicação entre estes módulos é assegurada pela Plataforma de Comunicação Inter-processos (IPC)(CARMEN-Team). É uma aplicação que usa a troca de mensagens para a comunicação e funciona em modo distribuído. Este método permite a interoperabilidade entre diferentes sistemas ou até entre diferentes sistemas operativos. O protocolo de comunicação é baseado em sockets TCP/IP, apesar de mensagens com grande dimensão serem transmitidas em memória partilhada (Oliveira et al., 2009). O projecto AtlasCar assenta sobre esta arquitectura, em que foram acrescentados módulos de suporte aos diferentes sensores e algoritmos desenvolvidos de tratamento de dados.

Cada módulo tem associado uma mensagem própria, onde estão presentes os dados. A estrutura da mensagem é especificamente construída de acordo com o tipo de dados devolvido pelo módulo e são fornecidos métodos de acesso aos mesmos. Um módulo de uma câmara possui na sua mensagem suporte para dados vídeo enquanto que um módulo de um laser possui informação de um conjunto de coordenadas de pontos tridimensionais. Todas as mensagens produzidas pelos módulos são enviadas para o IPC, que serve de intermediário. O IPC fornece duas formas de troca de mensagens, o método publicação-subscrição, onde o módulo central recebe todas as mensagens e reencaminha-as a quem subscreveu àquele tipo de mensagens; e o método cliente-servidor, onde o cliente tem que efectuar um pedido explícito por um tipo de mensagem para a poder receber.

As mensagens que são enviadas pelos módulos são guardadas num registo. Além dos dados da mensagem, é guardado o instante de tempo de envio da mensagem. Parte da informação está contida num ficheiro xml que possui a organização das mensagens com base no instante temporal e o tipo de mensagem para identificar o tipo de módulo que a enviou. Os dados específicos de cada módulo estão armazenados em ficheiros separados.

Leitura dos *logs*

Existe um sistema de leitura dos registos guardados desenvolvida no projecto AtlasCar que baseado nos ficheiros armazenados, é possível reconstruir a missão do AtlasCar que os originou sem

necessidade de deslocar o veículo para o local de aquisição sempre que se pretende obter dados para processamento.

Esta aplicação de reprodução de dados guardados em registo simula os módulos utilizados no AtlasCar, enviando as mensagens para o IPC como se de um módulo real se tratasse. Instalando o ambiente do AtlasCar no computador pessoal, combinando com o reproduzidor de dados de registo é possível desenvolver novos módulos e aplicações sem necessidade da presença física do veículo.

Na Fig.II.5 está a interface de navegação das mensagens armazenadas, onde se pode navegar para uma mensagem específica ou colocar em *play* para reproduzir sequencialmente as mensagens. Cada linha representa um sensor (câmara ou laser), sendo que cada impulso corresponde a uma mensagem. O envio das mensagens simula um ambiente real, pelo que um módulo testado com o reproduzidor dos *logs* funcionará em ambiente real dentro do AtlasCar.



Fig. II.6 - Player utilizado para reproduzir os dados dos lasers e câmaras armazenados em Log.

Cada linha visível na interface corresponde a um sensor: três lasers planares, vídeo fornecido pelas duas câmaras pan-tilt, laser 3D, dados câmara estéreo (imagem e nuvem de pontos) e outros sensores cujos dados não são usados para o caso em particular.

II.3. Plataforma de programação

O desenvolvimento do projecto é feito em ambiente Linux para compatibilidade com os módulos e bibliotecas usadas no AtlasCar. Para a visualização e interacção com os dados dos sensores é usada a plataforma VTK (Schroeder et al., 2004) em linguagem C/C++. Esta ferramenta é livre e assenta

em gráficos 3D, processamento de imagem e visualização. Possui uma biblioteca com um conjunto de algoritmos de processamento 2D e 3D e uma vasta lista de métodos de interacção e *widgets*.

São realizadas diversas operações matemáticas que recorrem a matrizes, incluindo alguns algoritmos de reconstrução e processamento de imagem e para isso recorre-se à biblioteca Open Source Computer Vision (Bradski and Kaehler, 2008). Esta biblioteca possui uma interface C/C++ compatível com a linguagem usada no desenvolvimento do projecto.

A plataforma Carmen está desenvolvida também em linguagem C/C++ o que torna fácil a sua integração. Existem métodos para subscrever às mensagens disponibilizadas pelo Carmen que permite a execução de um método aquando a recepção de uma mensagem. Cada sensor ou câmara possui um método de subscrição próprio.

O facto de o projecto ser desenvolvido em ambiente Linux, obrigou à modificação das Makefiles existentes no AtlasCar para a integração das bibliotecas do VTK. Um exemplo das modificações a efectuar está presente no Apêndice A.

Subscrição das mensagens

No desenvolvimento da aplicação de integração dos dados sensoriais com o VTK, surgiram alguns problemas a nível da recepção das mensagens enviadas pelo Carmen. Ao iniciar uma janela de visualização do VTK, a aplicação bloqueia e aguarda eventos de interacção, impedindo a execução dos métodos definidos no momento da subscrição às mensagens. A forma de contornar esse entrave passou pela criação de um *timer*, que num período definido, executa o comando CARMEN_IPC_SLEEP. Este método permite que o subscritor aguarde por uma nova mensagem após o processamento da mensagem anterior. Sem a execução deste método, a aplicação não recebe novas mensagens.

As mensagens que são trocadas estão definidas nos módulos desenvolvidos especificamente para o AtlasCar com campos específicos para cada grupo de sensores (câmaras, lasers, etc.). No caso dos lasers, as mensagens possuem informação de coordenadas 3D e no caso das câmaras, possuem informação de imagem.

Cada mensagem com informação de dados 3D possui N triplos de coordenadas 3D, de acordo com o número de pontos definidos nos parâmetros do AtlasCar, que normalmente corresponde a um ciclo de aquisição de um sensor.

As mensagens com dados das câmaras possuem informação de uma imagem por mensagem, estando a visualização destas a cargo do openCV, que possui métodos (por omissão) para visualização de imagens.

Processamento das mensagens

O processamento das mensagens é despoletado pelo *Timer* que é executado com período de 30 milissegundos. Este valor foi estipulado para que os dados visualizados sejam em tempo real, tendo como base a frequência das câmaras (30 fps).

Este método acarreta alguns problemas de desempenho pois são acedidas todas as mensagens de todos os sensores e câmaras sempre que o *timer* é despoletado. Para evitar esta situação, são definidas *flags* que sinalizam a recepção de uma nova mensagem para um determinado sensor/câmara. Cada método de subscrição permite definir um método que é executado sempre que é recebida uma mensagem da sua subscrição. A activação da *flag* é efectuada neste método definido. Desta forma, sempre que o *timer* é activado, apenas são actualizadas na janela de visualização os dados das mensagens recebidas.

II.4. Visualização do modelo

Foi desenvolvida uma aplicação de visualização dos dados sensoriais do AtlasCar. Recorrendo à ferramenta VTK, que disponibiliza métodos para a visualização de dados, cada nuvem de pontos de cada sensor foi armazenada numa estrutura de dados independente que é adicionada ao visualizador. A separação em diferentes estruturas permite tratar os dados separadamente e atribuir diferentes representações em termos de cores para fácil identificação. Na Fig. II.7 estão representados os pontos vindos de vários lasers e câmaras. A vermelho estão os dados da câmara estéreo; a preto os dados do laser 3D e a verde os dados vindos do laser planar Hokuyu. O visualizador suporta a adição/remoção de novos lasers/câmaras que possam ser futuramente instaladas no AtlasCar, bastando para isso definir os métodos de subscrição às mensagens do novo laser/câmara, ler os dados vindos deste e acrescentar à janela de visualização.



Fig. II.7 - Visualização dos dados do AtlasCar da oficina do DEM.

Em cima: Nuvens de pontos recolhidos pelo AtlasCar em tempo real, diferenciado por cores. Em baixo: imagens recolhidas pelas diversas câmaras presentes no AtlasCar (as duas primeiras das câmaras pan-tilt e a última da câmara estéreo)

Cada laser/câmara possui uma estrutura VTK onde estão armazenados os dados. Sempre que é recebida uma nova mensagem, é usada a mesma estrutura em que apenas são substituídos os dados já presentes pelos novos dados. Após cada actualização da estrutura é forçada a actualização do modelo associado na janela de visualização para que as alterações tenham efeito no conjunto final.

II.5. Ferramentas de interacção

Foram desenvolvidas ferramentas para interacção com o modelo para melhorar a visualização do modelo ou para filtrar dados. O VTK disponibiliza métodos que facilitam a implementação e que poderão ser úteis na manipulação do modelo 3D. Foram implementadas as seguintes funcionalidades:

- Navegação no modelo: usando o rato e o teclado, é possível rodar o modelo, ampliar e mover, ajudando a visualização e/ou ajustar a posição do modelo na janela de visualização;
- Adicionar/Remover pontos dos diferentes dispositivos de aquisição dos pontos: cada dispositivo está representado com uma cor distinta para que cada um possa ser identificado facilmente e ver a sua contribuição para o modelo. Desta forma é possível adicionar ou

remover a representação dos pontos dos diversos dispositivos para ver dados isolados de um só laser;

- Acumulação de pontos: é possível activar o modo de acumulação em que a alteração que introduz é que os novos dados vindos de uma nova mensagem não substituem os dados já presentes na estrutura mas são acrescentados aos dados já existentes. Esta opção poderá justificar-se no caso do Laser 3D, onde os pontos adquiridos numa só mensagem criam uma nuvem de pontos pouco densa e existe a necessidade de acumular várias mensagens de forma a obter um modelo mais completo;
- Guardar o modelo visualizado em ficheiro: é possível exportar o modelo a ser visualizado na janela de visualização em ficheiro no formato .OBJ e .PLY. Os formatos têm a ver com a organização das estruturas que representam os pontos e/ou superfícies e são os mais usados pelos *softwares* de modelação 3D (MeshLab, AutoCad,etc.).

III. Reconstrução 3D

Com a aplicação de visualização dos dados desenvolvida, é possível obter os pontos necessários para efectuar a triangulação. Devido à configuração do laser 3D especificada na secção II.1, o facto de este rodar sobre um eixo externo, criará uma zona onde a densidade de pontos é maior, na proximidade do eixo de rotação e menor à medida que se afasta desse centro. Esse efeito é visto nos exemplos nas secções III.5.1e III.5.2. Isto cria zonas com menor resolução e que dificulta o detalhe mas que são limitações subjacentes ao laser.

III.1. Reconstrução de malhas 3D

Serão apenas referidas as reconstruções baseadas em nuvens de pontos não-estruturadas produzidas a partir de lasers, visto ser esta a organização dos pontos devolvidos pelos diversos sensores. O objectivo da reconstrução das malhas 3D não-estruturada baseia-se no seguinte:

Dado um conjunto de pontos num espaço 3D, assumindo que estes pertencem ou estão próximos de uma superfície S , criar uma superfície S' aproximada à superfície S . A reconstrução exacta da superfície S não pode ser garantida, visto existirem um número limitado de pontos e muito menos se os pontos são não-estruturados, pois não existe informação do vector ortogonal desse ponto relativo á superfície nem são conhecidos os pontos da sua vizinhança. Essa vizinhança deixa de fazer sentido devido à integração de diversos lasers na mesma estrutura e de várias aquisições sucessivas. No entanto não se trata de um problema insolúvel, visto que o objectivo é obter uma superfície que descreva a geometria básica do ambiente de forma a identificar obstáculos ou limites dados por edifícios ou uma estrutura semelhante em tempo útil na navegação do veículo.

Para uma reconstrução mais fidedigna, o fornecimento de dados adicionais como a vizinhança, as normais ou uma maior densidade de pontos facilita todo o processo, mas neste caso não existe nenhuma desta informação, apenas a localização espacial dos pontos. Além de desconhecer a topologia à priori, estão associados erros e ruído dos próprios sensores (lasers e câmara estéreo). Os erros dizem respeito à detecção de um ponto que está deslocado em relação à superfície real e o ruído está associado à detecção de um ponto onde na realidade não existe nenhuma superfície.

III.2. Algoritmos de triangulação em estudo

Na triangulação, todos os algoritmos oferecem parâmetros de afinação que permitem adaptar o algoritmo mediante os dados utilizados. O grande desafio nesta área é descobrir o melhor algoritmo que satisfaz um dado objectivo, sem ter a necessidade de os implementar todos e fazer testes da sua eficácia. Surge portanto a necessidade de estudar os algoritmos e perceber se concretizam o efeito esperado. Ao longo dos anos foram surgindo novas formas de tratar a triangulação, uns baseados em métodos já existentes mas que foram afinados para um determinado caso ou o surgimento de novos métodos de cálculo destas triangulações.

Foram analisados algoritmos baseados em estruturas em que os pontos formam uma malha fechada e regular e alguns deles necessitam da informação da direcção da sua Normal.

Foram estudados os algoritmos seguintes:

- **Delaunay 2D**

Este é o algoritmo de base para a reconstrução 2D ou 3D. O cálculo do diagrama de Voronoi é a base deste algoritmo. É um algoritmo genérico e muito usado na obtenção de um modelo triangulado. Existem muitos algoritmos que se baseiam no princípio deste algoritmo com o intuito de melhorar os seus resultados.

- **Crust**

Este algoritmo foi proposto por (Amenta et al., 1998) e é baseado num diagrama de Voronoi tridimensional e triangulação de Delaunay. O maior problema deste algoritmo são os tempos de processamento. Segundo o autor, apenas para o cálculo do diagrama de Voronoi, numa estrutura com 10000 pontos são necessários alguns minutos. Este facto inviabiliza o seu uso num caso real onde o que se pretende é resultados num curto espaço de tempo. O laser 3D instalado no AtlasCar devolve à volta de 15000 pontos por cada mensagem, o que implica tempos de processamento demasiado elevados.

- **Cocone**

Este algoritmo proposto por (Amenta et al.) tem como base o algoritmo Crust, contudo num modelo mais simplificado. O algoritmo reconstrói a superfície T a partir da triangulação usando Delaunay. É calculado apenas um diagrama de

Voronoi em comparação com o Crust e realiza restrições no cálculo dos triângulos que minimizam os tempos de execução. No entanto este algoritmo supõe uma superfície suave (sem cantos ou zonas com grande variação de geometria) com grande densidade e distribuição uniforme dos pontos. No nosso caso em particular nenhuma destas condições são verificadas.

- **Poisson Reconstruction**

Este algoritmo proposto por (Kazhdan et al., 2006) faz uso de uma função implícita para a reconstrução de superfícies a partir de pontos orientados, abordando a solução com formulação de Poisson. Tal como (Kazhdan, 2005), é calculada uma função 3D indicadora, atribuindo um peso a cada ponto (valor 1 se este se encontra no interior do modelo e valor 0 se o ponto se encontra no exterior do modelo), obtendo a superfície reconstruída extraíndo a iso-superfície.

É defendida que existe uma relação entre os pontos orientados na superfície do modelo com a função indicadora. Em especial, o gradiente da função indicadora é um campo vectorial que tem valor zero na maior parte dos pontos (visto que o valor da função indicadora é constante em grande parte do modelo), excepto nos pontos próximos da superfície onde o valor é igual à normal no interior da superfície. Deste modo, os pontos orientados podem ser vistos como o gradiente da função indicadora (Fig. III.1).

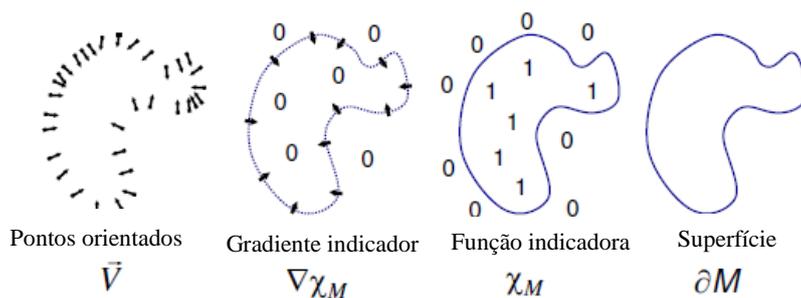


Fig. III.1 - Ilustração da reconstrução de Poisson em 2D retirado de (Kazhdan et al., 2006)

A natureza dos pontos obtidos pelo AtlasCar não nos permite aplicar este algoritmo pois não possuímos informação do vector referente à normal dos pontos.

- **LayerPeeling**

Este algoritmo proposto por (Lim and Tan, 2006) utiliza o princípio de vizinhança na construção de superfícies. Faz uso de critérios de convexidade local para extrair um conjunto de superfícies na sua vizinhança, para cada ponto, sob a forma de discos. Depois disso é aplicado um princípio de projecção para criar apenas uma superfície final, unindo os diversos discos formados em cada ponto. É projectado um raio de cada um dos vértices dos triângulos presentes no disco com direcção igual à normal desse disco. A intercepção desses raios com outros discos de outros pontos resulta numa junção dos dois discos, como ilustrado na Fig. III.2.

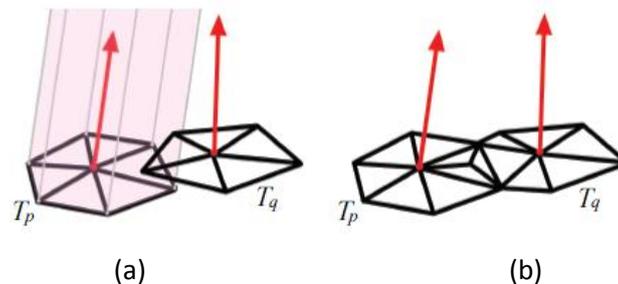


Fig. III.2 - União de dois discos T_p e T_q do algoritmo Layer Peeling
Em (a) são mostrados os raios de projecção do disco T_p que tocam numa face em T_q . Em (b) os dois discos são unidos para formar uma só superfície. Imagem retirada de (Lim and Tan, 2006)

- **Ball Pivoting**

O princípio básico deste algoritmo é criar uma esfera com um determinado raio e a partir de um triângulo inicial, a esfera roda sobre uma aresta do primeiro triângulo até tocar num outro ponto, criando um novo triângulo contendo o novo ponto como vértice. É efectuado este procedimento sucessivamente até não ser possível unir mais pontos. Este algoritmo funciona sobre dados não estruturados 2D ou 3D sem imposição da sua densidade ou distribuição. O tamanho do raio da esfera a usar é configurável para permitir uma adaptação ao tipo de dados fornecido.

Todos estes algoritmos, com excepção dos dois últimos são baseados em Delaunay ou são derivados da ideia original do algoritmo. De acordo com o funcionamento dos diversos algoritmos e por serem variantes do Delaunay, operam em 2,5D e implicam algumas limitações. Uma forma de aplicar estes algoritmos ao caso em estudo, seria projectar o modelo visualizado de acordo com o plano de visualização (2D), efectuar uma triangulação e re-projectar para 3D. Um problema desta aplicação surge nas zonas fronteira nos objectos da cena, ou seja, com a projecção em 2D, pontos

que estão afastados na realidade, irão aparecer como vizinhos nessa projecção. Após a triangulação e aquando a re-projecção em 3D, esses pontos podem ser unidos, criando falsos triângulos. Este fenómeno é difícil de controlar e depende muito da cena visualizada e da sua geometria. Além disso este tipo de algoritmos possuem geralmente um grau de complexidade de $O(n^2)$ e supõem superfícies com geometria suave, grande densidade de pontos e distribuição uniforme dos mesmos. O laser 3D instalado no AtlasCar inviabiliza essa restrição pela sua distribuição de pontos.

O algoritmo Ball Pivoting funciona directamente sobre os dados 2D ou 3D e não faz uso de projecções para efectuar a triangulação. Além disso, o algoritmo funciona em malhas abertas ou fechadas e em dados não estruturados. Este é o algoritmo que se adequa ao tipo de dados que possuímos e existem versões implementadas em *open source*. É explicado em mais detalhe o funcionamento deste algoritmo na secção seguinte. O algoritmo LayerPeeling é semelhante ao Ball Pivoting mas a sua implementação não é disponibilizada o que fez eleger o Ball Pivoting como o algoritmo a usar e o facto de este permitir aceder directamente ao código fonte e adaptar o algoritmo ao tipo de dados caso seja necessário.

III.3. Algoritmo Ball Pivoting

O algoritmo Ball Pivoting (BPA) (Bernardini et al., 1999) funciona tanto em ambientes 2D como em 3D e adapta-se muito bem ao nosso caso visto este funcionar sobre nuvens de pontos não estruturadas. O seu funcionamento é simples e aplica-se a diferentes ambientes com diversas geometrias, em malhas fechadas ou não.

O BPA é um algoritmo de interpolação baseado em regiões de crescimento: tem como ponto de partida um triângulo inicial e os pontos são acrescentados a essa estrutura inicial até que todos os pontos tenham sido considerados.

Considerando um conjunto de pontos de uma superfície de um objecto tridimensional com uma relativa densidade de tal forma que uma esfera (de raio i) não consegue atravessar a superfície sem entrar em contacto com nenhum ponto adjacente. Um exemplo em 2D está ilustrado na Fig. III.3(a), em que uma circunferência de um dado raio desloca-se ao longo dos pontos da superfície do objecto sem atravessar para a superfície.

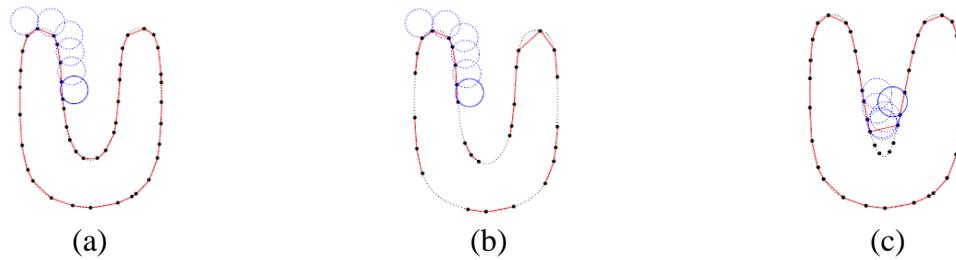


Fig. III.3 - Algoritmo Ball Pivoting em 2D.

Colocando inicialmente uma esfera de raio i no triângulo inicial tal que a esfera está em contacto com dois pontos desse triângulo, esta desloca-se sobre um dos pontos em contacto até que toque num outro ponto, formando um novo triângulo (ver Fig. III.4 para um esquema 3D deste processo). Os dados reais, obtidos no caso particular através de lasers 3D, têm erros associados, surgindo zonas com menor densidade de pontos, superfícies desalinhas e até *outliers*, que são pontos que não representam uma superfície ou que estão distantes da superfície original. O BPA não efectua nenhuma filtragem ou médias de pontos adjacentes.

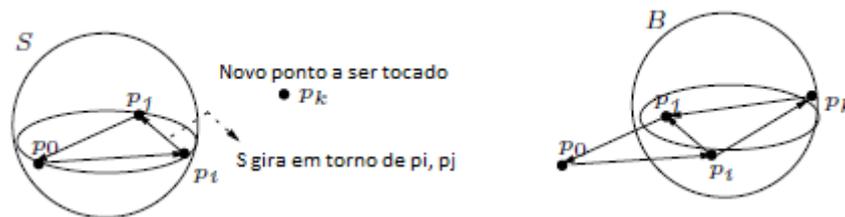


Fig. III.4 - Funcionamento do algoritmo Ball Pivoting

A presença de ruído na nuvem de pontos pode criar ambiguidade em casos específicos. Áreas de densidade maior do que o raio da esfera usada, ou seja, cuja distância entre os pontos seja menor do que o raio da esfera, não representam problemas ao algoritmo e a não existência de ruído nos dados fará com que todos os pontos sejam interpolados. A Fig. III.5(a) exemplifica o caso em que existe ruído, na medida em que não existe uma superfície regular, com pontos localizados fora da superfície real. A união dos pontos exclui dois pontos devido ao tamanho da circunferência.

Nos casos onde existe subamostragem de pontos, o algoritmo não preencherá esses dados em falta; em particular, o BPA pode ser aplicado diversas vezes com diferentes valores do raio da esfera até preencher a subamostragem. No entanto existem ambiguidades que são criadas pela falta de dados. Quando a esfera roda sobre uma zona limite, a esfera pode tocar um ponto que resulta de ruído ou de uma outra superfície. Para fazer face a esta ambiguidade o algoritmo faz uso das normais da superfície para decidir se o novo ponto é ou não válido (Fig. III.5 (b)). Um triângulo é rejeitado se o produto interno da sua normal com a normal da superfície é negativo. O ângulo máximo formado entre os dois vectores é um parâmetro do algoritmo.

A presença de pontos a uma distância superior à do tamanho da esfera, criará duas superfícies distintas. Este tipo de resultados pode ser ou não desejado e é influenciado pelo ruído presente nos dados. Em vez da criação de uma só superfície são criadas duas superfícies que aparentam representar duas superfícies distintas. Um exemplo em 2D está presente na Fig. III.5 (c).

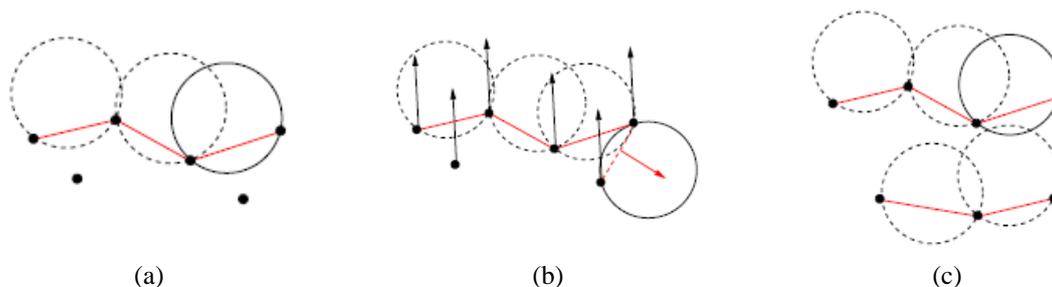


Fig. III.5 - BPA em presença de dados com ruído

III.4. Integração com o VTK

Uma variante do BPA encontra-se implementada na biblioteca usada pela aplicação MeshLab. Esta aplicação é de código livre e trata-se de uma ferramenta de visualização, processamento e edição de dados 3D não estruturados. A biblioteca usada por esta aplicação é a VcgLib (VCGLib, 2010) e será usada no projecto para o uso da implementação do algoritmo BPA.

A implementação do BPA presente nesta biblioteca depende de diversos módulos e usa estruturas de armazenamento de pontos e superfícies próprias e não foi possível integrar directamente nas estruturas do VTK. A solução passou por converter da estrutura usada pelo VTK para o formato usado no VcgLib. É depois feita a chamada à função do BPA com a estrutura preenchida com a nuvem de pontos e é devolvida uma estrutura com informação das superfícies criadas. Por fim é feita uma reconversão dos dados para a estrutura do VTK. Este método é eficaz e resolve as questões de compatibilidade, não justificando a procura de outra implementação do algoritmo.

Esta biblioteca tem que estar armazenada na pasta do projecto, podendo alterar a sua localização tendo que alterar o caminho presente na Makefile. Nas secções seguintes deste capítulo estão presentes os resultados da aplicação deste algoritmo, analisando os parâmetros de entrada do algoritmo e a sua influência no resultado.

III.5. Resultados

Foram guardados diversos ficheiros contendo números diferentes de mensagens do Laser3D correspondentes a nuvens de pontos mais ou menos densas. Numa das aquisições feitas com o AtlasCar, o tempo usado na aquisição resultou em 5 mensagens (~75 000 pontos), pelo que para comparação, as outras aquisições foram limitadas a um máximo de 5 mensagens.

Realizaram-se testes para cinco mensagens (~75 000 pontos) e uma mensagem (~15 000 pontos), variando os parâmetros do algoritmo para analisar o seu comportamento. As aquisições usadas foram as da Igreja da N^a Senhora das Febres, Igreja de Vera Cruz e Oficina do Departamento de Engenharia Mecânica.

Os parâmetros do algoritmo BPA alterados são:

1. Raio da esfera – define o raio fixo da esfera aplicado pelo algoritmo;
2. Tamanho mínimo de uma aresta, ou seja, se existirem pontos em que a distância entre eles é menor do que este valor definido, é simplesmente ignorado. É definido como uma percentagem do raio da esfera.

Além destes parâmetros é possível alterar o ângulo máximo formado entre duas faces, que está definido como sendo o $\cos(\alpha)$ e por omissão o algoritmo define $\alpha = \frac{\pi}{2}$. Este valor não necessita de ser alterado pois não se pode assumir um ângulo máximo se não existe qualquer informação a priori da topologia. Estando o valor fixado em zero apenas garante que a nova aresta não coincida consigo próprio. Em certas aplicações poderá fazer sentido definir um ângulo máximo mas não neste caso.

Um outro parâmetro possível de alterar é o comprimento máximo da aresta, que quanto maior for, levará a uma menor definição, ilustrado na Fig. III.3(c). Por uma questão lógica, a aresta nunca será superior ao dobro do valor do raio, mas que por omissão está definido como sendo:

$aresta_{máxima} = 1,8 * r$, sendo r o raio da esfera, que empiricamente se verificou ser o mais adequado.

O algoritmo possui uma opção de configuração automática do raio da esfera, cujo valor calculado por este é obtido através da seguinte fórmula:

$$r = \sqrt{\frac{DiagonalEspacial(boundingBox)^2}{n_{vertices}}}$$

, onde *boundingBox* representa um paralelepípedo que delimita o modelo 3D.

Este resultado não é mais do que um cálculo de uma distância média entre os vértices para um dado tamanho da *Bounding Box* do modelo. Esta fórmula será comparada e analisada com valores definidos manualmente de forma a identificar se o valor obtido é o mais adequado ou não.

III.5.1. Nuvens de pontos pouco densas

De seguida mostram-se alguns resultados obtidos com algumas aquisições efectuadas e com apenas 1 mensagem, que rondará os 15 000 pontos.

Igreja de Vera Cruz:

Foi obtida uma imagem a partir da câmara Pan-Tilt ilustrada na Fig. III.6. Analisando os resultados, vemos que quanto maior for o raio, maior área de superfície haverá no modelo, no entanto perde definição. Por exemplo, na Fig. III.7 (ao centro) as escadas parecem uma rampa e definindo um raio menor como na Fig. III.7 (à esquerda), os degraus são mais visíveis mas surgem muitas zonas não trianguladas no modelo.



Fig. III.6 - Imagem da Igreja de Vera Cruz
Esta imagem foi obtida através de uma das câmaras da pan-tilt.

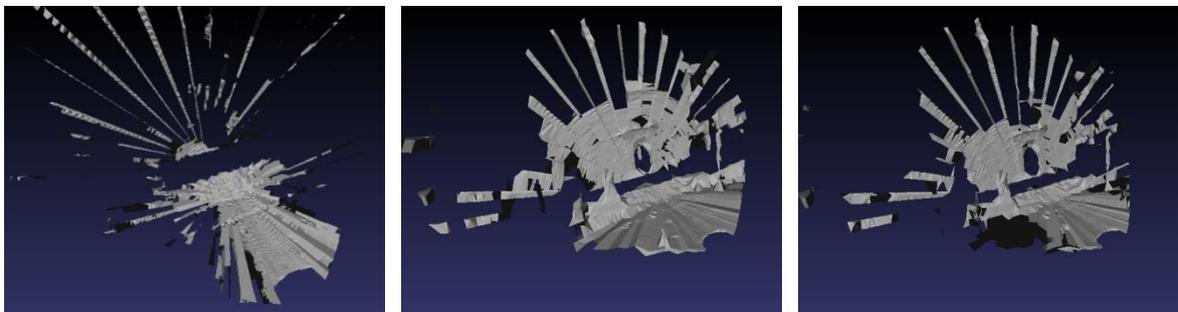


Fig. III.7- Triangulação do modelo da Igreja de Vera Cruz com nuvens de pontos pouco densas.
Variando o parâmetro do raio no algoritmo, da esquerda para a direita, com raio = 0.2, raio = 0.7 e configuração automática do raio (0.57975).

Com a opção de configuração automática activada e alterando o tamanho mínimo da aresta para 1% do raio em vez dos 5%, neste caso, a diferença não é visível pois os pontos estão muito espaçados. Onde poderia ser mais notória a diferença seria na zona central, onde a concentração de pontos é maior.

Igreja da N^a Sra. das Febres

Num outro exemplo, verifica-se o mesmo comportamento em relação à variação do raio da esfera usada pelo algoritmo de triangulação. Usando um raio pequeno (Fig. III.9 à esquerda), o modelo torna-se muito incompleto à medida que se afasta do centro óptico (referido no capítulo III), que neste caso se situa no centro da escadaria. Usando a configuração automática conseguimos maior detalhe (Fig. III.9 à direita) mas existem algumas zonas abertas. Na Fig. III.9 (ao centro) perde-se um pouco no detalhe mas consegue-se uma maior área triangulada. Está presente na Fig. III.8 uma imagem da área onde foi realizada a aquisição.



Fig. III.8 - Imagem da Capela da N^a Sra. das Febres
Esta imagem foi obtida através de uma das câmaras da pan-tilt.

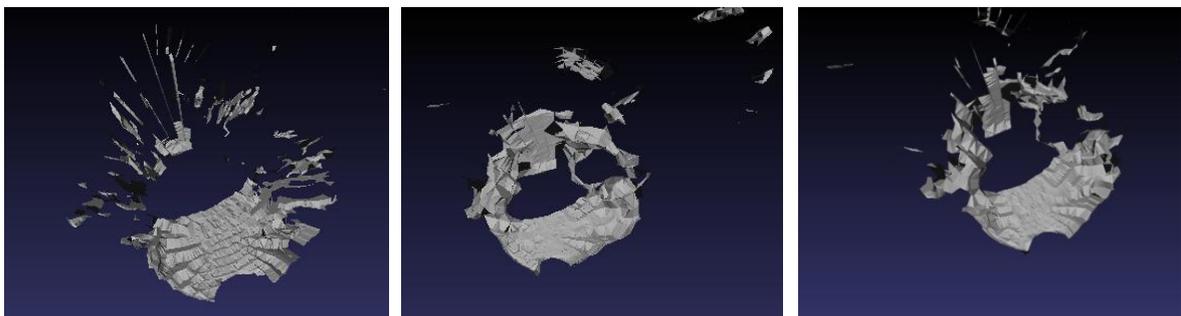


Fig. III.9 - Triangulação do modelo da Igreja N^a Sra das Febres com nuvens de pontos pouco densas.
Variando o parâmetro do raio no algoritmo, da esquerda para a direita, com raio = 0.3, raio = 0.7 e configuração automática do raio (0.5).

Oficina do Departamento de Engenharia Mecânica

Neste último exemplo, os dados adquiridos são de natureza diferente dos adquiridos anteriormente visto não representarem um edifício mas um ambiente amplo e daí os resultados serem diferentes. Dado existirem zonas bastante planas, a zona triangulada não é demasiada afectada pelo detalhe. Verifica-se que com aquisições de nuvens de pontos menos densas, um raio maior da esfera permite criar uma superfície mais completa do que com raios pequenos, perdendo em questão de detalhe. Está presente na Fig. III.10 uma imagem da zona de aquisição. Na Fig. III.11 estão presentes triangulações de modelos variando o tamanho do raio da esfera. O comportamento é o mesmo do verificado nos exemplos anteriores.



Fig. III.10 - Imagem da Oficina do Departamento de Engenharia Mecânica
Esta imagem foi obtida através da câmara estéreo.

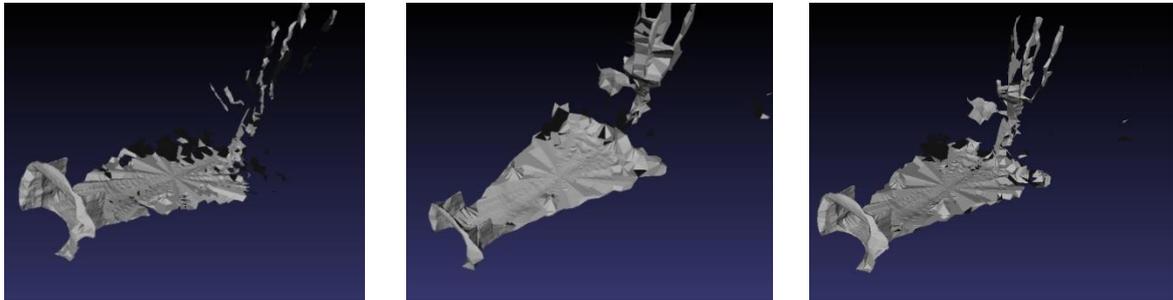


Fig. III.11- Triangulação do modelo do departamento do DEM com nuvens de pontos pouco densas.
Variando o parâmetro do raio no algoritmo, da esquerda para a direita, com raio = 0.3, raio = 0.7 e configuração automática do raio (0.509146).

Olhando para um detalhe da Fig. III.12, estão marcados a vermelho 2 pequenos postes que se encontravam no passeio na altura da aquisição e que por serem estreitos e definindo um raio da esfera maior, estes aparecem como sendo umas pirâmides, dando sensação de um obstáculo maior do que na realidade é. É uma desvantagem na triangulação de modelos com menor número de pontos.

Em modelos com poucos pontos, o melhor é definir raios um pouco superiores ao indicado pela configuração automática, poupando em tempo de processamento mas perdendo definição do modelo.

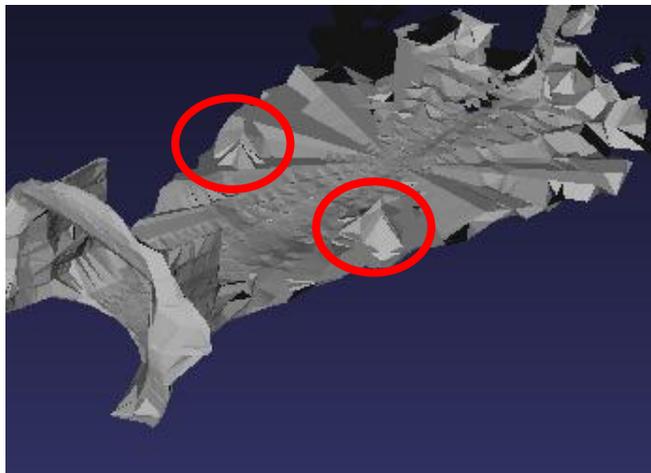


Fig. III.12 - Pormenores da oficina do DEM para o exterior

III.5.2. Nuvens de pontos densa

Segue-se a análise das mesmas aquisições estudadas anteriormente mas com cinco mensagens do Laser 3D, que corresponde a aproximadamente 75 000 pontos.

Analisando neste caso uma nuvem de pontos mais densa, o aumento ou diminuição do raio da esfera tem maior influência na definição dos objectos.

Com o aumento do número de pontos, o método de configuração automática fez com que o raio da esfera fosse aproximadamente de 0.2 em vez dos ~0.5 usando apenas 1 mensagem.

Quando se diminui o raio da esfera, o resultado é o mesmo daquele obtido nos casos estudados na secção anterior, ou seja, existem muitos espaços por triangular no modelo e conseguimos maior definição em zonas com mais concentração de pontos. Como foi referido, o objectivo é criar modelos 3D com a geometria básica e a diminuição do tamanho do raio não só implica maior poder computacional como produz modelos com zonas não trianguladas.

Quando usamos os valores da configuração automática (Fig. III.13, Fig. III.14 e Fig. III.15 à direita), os resultados são um equilíbrio entre definição e de zonas preenchidas. Nas zonas de maior concentração de pontos, o resultado é bastante satisfatório mas quanto mais afastado se está do

centro, mais esparsos os pontos se situam e fazem com que o algoritmo falhe na triangulação dessas áreas.

Quando definimos um valor de raio um pouco superior (Fig. III.13, Fig. III.14 e Fig. III.15 ao centro), obtemos modelos com maior área triangulada, perdendo um pouco na definição.

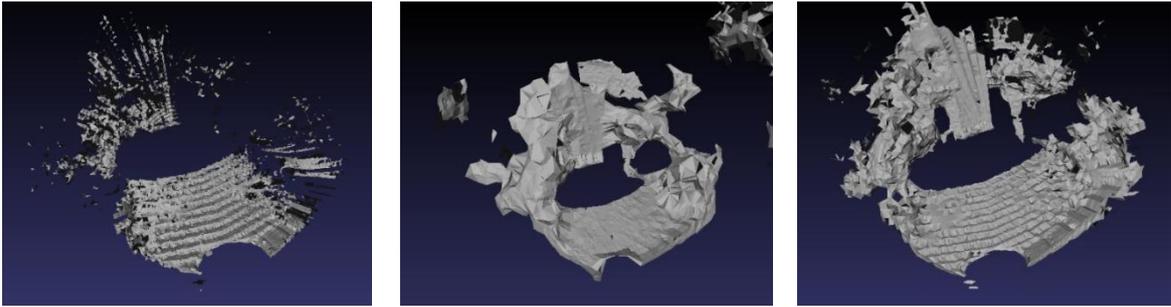


Fig. III.13 - Triangulação do modelo da Capela da N^a Sra. das Febres com nuvens de pontos densas. Variando o parâmetro do raio no algoritmo, da esquerda para a direita, com raio = 0.1, raio = 0.7 e configuração automática do raio (0.248875).

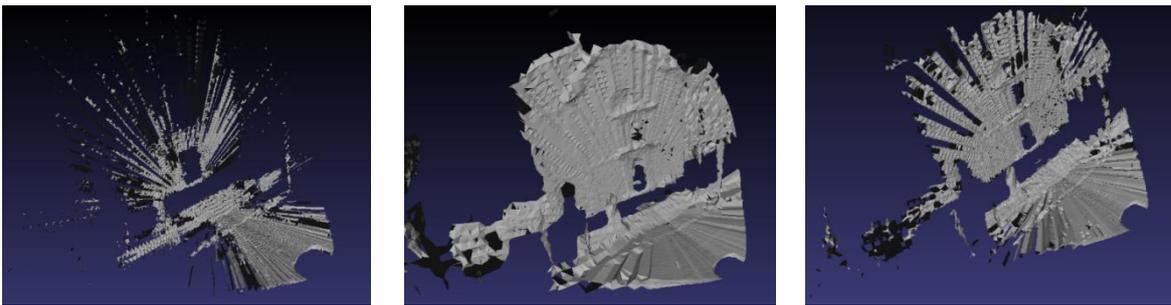


Fig. III.14 - Triangulação do modelo da Igreja de Vera Cruz com nuvens de pontos densas. Variando o parâmetro do raio no algoritmo, da esquerda para a direita, com raio = 0.1, raio = 0.5 e configuração automática do raio (0.264425).

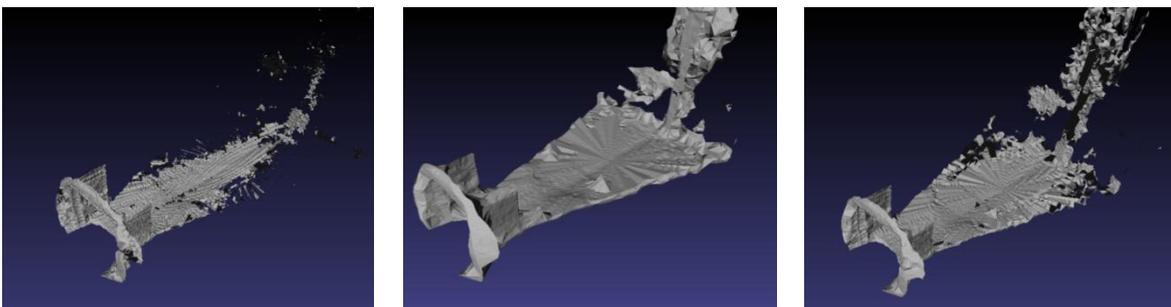


Fig. III.15 - Triangulação do modelo da oficina do DEM com nuvens de pontos densas. Variando o parâmetro do raio no algoritmo, da esquerda para a direita, com raio = 0.2, raio = 0.5 e configuração automática do raio (0.231293).

Nestas aquisições com uma nuvem de pontos mais densa foi alterado o valor do parâmetro do tamanho mínimo da aresta do triângulo para 1% do raio em vez do valor por omissão (5%). As

triangulações resultantes não diferem das já obtidas e conclui-se portanto que não é necessário alterar o valor. Definindo uma percentagem maior, obtém-se um modelo com menos triângulos mas com menor definição.

Em nuvens de pontos com maior densidade, é possível obter maior detalhe nas triangulações, levando a um maior tempo de processamento. Usando o cálculo automático do raio, os resultados são equilibrados, sendo que à medida que se afasta do centro, os pontos estão demasiados espaçados e produzem buracos no modelo. Para obter uma maior superfície triangulada, menos triângulos e por consequência, diminuindo o poder computacional, o mais adequado é usar um raio da esfera um pouco superior ao obtido com a configuração automática. Este valor situa-se entre 0.1 e 0.3, tal como foi usado nos resultados presentes nesta secção.

III.5.3. Análise de desempenho

Uma mensagem (15 000 pontos)			
Capela N^a Sra das Febres	R=0.3	R=0.7	Configuração automática (r=0.5)
	2,68s	5,96s	5,43s
Oficina DEM	R=0.3	R=0.7	Configuração automática (r=0.509146)
	3,77s	4,65s	3,22s
Igreja de Vera Cruz	R=0.2	R=0.7	Configuração automática (0.57975)
	2,3s	5,72s	4,82s

Tabela III.1- Tempos de processamento para 1 mensagem (~15 000pontos).

Cinco mensagens (75 000 pontos)			
Capela N^a Sra das Febres	R=0.1	R=0.7	Configuração automática (r=0.248875)
	44,92s	13,307s	41,9s
Oficina DEM	R=0.1	R=0.5	Configuração automática (r=0.231293)
	44,61s	135,14s	66,21s
Igreja de Vera Cruz	R=0.1	R=0.5	Configuração automática (r=0.264425)
	44,05s	86,23s	70,01s

Tabela III.2 - Tempos de processamento para 5 mensagens (~75 000 pontos)

média(1 mensagem) = 4,283s

média(5 mensagens) = 60,709s

III.6. Conclusões

O compromisso de qualidade/desempenho influencia a escolha dos parâmetros a usar na geração das malhas de superfície com o BPA. Num cenário em que a nuvem de pontos é pouco densa, a identificação de elementos geométricos na cena poderá ser dificultada visto haver perda de definição, sobretudo na periferia onde a distância entre os pontos aumenta consideravelmente. Tendo isto em conta e verificando os resultados obtidos no estudo efectuado na secção III.5, o valor do raio da esfera para o qual se obtêm modelos aceitáveis situar-se-á entre o valor obtido pela configuração automática e duas ou três décimas acima desse valor.

Verifica-se também que quanto mais denso é o modelo, menor é o raio definido, verificando neste caso uma diminuição de ~0.5 nos modelos com uma mensagem para ~0.2 nos modelos com cinco mensagens.

Na Tabela III.1e Tabela III.2, a diferença dos tempos de processamento para uma ou cinco mensagens é bastante significativo e dependendo do objectivo pretendido, a escolha do método é importante. Para uma triangulação rápida, o primeiro caso é o mais indicado apesar de perder detalhe. Para uma triangulação mais precisa, o segundo método será o mais adequado, sendo o tempo médio 14 vezes superior, tendo apenas 4 mensagens a mais. Com estes valores, tendo em conta a exigência computacional e a precisão obtida, o uso de quatro ou cinco mensagens na geração de malhas será um número adequado.

III.7. Aplicação de textura

O objectivo da textura é associar a cada triângulo do modelo um determinado pixel/conjunto de pixéis de uma fotografia digital. Esta transformação não é imediata, sendo necessária a rotação e/ou translação a aplicar para que cada píxel da imagem seja mapeado de forma correcta. Cada câmara instalada no veículo não está posicionada no centro de coordenadas deste, ou seja, cada câmara está na origem do seu sistema de coordenadas. Devido a isto, as câmaras sofrem uma primeira transformação do seu sistema de coordenadas para o centro de coordenadas do veículo, ilustrado na Fig. III.16.

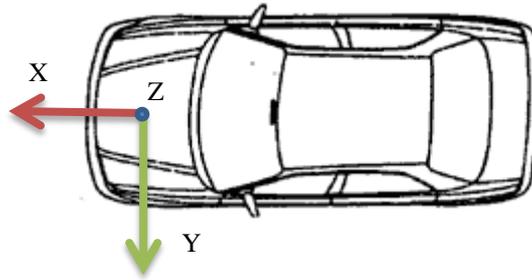


Fig. III.16 - Sistema de coordenadas de referência do AtlasCar.

A coordenada X está na direção do campo de visão do condutor; a coordenada Y está em direção ao lado esquerdo do condutor; e a coordenada Z está na direção vertical.

As câmaras têm que ser calibradas para que os pixéis sejam mapeados de forma correcta à superfície. Os parâmetros são chamados de intrínsecos e extrínsecos. O primeiro tem a ver com os parâmetros internos da câmara como distância focal, distorção da imagem, etc.; o segundo tem a ver com parâmetros externos como translação e rotação.

Apesar destas primeiras transformações, as imagens captadas pelas câmaras não têm o mesmo tipo de calibração entre elas e para aplicar várias imagens capturadas no modelo triangulado, são necessários cálculos das diversas matrizes de transformação (parâmetros intrínsecos e extrínsecos) para aplicar a cada imagem de cada câmara. Descreve-se no ponto seguinte em mais detalhe as matrizes de transformação.

III.8. Matrizes de Transformação de coordenadas

As transformações são descritas nas matrizes seguintes:

- Cada ponto (P) descrito no espaço tridimensional, é expresso nas coordenadas x, y e z e pode ser descrito no formato de coordenadas homogéneas:

$$P = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- A matriz de transformação (T) pode ser expressa da seguinte forma:

$$T = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Como não há mudança de escala entre o modelo e câmara, a transformação é do tipo transformação linear de corpo rígido. Sendo assim, a matriz T pode ser dividida em dois grupos:

$$T = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotação →
← Translação

Na matriz intrínseca estão presentes cinco parâmetros da câmara. Estes contêm distância focal, formato da imagem e centro óptico. A matriz pode ser expressa da seguinte forma:

$$A = \begin{bmatrix} \alpha_x & \gamma & u_0 & 0 \\ 0 & \alpha_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Sendo que $\alpha_x = f \cdot m_x$ e $\alpha_y = f \cdot m_y$ representam a distância focal em termos de pixéis, onde m_x e m_y são factores de escala, relacionando os pixéis à distância. γ representa o coeficiente de distorção entre os eixos x e y, e que no caso em estudo é igual a zero. u_0 e v_0 representam o centro focal.

A transformação das coordenadas tridimensionais no espaço para coordenadas bidimensionais na imagem é dada por:

$$[uvw]' = A T [x y z 1]'$$

Onde “u”, “v” e “w” representam as coordenadas de mapeamento do espaço 3D em 2D.

Para calcular a que pixel da imagem corresponde um dado ponto seleccionado no modelo, podemos usar a fórmula descrita acima, ficando:

$$x_{img} = \frac{u}{w}$$

$$y_{img} = \frac{v}{w}$$

Para a aplicação da textura foram utilizadas as calibrações já calculadas do AtlasCar. A matriz de transformação foi aplicada aos dados para obter a correspondência entre os pixels da imagem (x_{img} e y_{img}) e os triângulos do modelo. São usadas duas matrizes diferentes que correspondem à matriz de transformação da câmara Pan-Tilt (Igreja de Vera Cruz e Capela da N^a Sra das Febres) e à câmara estéreo (Oficina do DEM).

III.9. Resultados

Nos resultados obtidos (Fig. III.17, Fig. III.18) observa-se que existem áreas que não possuem textura pois a área obtida através da câmara Pan-tilt é muito inferior à área de aquisição do laser 3D. Na Fig. III.19 essa diferença não é notória porque para a textura foi usada uma imagem da câmara estéreo, que abrange uma área maior, quase coincidente com a área de visão do laser 3D.



Fig. III.17 - Aplicação de Textura ao modelo da Igreja de Vera Cruz.

À esquerda a Textura aplicada à triangulação da Igreja de Vera Cruz. À direita a imagem da câmara Pan&Tilt usada para aplicação da textura.

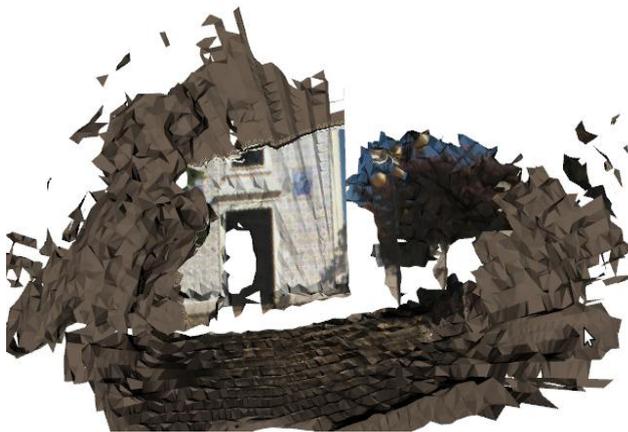


Fig. III.18 - Aplicação de Textura ao modelo da Capela da Nª Sra. das Febres.

À esquerda a Textura aplicada á triangulação da Capela da Nª Sra. das Febres. À direita a imagem da câmara Pan&Tilt usada para aplicação da textura.



Fig. III.19 - Aplicação de Textura ao modelo da oficina do DEM.

À esquerda a Textura aplicada á triangulação da oficina do DEM. À direita a imagem da câmara stereo usada para aplicação da textura.

III.10. Conclusões

Os resultados apresentados de textura assentam sobre triangulações resultantes de um só laser, neste caso o laser 3D. Os modelos apresentam textura parcial do modelo triangulado porque o laser 3D abrange uma maior área do que a das câmaras que fornecem as imagens, mais notório nos exemplos da igreja de Vera Cruz e a capela da N^a Sra. das Febres. Uma forma de resolver este problema é com recurso a várias imagens provenientes de diferentes câmaras, envolvendo uma calibração de câmaras em cada nova posição. Como não é o objectivo desta tese a fusão de diversas imagem para obter mapas de textura completos, focámo-nos num primeiro passo no desenvolvimento de uma plataforma que possa suportar este tipo de processamento. A calibração automática de câmaras do AtlasCar e fusão das diferentes imagens digitais fornecidas por estas será um trabalho futuro.

Uma outra solução para obter um modelo mais completo é a fusão de dados vindos de diferentes lasers, por exemplo do laser 3D para fornecer dados da zona de visão do condutor e os lasers planares para fornecer apoio ao estudo de obstáculos na zona lateral e posterior do AtlasCar. O problema em acumular pontos de diversos lasers reside no facto de que se estes não estiverem alinhados, a triangulação resultante poderá não fazer sentido sem uma calibração adequada. Existem áreas comuns aos lasers adquiridos mas se estes aparecem desalinhados, provocam erros na interpretação do modelo final

Actualmente, os parâmetros extrínsecos das matrizes de transformação aplicadas aos lasers são obtidos manualmente, com medições e cálculos morosos que tornam o processo de calibração e de obtenção de matrizes difícil de concretizar.

É neste contexto que a calibração dos lasers desempenha um papel importante pois influencia directamente a precisão dos resultados. O processo actual é susceptível a erros e sempre que um laser é deslocado, seja por acção voluntária ou involuntária é necessário reiniciar todo o processo de calibração manual e o mesmo se aplica na adição de novos lasers ao AtlasCar.

Sempre que é feita uma deslocação com o AtlasCar, as irregularidades do piso da estrada podem deslocar os lasers e comprometer a precisão dos dados lidos e obriga a uma recalibração dos lasers. Para corrigir esta situação, é proposto um método de calibração de um laser 2D relativamente ao laser 3D, descrito no capítulo seguinte.

IV. Calibração de lasers no AtlasCar

A calibração dos lasers do AtlasCar é um processo importante para obter resultados precisos e sobretudo na análise dos dados quando os mesmos são sobrepostos. Inicialmente, cada laser está no seu centro de coordenadas e a sobreposição dos dados não produz quaisquer resultados conclusivos quanto à geometria.

Na Fig. IV.1(a) estão representados dois lasers (Laser 1 e Laser 2), colocados em duas posições diferentes, adquirindo pontos 2D relativo ao seu próprio referencial, neste caso a azul os pontos adquiridos pelo Laser 1 e a vermelho os pontos adquiridos pelo Laser 2. A intenção final é sobrepor os dados vindos dos dois lasers para reconstruir parte do modelo inicial. Na Fig. IV.1 (b) está o resultado da sobreposição dos dados sem qualquer calibração prévia. Este exemplo ilustra a problemática da sobreposição de dados sem calibração. O resultado da sobreposição não permite reconstruir o modelo inicial.

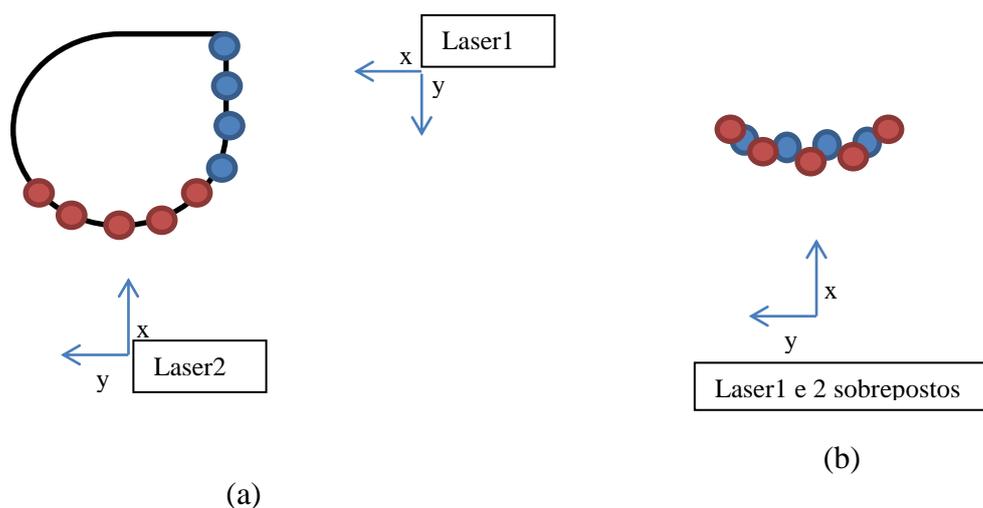


Fig. IV.1 - Aquisição de dois lasers 2D não calibrados

O objectivo deste capítulo é criar um método mais simples de calibração de um laser 2D relativamente a um laser 3D. O método de calibração proposto é um primeiro passo no processo de calibração de todos os sensores e câmaras do AtlasCar. É sugerido uma abordagem teórica e posterior validação com dados reais fornecidos pelos lasers do veículo. Para efectuar a calibração, foi desenvolvida uma aplicação em C/C++, recorrendo também à plataforma VTK e à biblioteca do OpenCV.

A periódica manutenção ao AtlasCar, competições de automação e outros projectos que incidem sobre o veículo limitam a sua disponibilidade para constantes aquisições, tendo em conta o tempo necessário em cada aquisição. Para fazer face a essa limitação, foi desenvolvido um simulador que produz os mesmos dados que os lasers do AtlasCar, ou seja, simulam a aquisição de nuvens de pontos de acordo com parâmetros configuráveis, como a localização dos lasers, orientação e a cena a adquirir.

Esse simulador é integrado na aplicação de calibração em que dependendo do tipo de dados a usar (simulados ou reais do AtlasCar), faz uso ou não do módulo de simulação. Esta arquitectura permite uma rápida mudança do contexto teórico para o contexto real sem alteração da sua estrutura funcional.

O módulo de simulação de nuvens de pontos adquire dados de um laser 3D e um laser 2D. Para a calibração, o laser de referência é o laser 3D e o laser a calibrar será o laser 2D. No caso real, os dados devolvidos pelo módulo de simulação serão substituídos pelo laser 3D do AtlasCar e o laser 2D (lasers planares).

Para efectuar a calibração entre lasers, a estratégia consiste em identificar uma geometria no ambiente capturado comum aos diferentes lasers para que se possa garantir o alinhamento correcto entre eles. Tomando como exemplo a Fig. IV.2(a) e (b), estão presentes dois lasers planares sem qualquer calibração, cada um no seu sistema de coordenadas. Visualmente, para sobrepor as duas aquisições, é necessário identificar uma zona comum aos dois lasers que permita o seu alinhamento. Na Fig. IV.2 (c) estão presentes os dois lasers calibrados e está identificada a zona comum aos dois que permitiu alinhá-los.

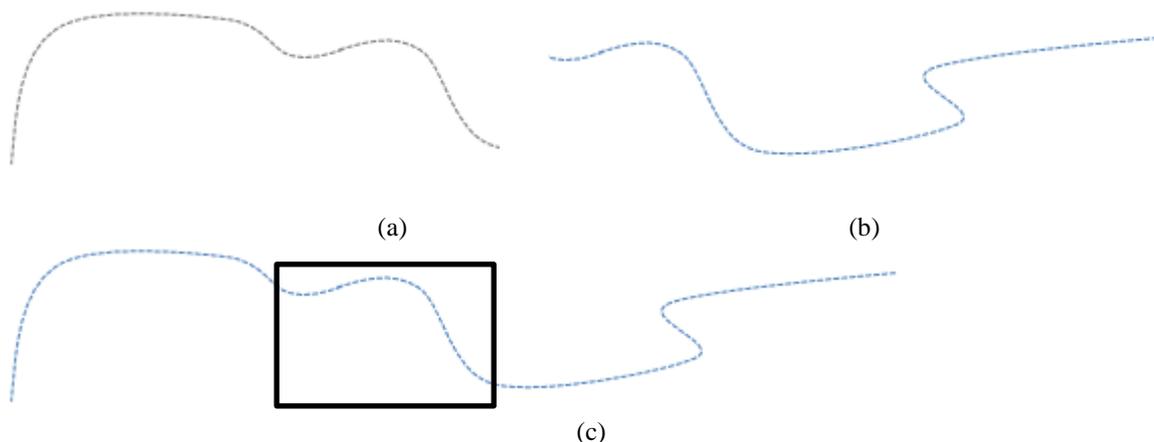


Fig. IV.2 - Calibração de dois lasers planares

(a) e (b) - estão dois conjuntos de dados de dois lasers planares representados no mesmo referencial mas descalibrados. (c) estão os mesmos dados laser mas após a aplicação de uma calibração com identificação da zona comum aos dois lasers que os permitiu alinhar.

No caso em estudo, como são usados um laser 3D e um laser 2D, acresce a dificuldade em determinar uma zona comum e que identifique univocamente a posição e orientação, principalmente do laser 2D. A forma adoptada para resolver esse problema passou pelo uso de um objecto colocado na cena, adquirido pelos lasers e que permitirá identificar a zona em comum e efectuar as transformações necessárias ao alinhamento. Esse objecto é chamado de objecto de calibração e está descrito na secção V.3.

O processo de calibração engloba duas fases em que é efectuado um processamento dos dados de ambos os lasers (3D e 2D). A Fig. IV.3 ilustra o processo geral de calibração onde cada fase possui um processo de identificação do(s) objecto(s) de calibração. O processo de identificação num caso e noutro são diferentes, dado também à natureza dos dados (3D e 2D) e o estudo da geometria é mais exigente em dados 2D pela falta de informação que estes fornecem de geometria comparativamente a um laser 3D.

Com o processo de identificação do(s) objecto(s) de calibração terminado, estão identificadas as zonas comuns aos dois lasers e o passo seguinte é o alinhamento desses dados. Esse alinhamento é terminado quando o erro da sobreposição das áreas comuns for mínimo, segundo o método dos mínimos quadrados. Com este resultado é devolvido uma matriz de transformação, que aplicada aos dados do laser 2D transformam o seu sistema de coordenadas para o sistema de coordenadas do laser 3D, estando assim o processo de calibração terminado. Este processo está descrito em detalhe na secção V.5.

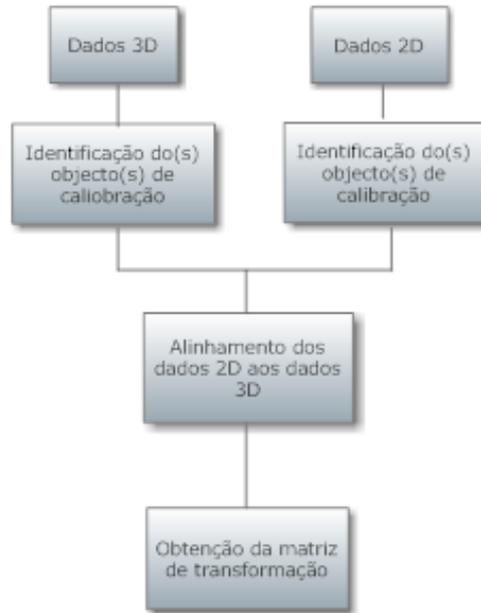


Fig. IV.3 - Diagrama de calibração

Descreve-se de seguida o diagrama estrutural da solução, os passos necessários à calibração e métodos usados para alcançar os objectivos.

IV.1. Diagrama funcional da solução

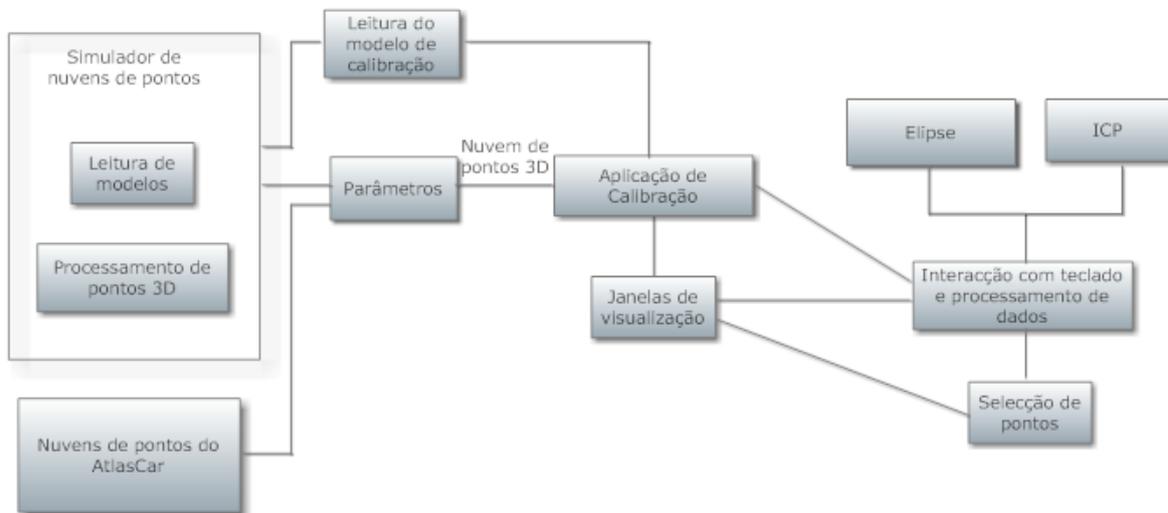


Fig. IV.4 - Diagrama funcional da aplicação

Na Fig. IV.4 está a organização da aplicação de calibração proposta. Como já foi referido, a aplicação está preparada para tratar dados obtidos pelo “Simulador de nuvens de pontos” ou dados vindos directamente do AtlasCar. O simulador possui um módulo de “Leitura de modelos” responsável pela leitura de modelos de superfície guardados localmente para uma estrutura VTK e

agrupa-o ao modelo fornecido inicialmente (Fig. IV.5(a)). O outro módulo de “Processamento de pontos 3D” é responsável por converter os modelos lidos em nuvens de pontos (Fig. IV.5 (b)).

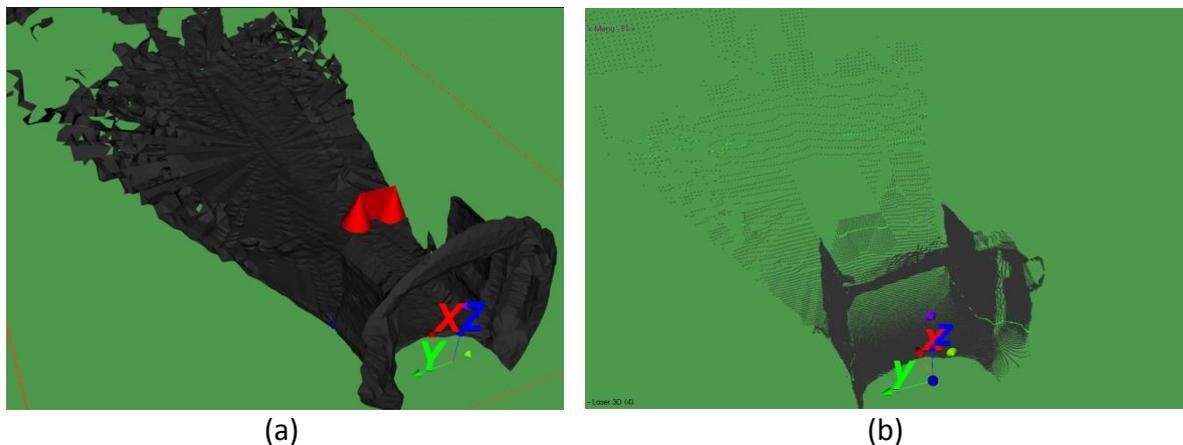


Fig. IV.5 - Nuvem de pontos resultante do modelo inicial

Em (a) está presente o modelo de superfície fornecido à aplicação com o objecto de calibração já adicionado a vermelho. Em (b) está a nuvem de pontos resultante (a preto o laser 3D e a verde claro o laser 2D).

O módulo de “Leitura do modelo da calibração” é responsável pela leitura do modelo do objecto de calibração que será usado no processo de calibração, quer seja para dados vindos do simulador ou do AtlasCar, daí a razão de este módulo estar separado do simulador. O simulador tem acesso a este módulo para leitura do objecto de calibração para geração da nuvem de pontos também desse objecto. Em dados reais, a nuvem de pontos do objecto de calibração já se encontra presente, necessitando apenas a leitura dos parâmetros do objecto de calibração armazenados em ficheiro por parte da aplicação de calibração.

O módulo “Parâmetros” depende dos parâmetros de entrada da aplicação fornecidos que informarão o tipo de dados a obter: simulados ou reais. Após a obtenção da nuvem de pontos, os dados são devolvidos à “Aplicação de calibração”, tornando transparente a origem dos dados (simulados ou reais).

O módulo “Interacção com teclado e processamento de dados” é responsável pela interacção do utilizador com a aplicação através do teclado. Este módulo inclui métodos e estruturas de dados usadas, estando algumas delas implementadas em dois módulos: “ICP”, onde está implementado um algoritmo de processamento e métodos de tratamento de dados 3D; e “Elipse”, onde estão implementados algoritmos de reconstrução, estruturas de armazenamento dos resultados e métodos de apoio e acesso a essas estruturas. Além destes dois módulos, é acedido também o módulo “Seleção de pontos” que fornece dados de selecção obtidos pela interacção com o rato para processamento.

A “Aplicação de Calibração” é responsável pela inicialização dos dados, janelas de visualização e dos módulos de interacção do utilizador (“Seleção de pontos” e “Interacção com teclado e processamento de dados”).

IV.2. Objecto de calibração

O objecto de calibração usado no simulador possui a geometria ilustrada na Fig. IV.6. Os parâmetros do objecto podem ser alterados no ficheiro especificado na secção IV.3.1 mas a estrutura base de dois cones e uma placa intermédia mantém-se inalterada. A razão da escolha do cone reside no facto de o corte a 2D obtido pelo laser corresponder a figuras geométricas conhecidas, neste caso a uma elipse. Este facto retira a ambiguidade em termos da altura no cone pois a elipse apenas possui uma forma de encaixe. A adição do segundo cone retira a ambiguidade em termos da orientação dos cones. A placa intermédia permite manter os cones a uma distância fixa e facilita o cálculo da orientação do objecto de calibração na cena e poderá no futuro levar informação adicional para calibração de câmaras como padrões *chessboard*.



Fig. IV.6 - Objecto de calibração virtual

No laser 3D, a nuvem de pontos obtida corresponde no máximo a metade do objecto de calibração e a sua identificação é mais simples do que em dados 2D. Num laser 2D, a nuvem resultante contém uma geometria mais específica do que com dados 3D. Assumindo que um laser 2D se comporta como um plano, a intercepção desse plano com o cone resulta num de quatro casos: círculo, elipse, parábola ou hipérbole.

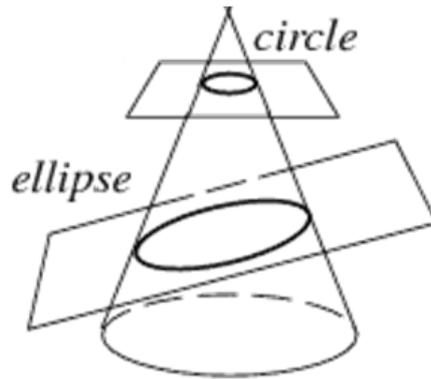


Fig. IV.7 - Casos de estudo da intercepção de um plano no cone

Os casos como a hipérbole e a parábola não são contemplados pois assume-se que o processo de aquisição irá garantir que o laser 2D intersecta o objecto de calibração num ângulo e altura suficientes para obter uma elipse ou um círculo (Fig. IV.7).

IV.3. Simulador de nuvens de pontos

No simulador, o modelo inicial é lido, é acrescentado o objecto de calibração à cena e posteriormente de acordo com os parâmetros definidos para os lasers, é feita uma aquisição. Para a aquisição, os lasers estão localizados no mesmo referencial de coordenadas. No final da aquisição, é efectuada uma transformação aos dados do laser 2D para que estes sejam mostrados numa nova janela de visualização e aparente estar sem nenhuma calibração inicial, ou seja, que a posição do laser 2D esteja localizada em $(0,0,0)$ com vector director na direcção do eixo X.

Esta transformação corresponde à aplicação inversa dos parâmetros de posição e orientação definidos nos parâmetros do laser aos dados, ou seja, é criada uma matriz de transformação em que a translação corresponde a $(-x,-y,-z)$ em que (x,y,z) corresponde á posição do laser e a rotação corresponde à inversa da rotação aplicada na orientação do laser. Desta forma, cada um dos lasers está no seu próprio sistema de coordenadas, centrado nele próprio.

Para facilitar o processo visual, o laser de referência e o laser a calibrar estão em janelas de visualização diferentes, cada um no seu referencial, como descrito anteriormente. Assim os processos necessários à calibração ficam separados e é possível visualizar o que seria a calibração ideal do laser 2D directamente na janela do laser de referência. Esta opção permite calcular e visualizar no final da calibração, se o processo aqui proposto é eficaz e quantificar a precisão. Este parâmetro corresponde ao cálculo da distância média entre os pontos 2D ideais e os correspondentes pontos 2D obtidos através do processo de calibração.

O processo propriamente dito de aquisição da nuvem de pontos em ambos os lasers consiste na intercepção de linhas entre a posição do laser e pontos que dependem do ângulo de abertura, da resolução angular e do alcance do laser e são percorridos todos os pontos entre esse intervalo. O ponto resultante dessa intercepção é guardado até percorrer todos os pontos do modelo. Está exemplificado na Fig. IV.8 um exemplo em 2D do processo de intercepção de linhas com o modelo para obter a nuvem de pontos. Verifica-se o caso em que o alcance do laser não é suficiente para haver uma intercepção com o modelo.

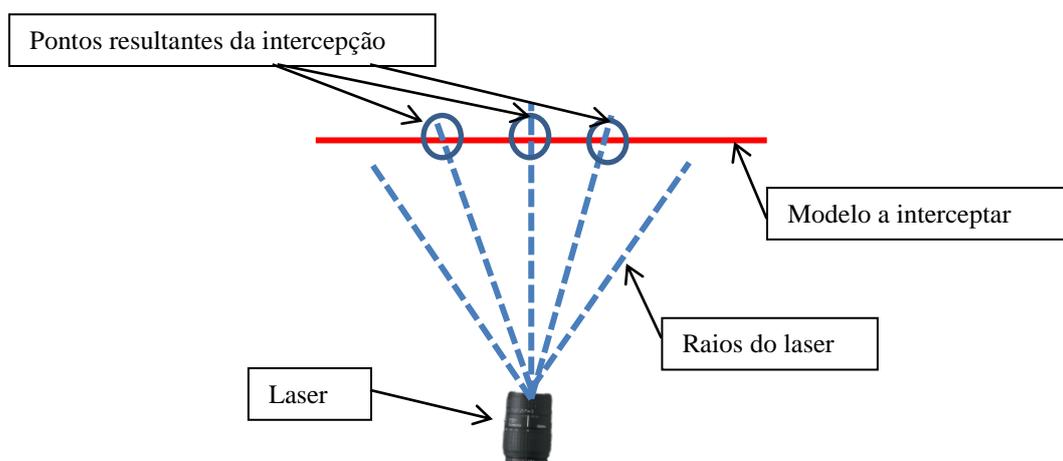


Fig. IV.8 - Processo de aquisição 2D de pontos no simulador

IV.3.1. Parâmetros do Simulador

Os parâmetros para a geração de uma nuvens de pontos 3D simuladas são:

- Um Ficheiro que contém uma cena 3D, podendo este ficheiro ter o formato *.obj, *.ply ou *.asc. O caso específico em que o formato do ficheiro de entrada for *.asc, este tipo de organização de dados não possui informação de superfícies mas apenas uma lista de coordenadas no espaço tridimensional. Deste modo, a leitura dos pontos resulta numa nuvem de pontos, não sendo necessária realizar uma aquisição de pontos. No entanto, é adquirida apenas uma nuvem de pontos com o(s) objecto(s) de calibração, sendo a nuvem de pontos resultante sobreposta sobre a nuvem de pontos inicial *.asc.
- Parâmetros para ambos os lasers:
 - Posição do laser (x,y,z);
 - Orientação (Ângulos de Euler (vector));
 - Ângulo de abertura;

- Resolução angular;
- Alcance (m).

Estes parâmetros estão ilustrados na Fig. IV.9, num exemplo em 2D. No caso do laser 3D, a única diferença nos parâmetros está na necessidade de especificar dois ângulos de abertura pois é efectuado um varrimento da esquerda para a direita e de cima para baixo.

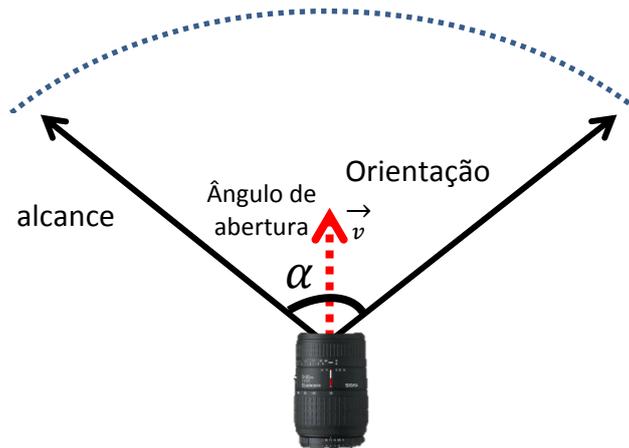


Fig. IV.9 - Parâmetros dos lasers

Para os dados simulados ou reais é necessária a existência de um ficheiro no mesmo directório da aplicação com nome “CalibrationModel.txt” que possui os parâmetros do objecto de calibração, entre eles:

- Altura do cone;
- Raio do cone;
- Distância entre os centros dos dois cones;
- Centro do objecto de calibração;
- Rotação do objecto na cena

Na Fig. IV.10 está um exemplo de um ficheiro com as configurações do objecto de calibração.

```

1 height
2 0.924
3 radius
4 0.3825
5 center
6 5 -1 0.2
7 direction
8 1 0 0
9 distanceCones
10 1
11 rotation
12 0 0 0

```

Fig. IV.10 - Exemplo de um ficheiro de configuração do objecto de calibração

Caso um dos parâmetros do simulador referente aos lasers não esteja definido, serão usados os valores predefinidos apresentados na Tabela IV.1.

Laser 3D	Laser 2D
Posição: (0, 0, 1.5)	Posição: (0, 0.5, 0.5)
Ângulos de Euler: (0, 0, 0)	Ângulos de Euler: (0,0,0)
Ângulo de abertura: (180,180)	Ângulo de abertura: 180
Resolução angular: 0.5	Resolução angular: 0.5
Alcance: 30	Alcance: 30

Tabela IV.1 - Valores predefinidos dos parâmetros dos lasers

Orientação

Tal como descrito, um dos parâmetros de entrada da aplicação é a orientação do laser dada em ângulos de Euler. Estes ângulos correspondem a um vector com ângulos de rotação em x, y e z relativo ao centro de coordenadas do laser. No entanto existem variantes e é necessário definir qual usar pois não estão definidas regras para a sua aplicação. Descreve-se de seguida as variantes existentes e as escolhas.

Tipos de ângulos de Euler(Baker, 2011):

Existem diversas combinações de rotações diferentes, sendo a ordem importante. Torna-se portanto importante escolher:

- Ordem de combinação de ângulos;
- Rotação relativa ao objecto ou coordenadas absolutas;

Ordem de combinação de ângulos:

Podem ser aplicadas transformações 3D na ordem xyz , yzx , zxy ou invertendo a ordem zyx , xzy , yxz , que dá seis permutações. Existem mais porque os ângulos não são independentes, ou seja, aplicando uma rotação de 90° em torno de x , seguido de 90° em torno de y e de volta -90° em x , é a mesma coisa aplicando 90° em torno de z . Significa que é possível fazer combinações de rotações 3D em apenas dois planos, podendo portanto usar as sequências xyx , xzx , yxy , zyz , zxz e zyz . Existem doze possibilidades de sequências de ângulos possíveis. Na aplicação, a combinação de ângulos usada é xyz .

Rotação relativa ao objecto ou coordenadas absolutas

É possível aplicar ângulos relativos ao eixo de coordenadas de referência, o que aumentaria o número de combinações de doze para catorze. No entanto, torna-se muito mais simples aplicar as transformações ao sistema de coordenadas do próprio objecto. Foi escolhido a aplicação do segundo caso.

IV.3.2. Aquisição de dados com um laser 2D

Os pontos que serão usados para a intercepção com o modelo, são inicialmente definidos num sistema de coordenadas em que a posição do laser está localizada na origem e orientado no eixo x . De seguida são calculados os pontos de acordo com os parâmetros de ângulo de abertura, resolução angular e alcance (Fig. IV.11).

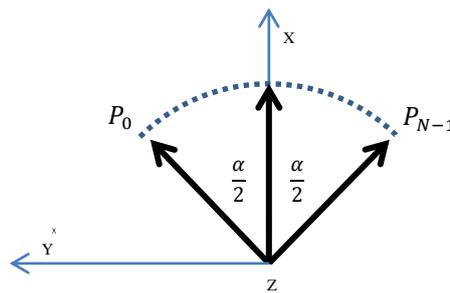


Fig. IV.11 - Parâmetros de aquisição do laser 2D

Sendo $\alpha = \text{ângulo de abertura}$ e o passo entre cada valor de θ igual à resolução angular

$$x = alcance * \cos(\theta), \theta \in \left[-\frac{\alpha}{2}, \frac{\alpha}{2}\right]$$

$$y = alcance * \sin(\theta), \theta \in \left[-\frac{\alpha}{2}, \frac{\alpha}{2}\right]$$

$$z = 0$$

São obtidos no total $\left(\frac{\alpha}{\text{resolução angular}}\right) + 1$ pontos.

Por fim é aplicada a matriz de transformação aos pontos resultantes. A matriz de transformação criada, tal como foi descrito na secção IV.1, é subdividida em rotação e translação. A matriz 3x3 de rotação é obtida da aplicação do método `cvRodrigues2` disponibilizada pelo OpenCV que tem como parâmetro de entrada um vector de rotação 1x3 que corresponde aos ângulos de Euler dados pelos parâmetros iniciais do simulador. O vector de translação é dado pela posição do laser também introduzido pelos parâmetros. Aplicando a matriz resultante aos pontos, obtém-se a posição dos pontos que serão adquiridos pelo laser de acordo com a posição e orientação deste.

O simulador utiliza a classe `vtkCellLocator` disponibilizada pela ferramenta VTK que retorna o ponto resultante da intercepção de um segmento de recta com um modelo. É usada para obter todos os pontos dos raios correspondentes ao laser com o modelo, que dão origem aos pontos simulados de aquisição (ver Fig. IV.8 para um exemplo).

IV.3.3. Aquisição de dados com o laser 3D

O processo usado para o laser 3D não difere da metodologia descrita na secção anterior. São aplicadas as mesmas transformações e métodos de aquisição dos pontos de simulação, de acordo com os parâmetros introduzidos. O que difere são as equações usadas para cálculo dos pontos iniciais antes de aplicar as transformações. Na Fig. IV.12 estão representados os parâmetros dos ângulos de abertura do laser 3D expressos por θ e φ .

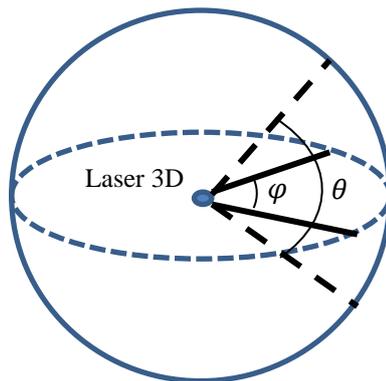


Fig. IV.12 - Dados de aquisição do laser 3D simulado

As equações são expressas da seguinte forma:

$$x = alcance * \sin(\theta) * \cos(\varphi)$$

$$y = alcance * \sin(\theta) * \sin(\varphi)$$

$$z = alcance * \cos(\theta)$$

Onde o ângulo de abertura está compreendido entre: $0 \leq \theta \leq \pi$ e $0 \leq \varphi \leq 2\pi$.

No total são adquiridos $\left(\frac{\theta * \varphi}{resolução\ angular}\right) + 2$ pontos.

Obtidos os pontos através das equações, tal como é efectuado na secção anterior, é aplicada uma matriz de transformação de acordo com os parâmetros do laser introduzidos e posteriormente com o uso da estrutura vtkCellLocator são calculados os pontos de intercepção com o modelo.

IV.3.4. Resultados

Na Fig. IV.13 estão os resultados através do módulo de simulação usando como modelos iniciais os modelos triangulados (com o BPA) das aquisições efectuadas com o AtlasCar. A preto está o resultado do laser 3D e a linha verde é o resultado do laser 2D. Neste exemplo os dois lasers representados estão no mesmo sistema de coordenadas e representam aquisições ideais e com calibrações ideais. O laser 2D presente servirá para obter um cálculo da precisão do método de calibração proposto. Na secção IV.4 é explicado o processo de calibração proposto e o resultado da calibração deverá coincidir com a linha representada a verde, ou seja, quando menor for o erro entre a linha a verde e o resultado da calibração, maior é o grau de precisão do método.

O sistema de coordenadas presente é o usado pelo laser 3D. Como se trata de um ambiente simulado, os erros subjacentes aos lasers podem ser ou não tidos em conta.

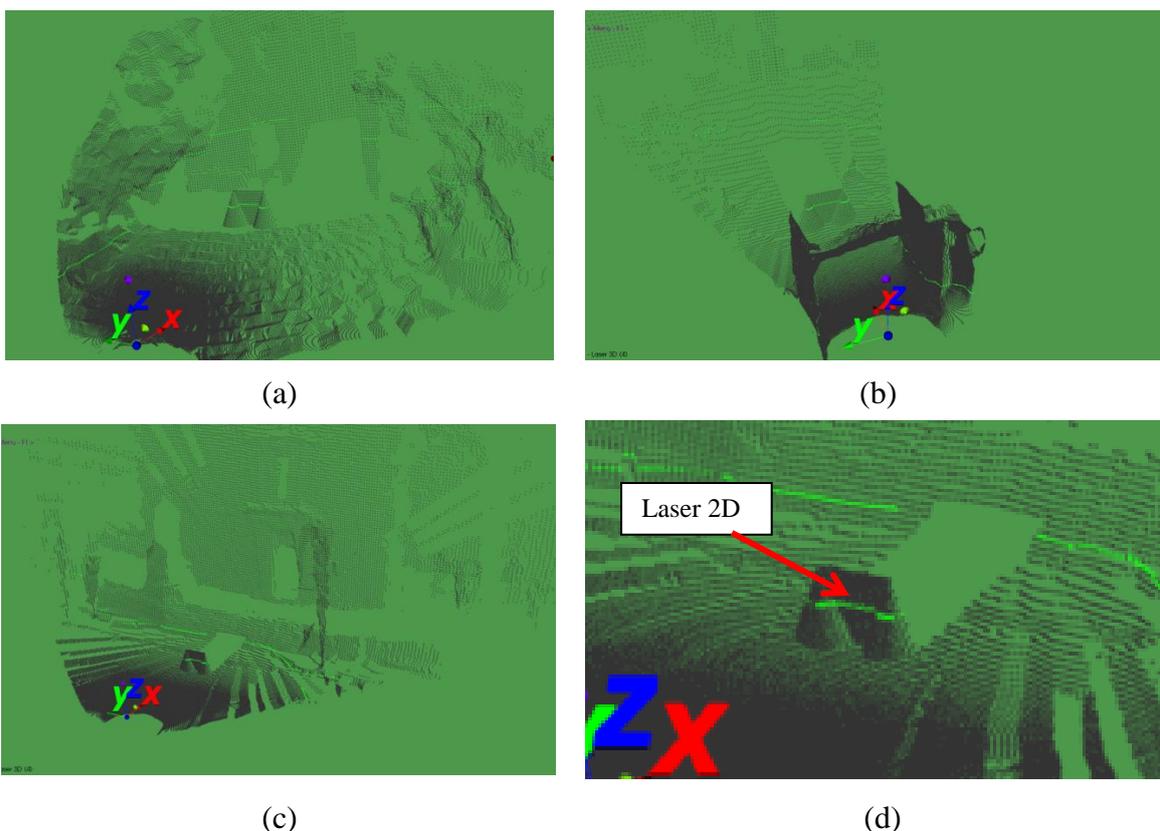


Fig. IV.13 - Capturas simuladas em diversos ambientes com o objecto de calibração.
 (a) – Capela da N.ª Sra. das Febres; (b) – oficina do DEM; (c) – Capela de Vera Cruz; (d) – pormenor do objecto de calibração com a nuvem de pontos 3D e 2D sobreposta.

IV.4. Processo de calibração do laser 2D

Tendo as nuvens de pontos de ambos os lasers, o próximo passo é a calibração. É usado o objecto de calibração ideal descrito anteriormente para fazer um encaixe (*fitting*) aos pontos 3D. Este *fitting* fará com que os erros do laser 3D sejam minimizados e qualquer operação de calibração será com base no resultado desta operação, sobre o objecto de calibração.

Aos dados do laser 2D é efectuada uma reconstrução de uma elipse a partir da secção obtida na aquisição, aos quais estão associados erros do próprio laser, à semelhança do que acontece com o laser 3D. A reconstrução da elipse permitirá minimizar o impacto do erro nos dados.

Descreve-se nas seguintes secções deste capítulo como a detecção do objecto é feita nos dados do laser 3D e qual o algoritmo aplicado para realizar o *fitting*; e no laser 2D como são detectadas as elipses, o algoritmo de reconstrução usado e como calibrar os dados 2D aos dados 3D.

A aplicação de calibração desenvolvida permite a presença de N objectos de calibração na cena. Em ambiente simulado, são sobrepostos N objectos de calibração na cena. O número de objectos a adicionar é definido nos parâmetros de entrada da aplicação, incluindo um ficheiro por cada objecto de calibração onde estão contidos os parâmetros dos mesmos. Em dados reais a nuvem de pontos fornecida ao módulo de calibração deverá conter os diversos objectos de calibração. A aplicação em dados reais é estudada em detalhe na secção IV.5.

Na secção IV.4.3 é analisado a eficiência da calibração com dados simulados, variando o número de objectos de calibração usados na calibração.

IV.4.1. Detecção do objecto de calibração no modelo 3D

O processo proposto é semi-automático, tendo o utilizador que seleccionar um ponto no centro do objecto de calibração e outro ponto no mesmo plano em que o primeiro se encontra. Tomando como exemplo a Fig. IV.14, o ponto a vermelho será o centro aproximado dado pelo utilizador e o ponto azul está presente para dar a direcção do objecto segundo o vector desenhado na imagem, baseado na placa intermédia que une os cones.

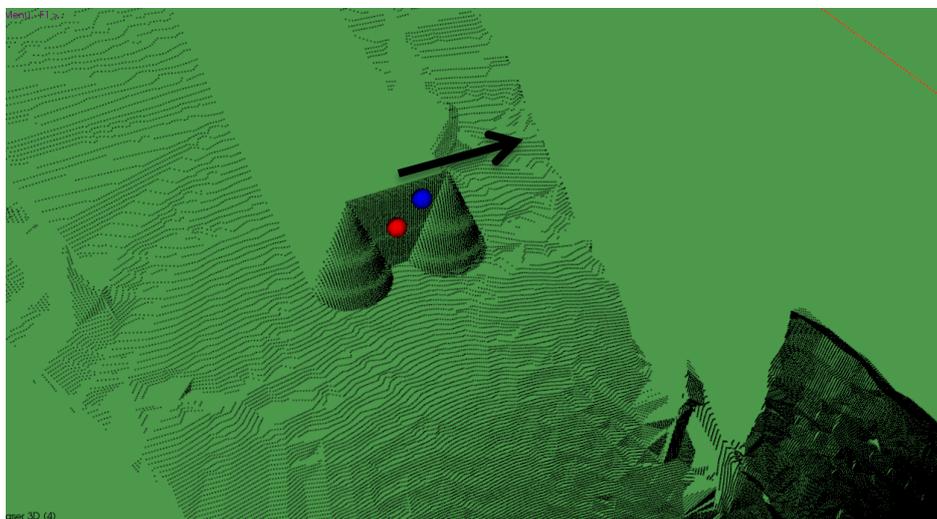


Fig. IV.14 - Detecção do objecto de calibração

A posição central é aproximada pois é apenas usado como uma aproximação inicial para o processo seguinte. O segundo ponto (azul) define em conjunto com o primeiro a orientação em torno do eixo Z. Não é considerada a rotação em torno dos outros eixos para evitar a selecção de mais pontos no

modelo, que com a presença de erros dos lasers torna a tarefa mais difícil. Assume-se que o objecto de calibração se encontra numa zona plana ou com pouco declive.

Escolhidos os pontos, os dados 3D são filtrados: conhecendo o tamanho do objecto de calibração, pode-se limitar a quantidade de dados a serem usados pelo algoritmo (Fig. IV.15), eliminando zonas do modelo completo que possam interferir com a precisão do *fitting*.

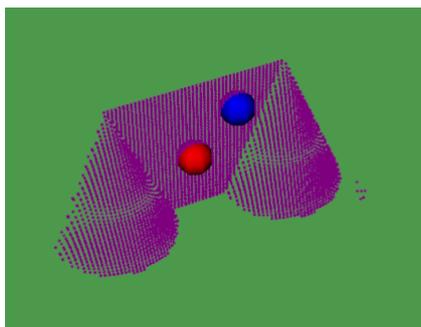


Fig. IV.15 - Dados do objecto de calibração filtrados

Feita a filtragem, é aplicado o algoritmo Iterative Closest Point (ICP) (Besl and McKay, 1992) que efectua o fitting do modelo aos pontos.

O ICP é um algoritmo iterativo e é aplicado em 3 fases principais:

- Estabelecer uma correspondência entre pares de pontos entre os dois modelos baseada na sua proximidade.
- Estimar uma transformação de corpo rígido que minimize a distância entre os pares de pontos.
- Aplicar a transformação ao primeiro modelo.

Estes passos são repetidos até o algoritmo convergir. No entanto o algoritmo pode convergir para mínimos locais, pelo que serão atingidos melhores resultados dando uma boa aproximação inicial.

A ferramenta VTK disponibiliza a implementação do ICP com alguns parâmetros que permitem definir a eficiência computacional e a convergência do algoritmo.

O algoritmo descrito acima exige um grande esforço computacional para determinar o ponto mais próximo de um ponto dado, que é influenciado pelo tamanho do modelo, que no caso em estudo será de aproximadamente ~80 000 pontos. Para contornar este problema, é possível definir o número máximo de pontos que o algoritmo usará no cálculo do vizinho mais próximo. Este valor é definido em “setMaxLandmarks” e por omissão está configurado a 200.

A alteração que este parâmetro introduz só reduz o número de pontos usados no algoritmo. Em vez de usar todos os pontos do modelo, é definido um passo. Este passo é dado pela divisão inteira do número de pontos do modelo pelo número máximo de landmarks definidos. Caso o número de pontos do primeiro modelo for inferior a esse limite, é usado um salto igual a 1.

Nos parâmetros do algoritmo são usados 700 landmarks em vez dos 200 predefinidos para que sejam usados no máximo 700 pontos na correspondência do algoritmo. Não são usados todos os pontos porque o algoritmo de detecção de um ponto adjacente é computacionalmente exigente e para minimizar esse efeito o VTK disponibiliza um método de definição desse valor. Foi definido esse valor máximo pois empiricamente não se verifica maior precisão aumentando esse valor e o tempo de execução com este número de landmarks é de entre dois e cinco segundos, sendo um tempo curto para os resultados obtidos. Contudo esse valor poderá ser ajustado com a utilização de outros lasers com parâmetros diferentes (resolução angular, precisão, etc.).

Como critério de paragem existem duas situações:

- O algoritmo atingiu o número máximo de iterações, que neste caso é de 150, por não se justificar ser maior até para o tempo de execução não ser elevado;
- O algoritmo atingiu a distância média mínima entre os pontos imposta, que empiricamente foi escolhido como sendo $1 * 10^{-5}$.

O método de transformação entre cada iteração é do tipo Transformação de Corpo Rígido pois o tamanho do objecto é conhecido e não se justifica o algoritmo alterar o tamanho do objecto de calibração, sendo apenas usadas rotações/translações. Como este algoritmo faz uso dos mínimos erros quadrados, espera-se que num ambiente real, os erros associados aos lasers possam ser minimizados.

Na aplicação, é possível visualizar o objecto de calibração na posição inicial dada pela escolha dos pontos e o resultado após a aplicação do algoritmo na Fig. IV.16. Observa-se o resultado da aplicação do algoritmo ICP aos pontos filtrados. O objecto de calibração está representado a azul e os pontos resultante da filtragem a roxo. Estão sinalizados *outliers* que acabaram por não ser filtrados mas que não influenciaram o resultado do algoritmo.

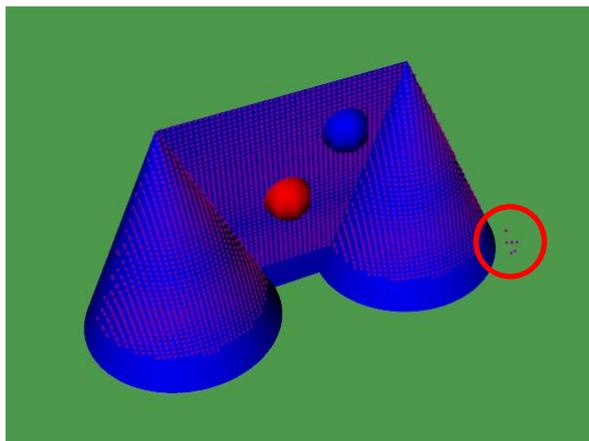


Fig. IV.16 - Resultado do fitting do objecto de calibração

IV.4.2. Processamento dos dados 2D

Devido às transformações aplicadas aos dados mencionados na secção IV.3, os dados do laser 2D estão presentes no plano $Z=0$. De uma forma mais detalhada, para a aquisição de dados do laser 2D, este laser e o laser 3D estão no mesmo referencial e inicialmente cada um está na origem do referencial. Tomando como um exemplo a definição da posição do laser 2D como $(0,1,1)$ e orientação $(10^\circ, 10^\circ, 10^\circ)$. Este sofre uma translação que corresponde aos parâmetros relativos à posição. De seguida, sofre uma rotação de 10° em X, 10° em Y e 10° em Z e é a partir desta posição final que é efectuada uma aquisição.

Após a aquisição, para colocar os dados no plano $Z=0$ realiza-se o processo inverso aos dados adquiridos: realiza-se uma rotação de -10° em X, -10° em Y e -10° em Z seguida de uma translação de 0 em X, -1 em Y e -1 em Z. Esta transformação é realizada para simular a descalibração do laser 2D, que passa a estar centrado num referencial próprio e os dados correspondem a dados 2D, visto que foi eliminada a componente Z.

IV.4.2.1. Localização e cálculo das elipses com base nos arcos obtidos da nuvem de pontos

É necessário efectuar uma selecção em cada um dos arcos, definindo a ordem como sendo primeiro o arco mais à esquerda do ponto de visualização, seguido do arco da direita.

A ordem da selecção é de acordo com o ilustrado na Fig. IV.17 para garantir que são usados o máximo de pontos para a criação das elipses para assegurar maior precisão no resultado. Assim são seleccionados todos os pontos entre os pontos de selecção, inclusive.

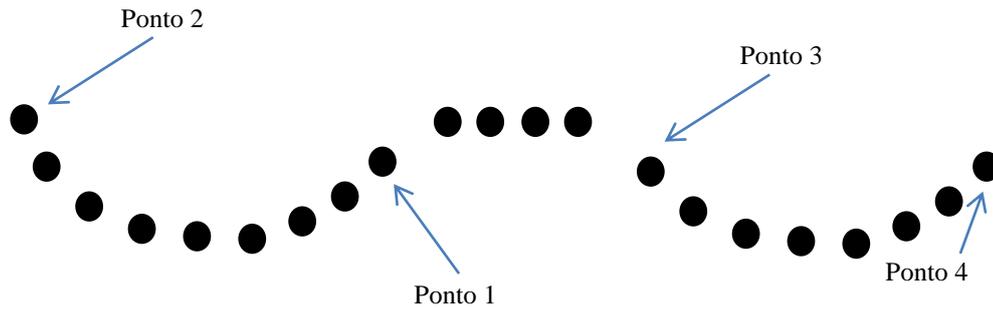


Fig. IV.17 - Sequência de selecção de pontos das elipses

O mesmo mecanismo é aplicado ao outro arco da elipse e o processo de cálculo da elipse é efectuado a cada conjunto de pontos em separado. Os pontos são obtidos através do estudo da vizinhança de pontos através da estrutura `vtkCellLocator` do VTK. Tomando a Fig. IV.17 como exemplo, ao seleccionar o ponto 1, este é tomado como o ponto de partida e são obtidos os N pontos mais próximos deste em que inicialmente $N=3$ (onde está incluído o próprio ponto) e é seleccionado o ponto cuja distância seja menor ao Ponto 2. Se não for encontrado é incrementado o valor de N até encontrar um ponto cuja distância seja menor. Ao encontrar um ponto, o valor inicial de N é repostado. O algoritmo termina quando atingir o Ponto 2. O processo para a outra elipse é análogo ao descrito anteriormente. No final obtêm-se um conjunto de pontos pertencentes a cada uma das elipses.

Ao conjunto de pontos pertencentes às elipses, é aplicado um algoritmo presente na biblioteca do OpenCV(`cvFitEllipse()`), em que dado um conjunto de pontos em (x,y) são devolvidos os parâmetros da elipse formada por esses pontos que minimiza o erro segundo os mínimos quadrados. O algoritmo pressupõe um mínimo de seis pontos para determinar a elipse.

Os parâmetros devolvidos são eles: centro da elipse, comprimento do eixo maior e menor e o ângulo formado entre o eixo maior e o eixo YY, no sistema de coordenadas do laser 2D.

O parâmetro do ângulo devolvido requer especial atenção pois não corresponde ao ângulo de rotação formado entre o eixo maior e o eixo formado pelo vector director do objecto de calibração.

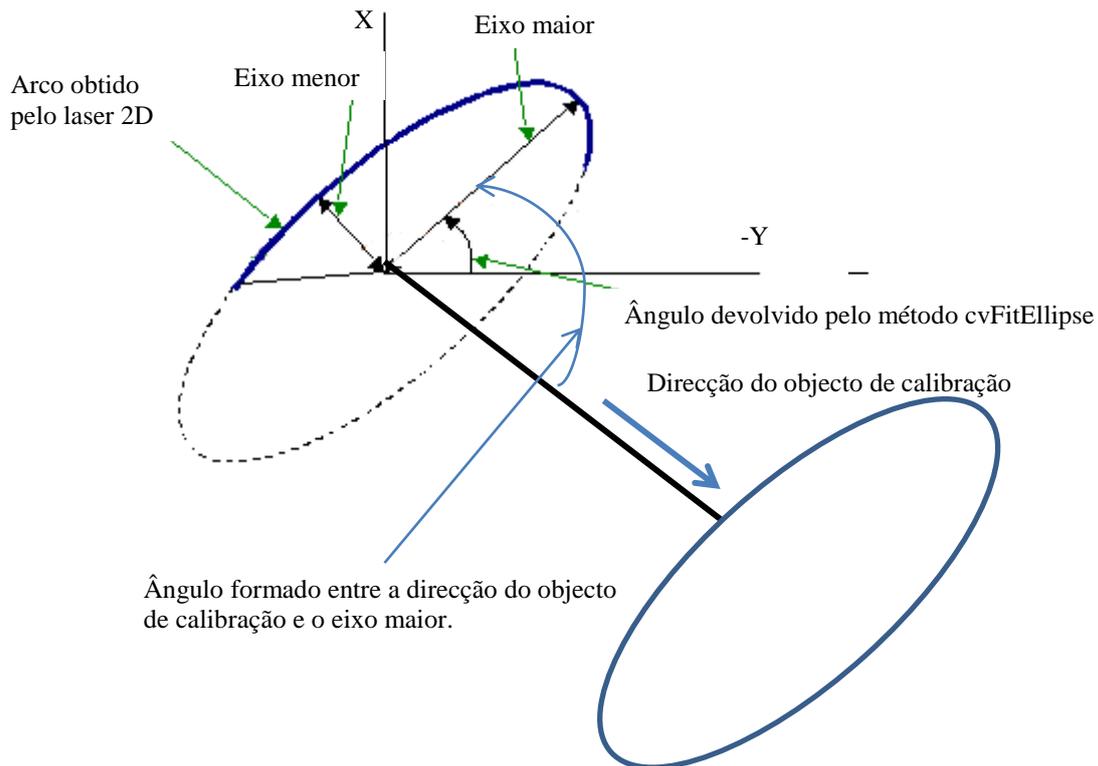


Fig. IV.18 - Esquema das elipses obtidas no laser 2D

Tomando como exemplo a Fig. IV.18 é possível observar de que é necessário ter em conta dois ângulos: ângulo entre o raio maior e o eixo -Y; e o ângulo formado entre o raio maior e a direcção do objecto de calibração. O ângulo final é dado pela diferença dos dois, contido entre $[0; \frac{\pi}{2}]$. Este processo permite obter o parâmetro da rotação da elipse, que será necessária no processo seguinte.

Para o cálculo da direcção do objecto de calibração, é usado o método *cvFitLine* fornecido pela biblioteca openCV. Este método tem como parâmetro de entrada um vector de pontos (que neste caso em particular são dados bidimensionais). O algoritmo devolve o vector que minimiza a distância dos pontos segundo o método dos mínimos quadrados. Este método permite obter o vector director na presença de dados com erros associados.

Os pontos necessários para obter o vector director do objecto de calibração são seleccionados automaticamente. Através da selecção anterior dos pontos das elipses ilustrado na Fig. IV.17, os pontos a usar para efectuar o *fitting* da linha corresponde aos pontos entre a selecção do Ponto 1 e Ponto 3. Os pontos são obtidos através do estudo da sua vizinhança através da estrutura *vtkCellLocator* do VTK, cujo algoritmo usado é análogo ao aplicado na selecção de pontos da elipse.

A partir dos parâmetros da elipse, são determinados oito pontos (quatro por elipse) localizados de acordo com a Fig. IV.19. Sendo os pontos definidos num espaço bidimensional, estes são obtidos da seguinte forma:

$$\begin{aligned}
 P_1 &= \left(C_x + \frac{Rx}{2} * \cos(\alpha), C_y + \frac{Ry}{2} * \sin(\alpha) \right) \\
 P_2 &= \left(C_x + \frac{Ry}{2} * \cos\left(\frac{\pi}{2} + \alpha\right), C_y + \frac{Rx}{2} * \sin\left(\frac{\pi}{2} + \alpha\right) \right) \\
 P_3 &= \left(C_x + \frac{Rx}{2} * \cos(\alpha), C_y + \frac{Ry}{2} * \sin(\alpha) \right) \\
 P_4 &= \left(C_x + \frac{Ry}{2} * \cos\left(\frac{\pi}{2} + \alpha\right), C_y + \frac{Rx}{2} * \sin\left(\frac{\pi}{2} + \alpha\right) \right)
 \end{aligned}$$

Onde Rx e Ry correspondem ao tamanho do eixo maior e menor respectivamente, Cx e Cy correspondem ao centro da elipse e α corresponde ao ângulo formado entre Rx e o vector director do objecto de calibração calculado anteriormente. Nos pontos P2 e P4 é adicionado $\frac{\pi}{2}$ ao ângulo porque aquele que é devolvido pelo algoritmo *cvFitEllipse* é relativo ao eixo X e no caso particular é pretendido o ângulo formado com o eixo Y.

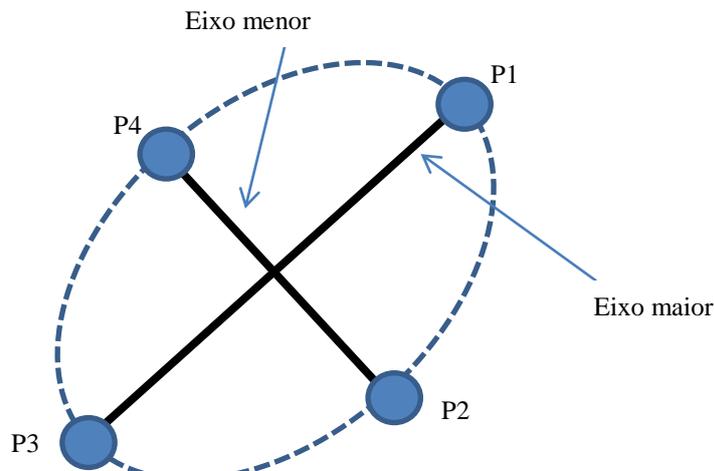


Fig. IV.19 - Obtenção dos pontos das extremidades dos eixos da elipse

IV.4.2.2. Fitting da elipse ao cone

Geometricamente, uma elipse possui apenas uma forma de encaixe num cone em termos de altura e inclinação. A única indeterminação reside no ângulo de rotação em torno do eixo vertical do cone, que é resolvida pelo ângulo devolvido no cálculo das elipses no ponto anterior.

Cada elipse é tratada individualmente, ou seja, para cada elipse é feito o cálculo do seu encaixe num cone que é igual aos cones usados no objecto de calibração. Efectuando um corte no cone

segundo a direcção do raio maior da elipse (Fig. IV.20) obtemos um triângulo e o eixo maior da elipse que resulta da projecção do cone em 2D. Deste corte podemos obter a altura e o ângulo de encaixe no cone.

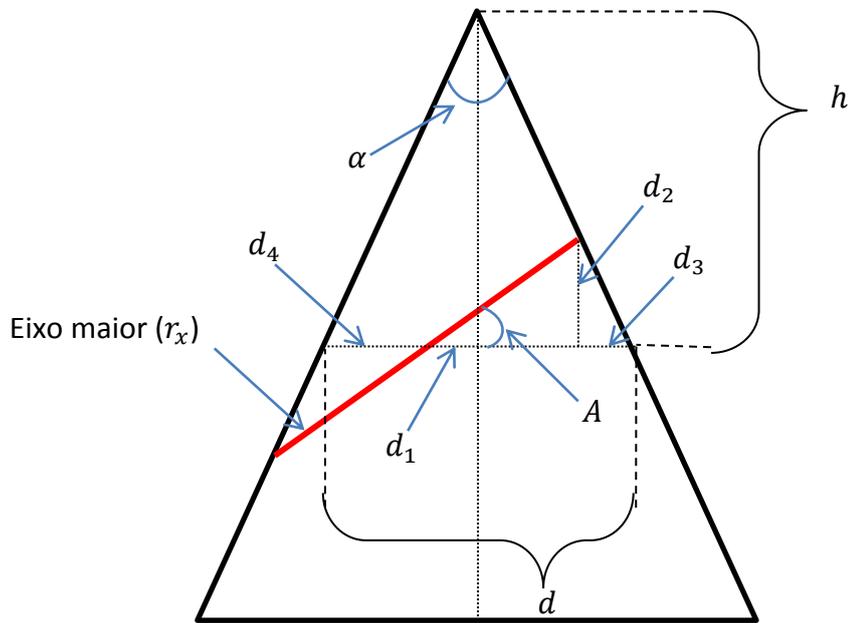


Fig. IV.20 - Corte do cone segundo o eixo maior

No entanto, é necessário relacionar o eixo maior com o eixo menor para determinar univocamente o encaixe da elipse no cone. Para tal é usada a elipse formada pelos parâmetros de eixo maior e menor projectada em duas coordenadas (Fig. IV.21). O que relaciona esta figura com a Fig. IV.20 é o valor d_1 , que tem que ter o mesmo valor. Este é um caso que se resolve algebricamente e que está descrito nos parágrafos seguintes.

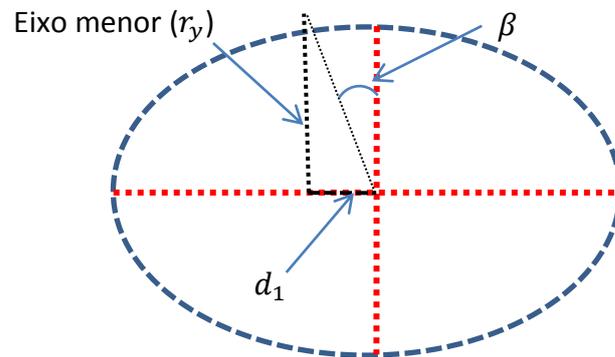


Fig. IV.21 - Elipse projectada em 2D

Podemos deduzir da Fig. IV.20:

$$\sin(A) = \frac{d_2}{\frac{r_x}{2}} \Leftrightarrow d_2 = \frac{r_x}{2} \sin(A) \quad (1)$$

$$\tan\left(\frac{\alpha}{2}\right) = \frac{d_3}{d_2} \Leftrightarrow d_3 = d_2 \tan\left(\frac{\alpha}{2}\right) \quad (2)$$

$$d_4 = \frac{d}{2} - d_3 \Leftrightarrow d_1 = \frac{d}{2} - d_4 \Leftrightarrow d_1 = \frac{d}{2} - \frac{d}{2} + d_3 \Leftrightarrow d_1 = d_3 \quad (3)$$

A partir de (2), substituindo com o valor de (3):

$$d_1 = d_2 \tan\left(\frac{\alpha}{2}\right) \quad (4)$$

De (2) e (4), obtemos:

$$d_1 = \frac{r_x}{2} \sin(A) * \tan\left(\frac{\alpha}{2}\right) \quad (5)$$

A partir da Fig. IV.21 pode-se deduzir:

$$r_y = d * \cos(\beta) \Leftrightarrow \beta = \text{acos}\left(\frac{r_y}{d}\right) \quad (6)$$

$$d_1 = d * \sin(\beta) \quad (7)$$

De (6) e (7) obtemos:

$$d_1 = d * \sin\left(\text{acos}\left(\frac{r_y}{d}\right)\right) \quad (8)$$

Para resolver as equações (5) e (8) é efectuada uma aproximação numérica, variando o valor do ângulo “A” e altura “h” de forma a minimizar o erro até a um máximo de N iterações.

Com o cálculo da altura “h” e o ângulo “A” a partir deste algoritmo e com a orientação da elipse obtida no ponto anterior, possuem-se todos os dados que permitam encaixar de forma unívoca uma

elipse a um cone. Dessa forma, é possível obter os quatro pontos das extremidades dos eixos (maior e menor) da elipse no espaço tridimensional, ilustrado na Fig. IV.22. O mesmo cálculo é efectuado para a outra elipse, fazendo o total de oito pontos.

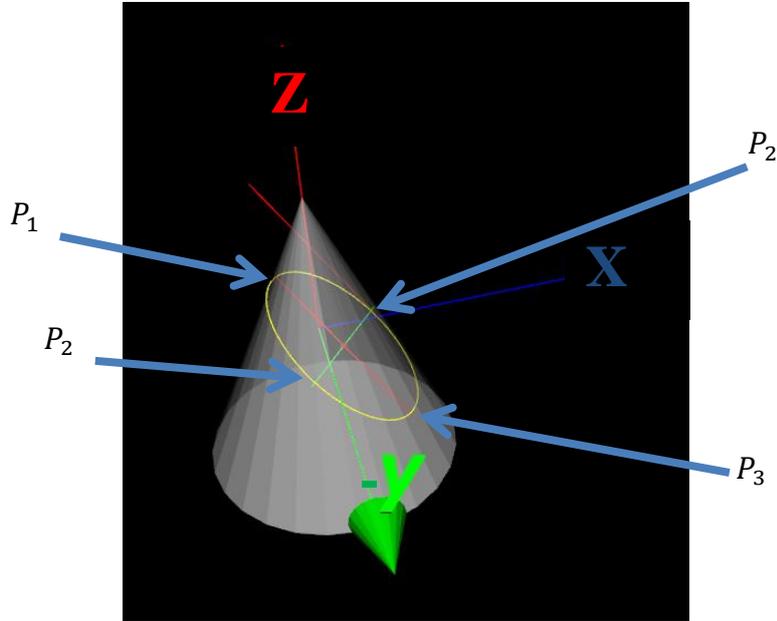


Fig. IV.22 - Encaixe da elipse ao cone no espaço tridimensional

Os pontos presentes na Fig. IV.22 são obtidos da seguinte forma:

Sendo $C(c_x, c_y, c_z)$ o centro da elipse no espaço tridimensional e θ a orientação da elipse calculada no ponto anterior.

$$c_x = \frac{R_x}{2} * \sin(A) * \tan\left(\frac{\alpha}{2}\right) * \cos(\theta)$$

$$c_y = \frac{R_x}{2} * \sin(A) * \tan\left(\frac{\alpha}{2}\right) * \sin(\theta)$$

$$c_z = \frac{H}{2} - h$$

$$P_1 = \left(c_x + \frac{d}{2} * \cos(\theta), c_y + \frac{d}{2} * \sin(\theta), c_z + \frac{R_x}{2} * \sin(A) \right)$$

$$P_2 = \left(c_x + \frac{R_y}{2} * \cos\left(\frac{\pi}{2} + \theta\right), c_y + \frac{d}{2} * \sin\left(\frac{\pi}{2} + \theta\right), c_z \right)$$

$$P_3 = \left(c_x - \frac{d}{2} * \cos(\theta), c_y - \frac{d}{2} * \sin(\theta), c_z - \frac{R_x}{2} * \sin(A) \right)$$

$$P_4 = \left(c_x - \frac{R_y}{2} * \cos\left(\frac{\pi}{2} + \theta\right), c_y - \frac{d}{2} * \sin\left(\frac{\pi}{2} + \theta\right), c_z \right)$$

Sendo R_x e R_y o raio maior e menor respectivamente. C_x e C_y correspondem ao centro da elipse. θ representa o ângulo formado entre o eixo maior e o vector director do objecto de calibração já calculado e A representa o ângulo da inclinação da elipse no cone.

IV.4.2.3. Cálculo da transformação a aplicar aos dados do laser 2D

No processo de ajuste do objecto de calibração aos dados 3D, inicialmente este situa-se com centro em (0,0,0) e através do processo de selecção e *fitting* já explicado, armazenaram-se as transformações necessárias ao objecto inicial até á sua posição final de *fitting*.

O método de encaixe das elipses aos cones descrito no ponto anterior, realiza-o para um cone centrado em (0,0,0). Esses dados sofrem uma transformação, em que os dados da elipse da esquerda sofrem uma translação para o local do cone da esquerda do objecto de calibração e a outra elipse sofre a translação para o lado direito.

Com estas transformações, os oito pontos das duas elipses coincidem com a posição inicial do objecto de calibração. Estando nesta posição e como estão armazenadas as transformações do objecto de calibração no modelo 3D, é apenas necessário aplicar as mesmas transformações para obter a posição final dos oito pontos das elipses no referencial do laser 3D.

No laser 2D, com o cálculo dois oito pontos bidimensionais das elipses, obtemos também oito pontos referentes às extremidades do eixo maior e menor das elipses. A ferramenta VTK disponibiliza um método de transformação de corpo rígido (*vtkLandMarkTransform*), em que dando uma correspondência entre um conjunto de pontos, é devolvida a matriz de transformação do primeiro conjunto de pontos origem para o conjunto de pontos de destino de acordo com os erros mínimos quadrados. Neste caso, os pontos de origem são os oito pontos das elipses do laser 2D; e os pontos de destino são os oito pontos das elipses em 3D. O resultado dessa transformação resulta numa matriz de calibração a aplicar no laser 2D.

IV.4.3. Resultados

Foram realizados testes com diversos objectos de calibração para avaliar a qualidade da calibração. Para quantificar o desempenho é calculada a distância média entre cada ponto do laser 2D ideal com o seu ponto correspondente utilizando o processo de calibração. Desta forma, determina-se a distância média a que cada ponto do laser 2D calibrado está relativo à sua posição ideal.

Os parâmetros relativos ao laser 3D (posição, orientação) não são alterados por se tratar do laser de referência. A sua posição é fixa e aproximada à encontrada no AtlasCar, tendo em conta que o centro de coordenadas é o próprio carro, o laser 3D sofre uma translação dada pelo vector (0,0,1.5).

Foram simulados vários lasers 2D em posições diferentes e foram efectuados sucessivas calibrações usando um, dois e três objectos de calibração. O valor de resolução angular do laser 2D e 3D foram mantidos igual a 0,25 para todas as experiências por corresponderem aos valores reais dos respectivos lasers no AtlasCar. Todos os resultados simulados são baseados no modelo triangulado da oficina do Departamento de Engenharia Mecânica adquirido anteriormente com o veículo. O objecto de calibração usado mantém o tamanho, variando apenas a sua posição e orientação. Estão descritos na Tabela IV.2 os parâmetros do objecto de calibração, resultantes das medições efectuadas ao objecto real.

Altura	0.9
Raio dos cones	0.35
Distância entre cones	1

Tabela IV.2 - Parâmetros fixos do objecto de calibração

IV.4.3.1. Uso de um objecto de calibração

Foram efectuadas duas calibrações, variando a posição e orientação do laser 2D segundo a Tabela IV.3.

Aquisição	Posição (x,y,z)	Orientação
1	(0, 0.5, 1)	(2°, 10°, 0°)
2	(0, -0.5 0.8)	(2°, 8°, 0°)

Tabela IV.3 - Parâmetros do laser 2D usando 1 objecto de calibração

Os parâmetros do objecto de calibração estão representados na Tabela IV.4.

Centro do objecto (x,y,z)	(5, -1, 0.2)
Rotação	(0°, 0°, 0°)

Tabela IV.4 - Parâmetros do objecto de calibração

Na Fig. IV.23 está uma vista mais abrangente da posição do objecto de calibração (marcado a vermelho). Estão presentes também a preto os dados do laser 3D e a verde claro os dados do laser 2D.



Fig. IV.23 - Vista da posição do objecto de calibração na cena
A preto estão os dados do laser 3D e a verde claro os dados do laser 2D. Está assinalado o objecto de calibração obtido.

Na Fig. IV.24 (a) estão representados os dados após calibração (roxo) e os dados ideais (verde claro) e é visível que os dados não se sobrepõem. No entanto, no pormenor da Fig. IV.24 (b), na zona do objecto de calibração os dados 2D estão sobrepostos. A distância média entre os pontos calibrados e os pontos ideais é de aproximadamente 5cm.

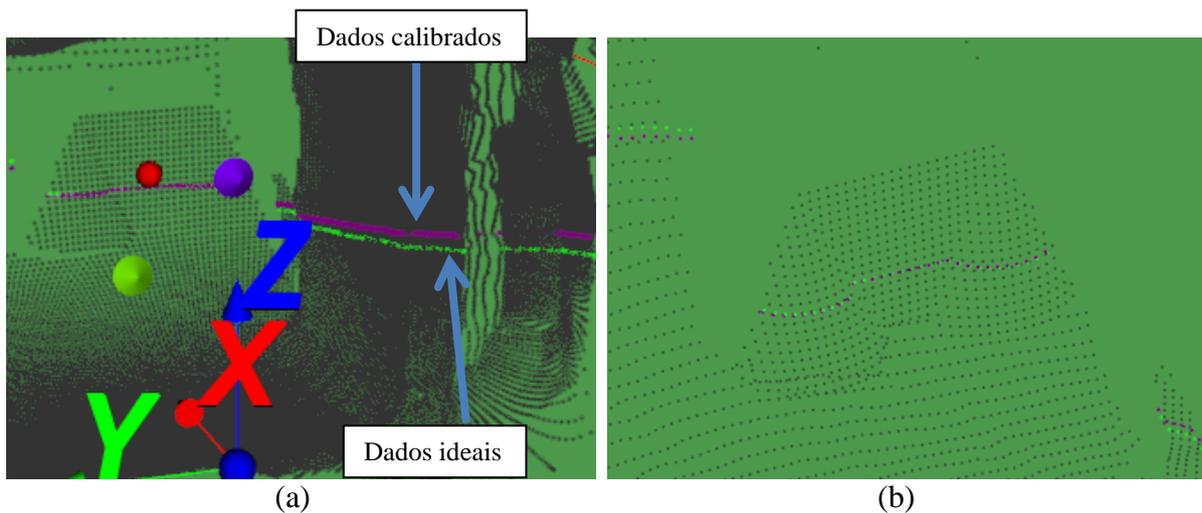


Fig. IV.24 - Calibração usando um objecto de calibração

IV.4.3.2. Uso de dois objectos de calibração

Foram efectuadas duas calibrações, desta vez usando dois objectos de calibração, variando ligeiramente a posição e orientação do laser 2D segundo a Tabela IV.5.

Aquisição	Posição (x,y,z)	Orientação
1	(0, -0.5, 0.8)	(2°, 8°, 0°)
2	(0, 0.5, 1)	(2°, 10°, 0°)

Tabela IV.5 - Parâmetros do laser 2D usando dois objectos de calibração

Os parâmetros dos objectos de calibração estão descritos na Tabela IV.6.

Objecto de calibração	Posição (x,y,z)	Orientação
1	(5, -1, 0.2)	(0°, 0°, 0°)
2	(9, 0.5, 0)	(0°, 0°, 10°)

Tabela IV.6 - Parâmetros dos dois objectos de calibração utilizados

Na Fig. IV.25 está uma visão abrangente da posição dos objectos de calibração na cena (assinalados a vermelho). A preto estão representados os dados do laser 3D e a verde claro os dados do laser 2D.



Fig. IV.25 - Aquisição com dois objectos de calibração

Na Fig. IV.26 está presente a calibração usando dois objectos de calibração. Os dados não estão sobrepostos, estando os dados calibrados a uma distância média de 17mm dos dados ideais.

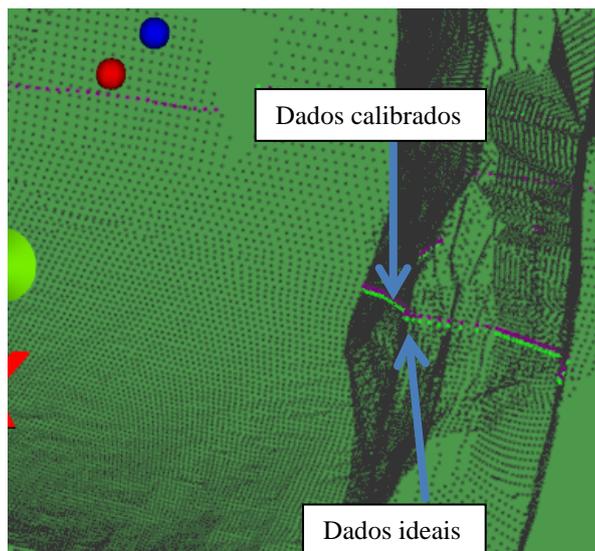


Fig. IV.26 - Calibração usando dois objectos de calibração

IV.4.3.3. Uso de três objectos de calibração

Foram efectuadas duas calibrações, variando ligeiramente a posição e orientação do laser 2D segundo a Tabela IV.7.

Aquisição	Posição (x,y,z)	Orientação
1	(0, -0.5, 0.8)	(-3°, 7°, 0)
2	(0, 0.5, 0.7)	(3°, 7°, 0)

Tabela IV.7 - Parâmetros do laser 2D usando três objectos de calibração

Os parâmetros dos objectos de calibração estão descritos na Tabela IV.8.

Objecto de calibração	Posição (x,y,z)	Orientação
1	(5, -1, 0.2)	(0°, 0°, 0°)
2	(9, 0.5, 0)	(0°, 0°, 10°)
3	(6.5, 2, 0)	(0°, 0°, 10°)

Tabela IV.8 - Parâmetros dos três objectos de calibração utilizados

Na Fig. IV.27 está uma visão abrangente da posição dos objectos de calibração na cena (assinalados a vermelho). A preto estão representados os dados do laser 3D e a verde claro os dados do laser 2D.



Fig. IV.27 - Aquisição usando três objectos de calibração

Na Fig. IV.28 está o resultado da calibração do laser 2D, onde a linha a roxo são os dados do laser 2D calibrados. Neste exemplo não é possível verificar a existência da linha a verde claro porque os dados calibrados sobrepõem-se aos dados ideais. A distância média entre os pontos após calibração e os dados ideais é de ~5mm, o que não é visível e apenas verificado numericamente. Em ambas as calibrações os dados calibrados estão sobrepostos, sendo apenas necessário ilustrar um dos exemplos.

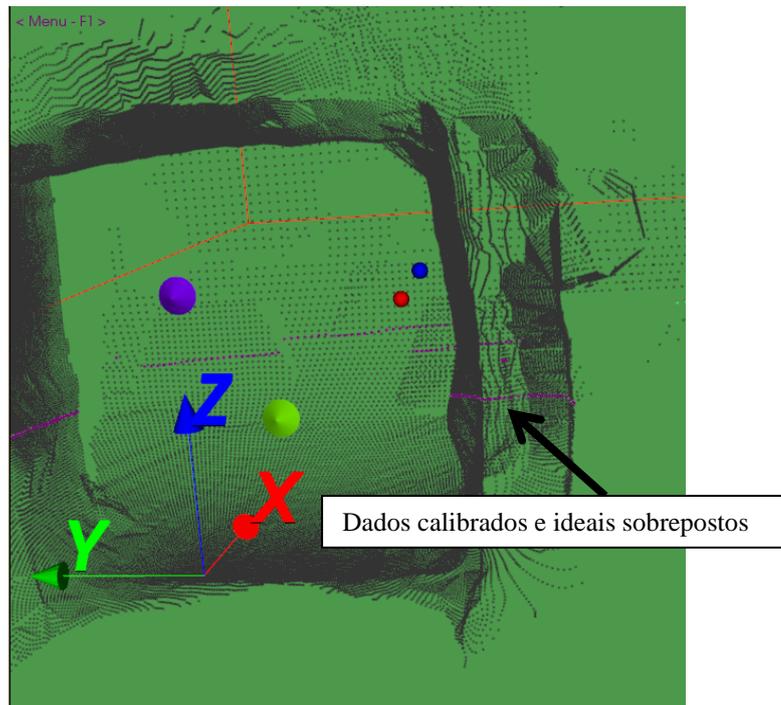


Fig. IV.28 - Calibração com três objectos de calibração.

A linha a roxo representa os dados do laser 2D após calibração. Neste exemplo a linha a verde não é visível porque os dados calibrados se sobrepõem aos dados ideais.

IV.4.4. Conclusões

Analisando os resultados obtidos, verifica-se que no caso em que só é usado um objecto de calibração, existe um menor desempenho, causado pelo algoritmo de reconstrução das elipses. Como se trata de reconstruir uma elipse com apenas parte dos dados e dependendo do erro introduzido nos dados, os parâmetros das elipses devolvidos pelo algoritmo também contém algum erro associado.

Como é efectuado uma calibração a uma zona restrita, que neste caso em particular é a zona do objecto de calibração, faz com que pequenas variações a nível do *fitting* da elipse produzam grandes erros à medida que se afastam da zona do objecto de calibração. Este fenómeno é verificado no pormenor da Fig. IV.24 (b) em que os dados na área do objecto de calibração estão visualmente sobrepostos mas na Fig. IV.24 (a) verifica-se que os dados calibrados estão desalinhados.

Uma forma de contornar este problema passa pela adição de objectos de calibração pois isso significa utilizar mais pontos para realizar a transformação de corpo rígido, minimizar o erro e a distância média para a calibração ideal. Com um objecto de calibração são usados um total de oito pontos, referentes aos pontos dos eixos maior e menor de cada elipse, no entanto o número de pontos aumenta em $8*N$, sendo N o número de objectos de calibração usados. A Tabela IV.9 mostra uma diminuição da distância média dos pontos calibrados dos pontos ideais na ordem dos 90% adicionando três objectos de calibração.

Número de Objectos de calibração	Distância média (mm)
1	50
2	17
3	5

Tabela IV.9 - Distância média em função do número de objectos de calibração

O processamento exigido no cálculo da transformação dos $8*N$ pontos manteve-se nos três casos em estudo igual ou menor a um segundo, pelo que o desempenho não fica prejudicado. Apenas é exigido maior interacção do utilizador com a aplicação na selecção dos pontos e identificação visual dos objectos de calibração.

IV.5. Aplicação em dados reais

Verificando a validade teórica do método proposto, foram realizados testes com dados reais. Com dados reais surgem problemas acrescidos: os erros associados aos próprios lasers; a sua sensibilidade; e a qualidade jogam um papel muito importante nesta fase. Um dos problemas que poderá surgir está ligado à densidade da nuvem de pontos devolvida pelos diversos lasers que podem ser insuficientes para aplicar o método.

IV.5.1. Objecto de calibração

O primeiro trabalho para testes reais passou pela criação de um objecto de calibração em tamanho real que será usado na validação. O protótipo usado foi desenvolvido com placas de metal zincado de 0.5mm de espessura para os cones e uma placa de contraplacado para a placa intermédia. O processo de fabrico foi artesanal respeitando o mais possível as questões de tamanho e estrutura. O tamanho final da estrutura, com base nas medições efectuadas está descrito na Tabela IV.10.

Altura	0.924m
Raio dos cones	0.3825
Distância entre cones	1m

Tabela IV.10 - Dados do objecto de calibração real

Na Fig. IV.29 está presente o protótipo desenvolvido. A estrutura é desmontável para facilitar o seu transporte. A superfície espelhada dos cones proporcionada pelo metal zincado levou a que fossem feitos testes preliminares no laboratório com um laser SICK LMS-100 para garantir bons resultados com os lasers presentes no AtlasCar.



Fig. IV.29 - Protótipo do objecto de calibração.

Após alguns testes, a superfície espelhada revelou ser um problema por reflectir os raios do laser para outras direcções, não obtendo uma superfície regular e precisa. A solução passou por pintar a superfície dos cones em cor branco mate. Com esta alteração obtiveram-se bons resultados, minimizando este problema.

Posteriormente foram feitas aquisições com o AtlasCar, em que o objecto de calibração foi colocado a diferentes distâncias e posições para efectuar testes com nuvens de pontos mais ou menos densas e em diferentes ângulos.



Fig. IV.30 - Local das aquisições feitas com o AtlasCar com o objecto de calibração.

IV.5.2. Aquisições de dados laser do AtlasCar

Foram feitas quatro aquisições com o AtlasCar do objecto de calibração, variando em cada uma delas a posição deste. Tendo em conta a limitação do laser 3D já referida da falta de definição à medida que se afasta do centro óptico, foram realizadas duas primeiras aquisições em que numa delas, o objecto de calibração se encontrava a sensivelmente 5m do veículo na direcção do centro óptico do laser 3D (Fig. IV.32 (a)) e na segunda aquisição o objecto encontrava-se a 5m mas mais à esquerda do veículo. Observando a Fig. IV.32 (b) verifica-se que visualmente se torna difícil de identificar o objecto de calibração.

Devido a este problema de definição dos dados 3D, foram realizados testes ao resultado da aplicação do algoritmo ICP usando uma aquisição e duas aquisições acumuladas do laser 3D. O nível de precisão apenas é possível de obter visualmente, comparando os resultados em ambos os casos. Está presente na Fig. IV.31 (a) o resultado com uma aquisição e na Fig. IV.31 (b) o resultado com duas aquisições acumuladas.

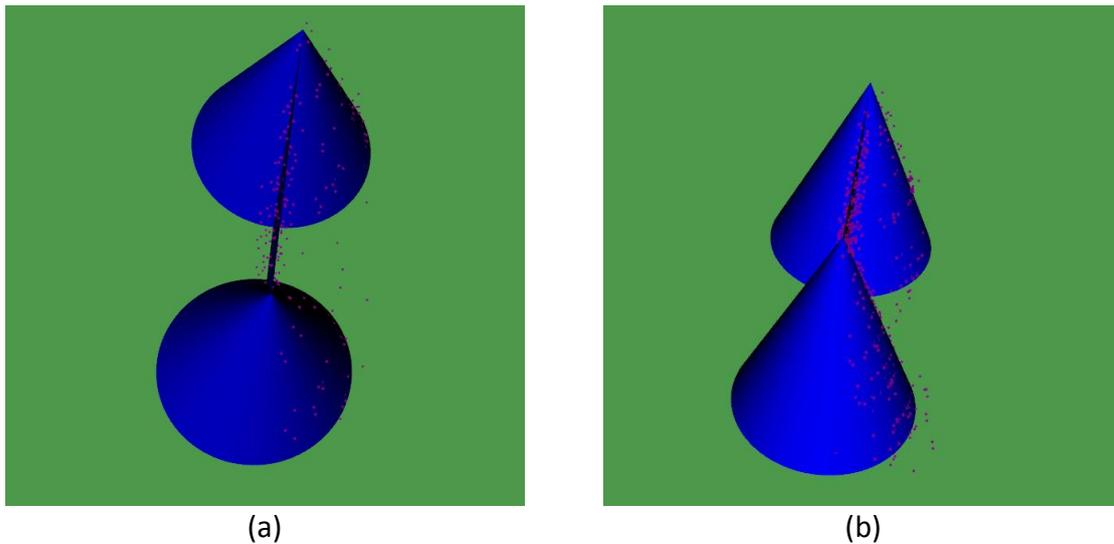
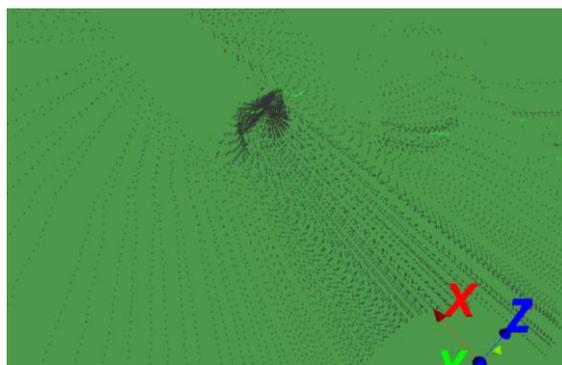


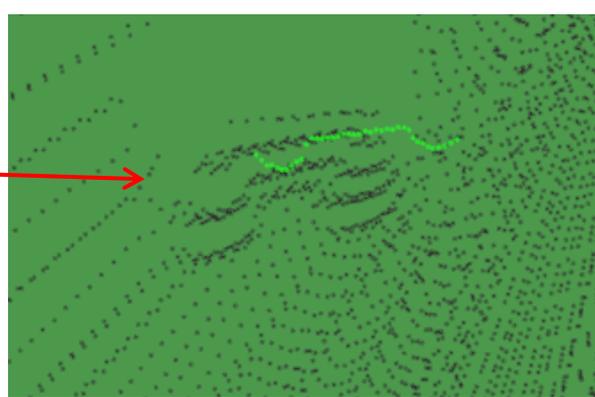
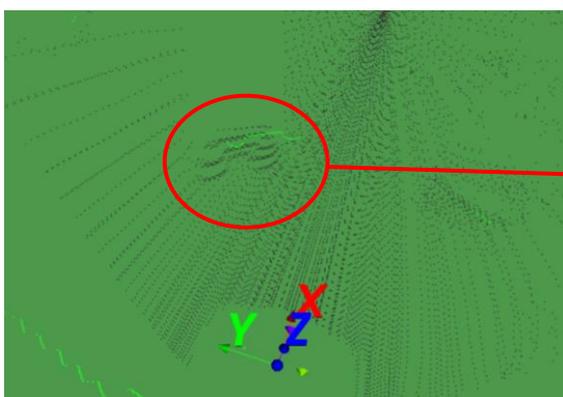
Fig. IV.31 - Comparação da aplicação do ICP a diferentes densidades de pontos
(a) resultado da aplicação do ICP com uma aquisição do laser 3D e (b) com acumulação de duas aquisições.

Visualmente não foram obtidos melhores resultados, mostrando que a interação do utilizador, com a selecção de dois pontos no objecto de calibração fornece dados suficientes para o algoritmo ICP produzir bons resultados. Com isto, serão realizados testes com apenas uma aquisição de cada objecto de calibração acumulada e não duas aquisições por objecto.

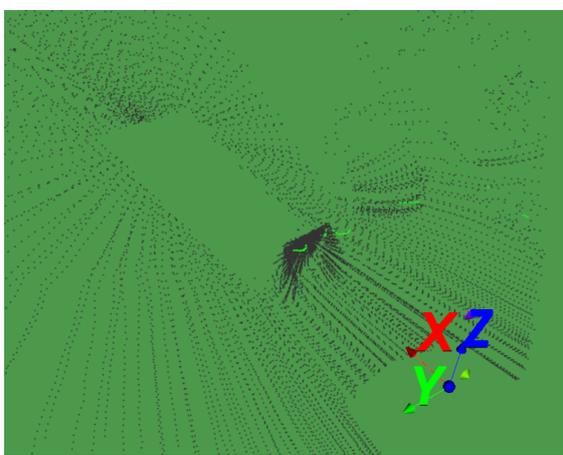
Para evitar alguns problemas de definição do objecto e também devido às limitações do local onde foram efectuadas as aquisições, foram realizadas mais duas aquisições com o objecto de calibração na proximidade do centro óptico do laser 3D, nomeadamente a 3m (Fig. IV.32(c)) e a 10m (Fig. IV.32(d)) do veículo.



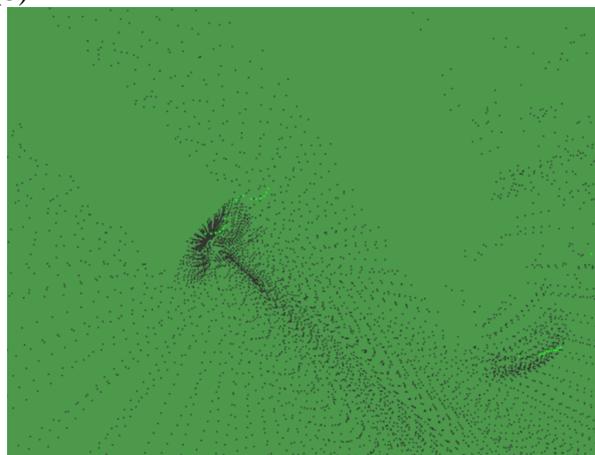
(a)



(b)



(c)



(d)

Fig. IV.32 - Dados do laser 2D e 3D do AtlasCar

Para que sejam efectuados testes com dados reais usando um, dois ou três objectos de calibração, estes teriam que ser construídos em número suficiente e depois efectuadas as aquisições. A construção artesanal do objecto de calibração torna a sua replicação exacta difícil e morosa. A solução passou pelo uso das ferramentas de interacção da aplicação de visualização de dados do AtlasCar desenvolvido no ponto II.5 e sobrepôr as diversas aquisições do objecto de calibração em posições diferentes, mantendo a posição do veículo inalterada.

Com o leitor de *Logs* e visualizando os dados do laser 3D e de um laser planar SICK LMS-100 obtém-se o resultado apresentado na Fig. IV.33(a). Activando a funcionalidade de acumulação de pontos do visualizador, os dados presentes na janela de visualização não são substituídos. De seguida é aberto outro *log* com outra aquisição do AtlasCar e é colocado em reprodução durante uma mensagem do laser 3D e do laser planar. Desta forma, com a funcionalidade de acumulação de pontos, podem ser sobrepostas diversas aquisições que no caso particular foram com o objecto de calibração a 3m e 5m (Fig. IV.33(b)), a 3m, 5m e 10m (Fig. IV.33(c)) e por fim com o objecto de calibração a 3m, 5m, 10m e o objecto a 5m mas mais à esquerda (Fig. IV.33(d)).

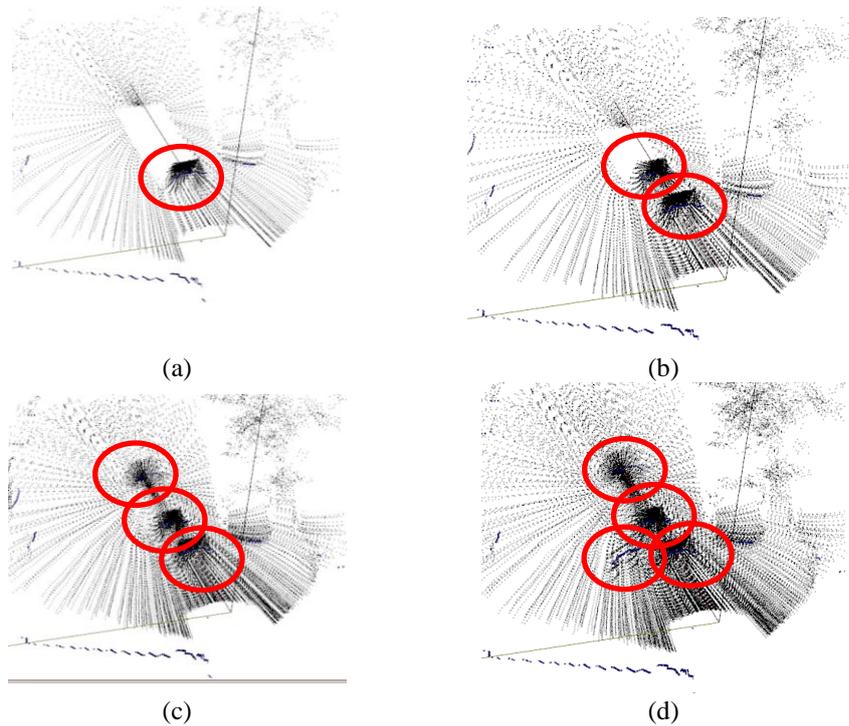


Fig. IV.33 - Sobreposição de dados 3D com vários objectos de calibração
Em (a) está uma aquisição com um objecto de calibração, em (b) a acumulação de um segundo, em (c) um terceiro e em (d) um quarto objecto de calibração.

Em cada aquisição visualizada, com ou sem sobreposição, os dados dos lasers 2D e 3D são guardados em ficheiros separados (funcionalidade também permitida pelo visualizador) para que sejam fornecidos como parâmetro de entrada à aplicação de calibração para efectuar os testes. Tal como foi verificado nos resultados com dados simulados, obtém-se resultados com erros de milímetros entre a calibração calculada e a ideal com recurso a três objectos de calibração. Devido a isso são apenas considerados nos testes práticos os dados contendo três ou quatro objectos de calibração.

IV.5.3. Resultados

Relativo aos dados do laser 3D, o *fitting* do objecto de calibração tal como com os dados simulados, após a selecção dos pontos, é realizado uma filtragem dos dados antes da aplicação do algoritmo ICP.

Na Fig. IV.34 (a) está presente a visualização do resultado da aplicação do algoritmo ICP quanto à placa intermédia entre os cones. Na Fig. IV.34(b) está ilustrado em maior detalhe a zona de superfície dos cones e o resultado após a aplicação do ICP.

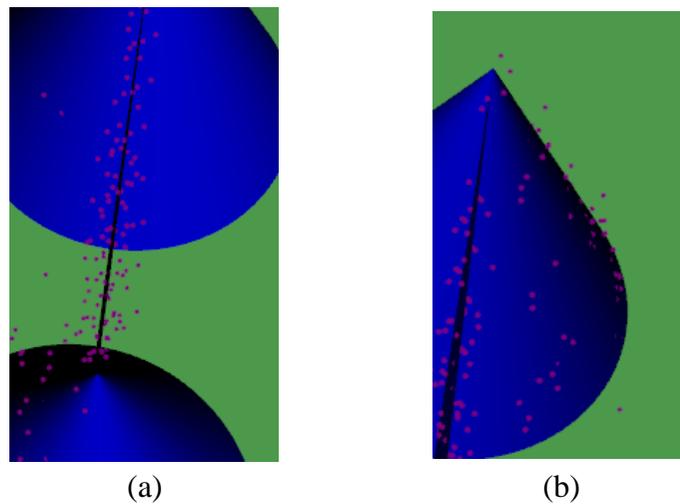


Fig. IV.34 - Resultado do ICP em dados reais.

Em (a) está presente o pormenor do *fitting* na zona da placa intermédia e em (b) o *fitting* de pontos na superfície dos cones.

Tendo em conta o erro e ruído associado ao laser 3D, os resultados obtidos são bons. Este resultado permite de alguma forma minimizar o erro presente nos dados através do ICP e todo o processamento será com base neste modelo ideal.

Quanto ao laser 2D, os resultados estão divididos em reconstrução das elipses e o resultado da calibração. Na Fig. IV.35 estão presentes alguns resultados da reconstrução das elipses nos diversos objectos de calibração. As elipses produzidas não são muito precisas devido aos erros subjacentes ao laser 2D e o número de pontos está quase sempre no limite mínimo da aplicação do algoritmo de construção das elipses (6 pontos).

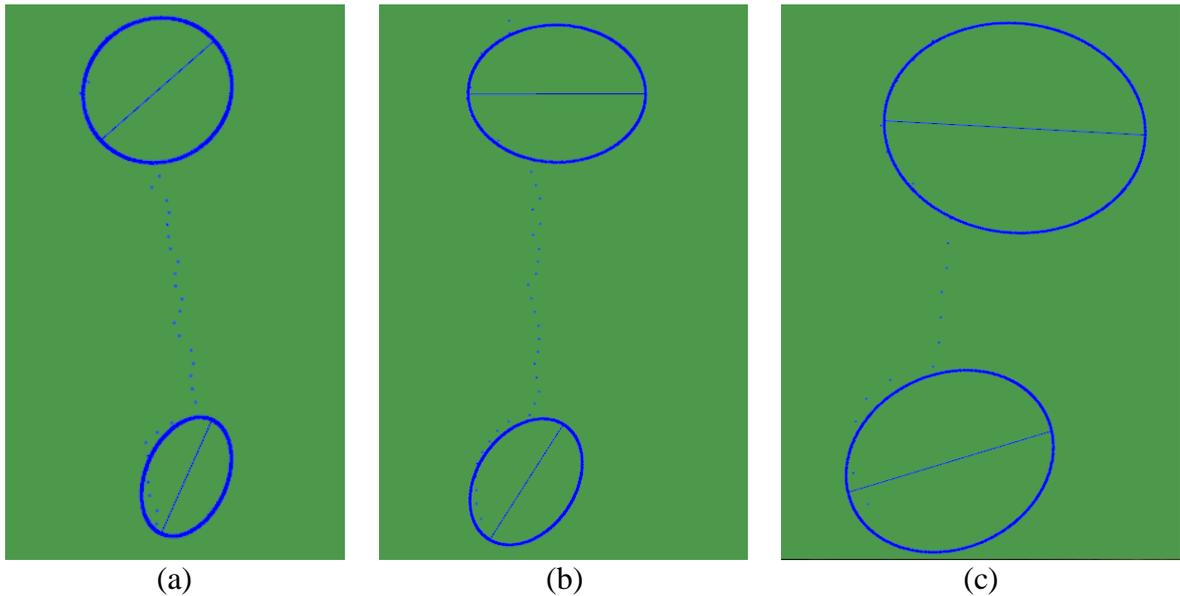


Fig. IV.35 - Reconstrução de elipses em dados laser 2D reais.
Resultado da aplicação do algoritmo de reconstrução das elipses. (a) Objecto de calibração colocado a 3m; (b) Objecto de calibração colocado a 5m; (c) Objecto de calibração colocado a 10m.

Numa tentativa de melhorar a qualidade das elipses, foram acumuladas duas aquisições do laser 2D de um objecto de calibração na tentativa de duplicar o número de pontos usados na criação das elipses. O resultado da acumulação de duas aquisições do laser 2D (Fig. IV.36) produz novos pontos que praticamente coincidem com os da aquisição anterior, assinalado a vermelho. Esta abordagem não é viável e devido a isso não foram realizados testes com este processo.

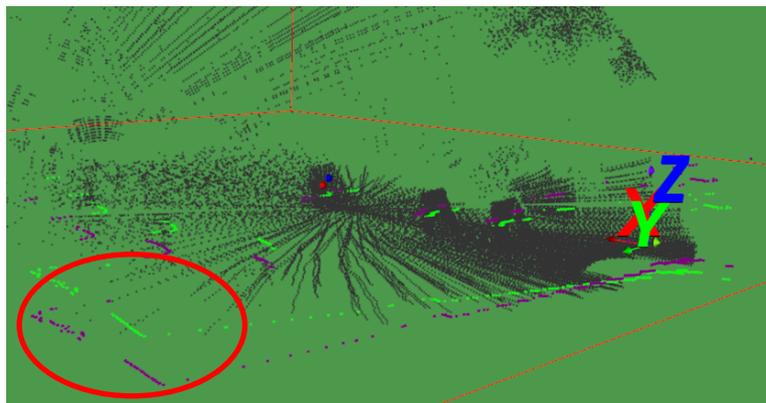


Fig. IV.36 - Resultado da acumulação de duas aquisições do laser 2D

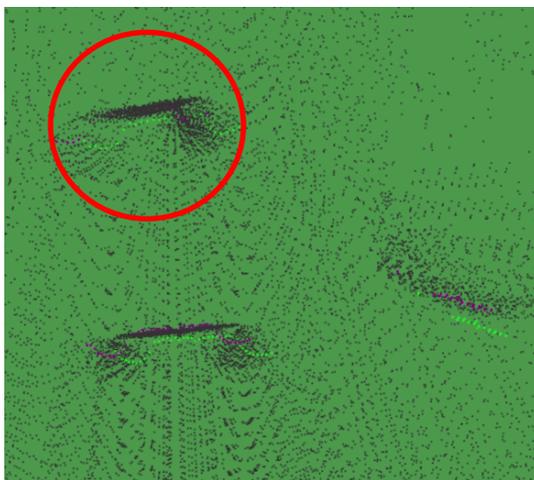
O resultado da calibração usando três objectos de calibração está ilustrado na Fig. IV.37. A roxo estão os dados obtidos através da calibração e a verde claro os dados originais. Na Fig. IV.37(a) está uma visão geral da cena 3D com os objectos de calibração e os resultados da calibração. A nível da translação é visível o melhoramento conseguido através da calibração em pormenor na Fig. IV.37(b) e (c), assinaladas a vermelho. Os dados a roxo estão visualmente mais alinhados em relação ao laser 3D do que os dados originais a verde claro. O erro da primeira calibração fornecida (pelos cálculos manuais efectuados no AtlasCar) é maior à medida que aumenta a distância ao veículo

(Fig. IV.37(c)). Neste caso, após o processo de calibração é conseguido um alinhamento mais preciso, analisando os dados visualmente.

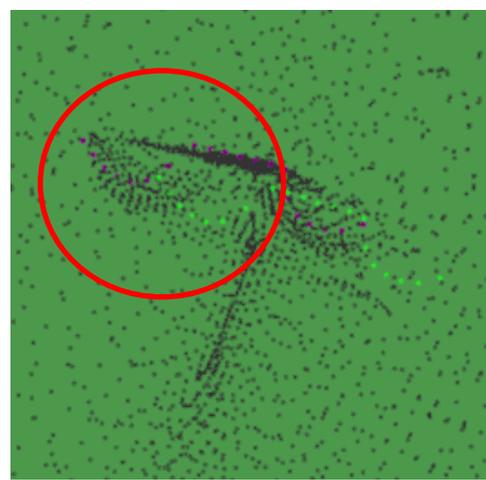
No entanto, existe uma maior discrepância de dados em questão da rotação destes em torno do eixo X. A existência de erros no cálculo das elipses estará na origem deste facto. Como o encaixe das elipses no cone é um método preciso e sensível a erro, a forma de como as elipses fornecidas ao algoritmo encaixam no cone produzem resultados que não correspondem à realidade. Observando a Fig. IV.37 (a) verifica-se um cálculo da rotação exagerada e visualmente não está correcto.



(a)



(b)



(c)

Fig. IV.37 - Resultado da calibração em dados reais usando três objectos de calibração.

A roxo os dados após calibração e a verde claro os dados originais. Em (a) está presente uma visão geral, mostrando o problema na rotação em torno do eixo X, em (b) o melhoramento conseguido através da calibração e (c) mostra em pormenor a precisão em dados situados a maiores distâncias do AtlasCar

O resultado da calibração usando quatro objectos de calibração está ilustrado na Fig. IV.38. A roxo estão os dados obtidos através da calibração e a verde claro os dados originais. Comparando este teste efectuado com quatro objectos de calibração verifica-se um ligeiro aumento na precisão dos

dados (Fig. IV.38(b),(c) e (d)). A nível da rotação em torno do eixo X continua a verificar-se de que os dados não correspondem à realidade, apesar de que a diferença seja menor do que com apenas três objectos de calibração.

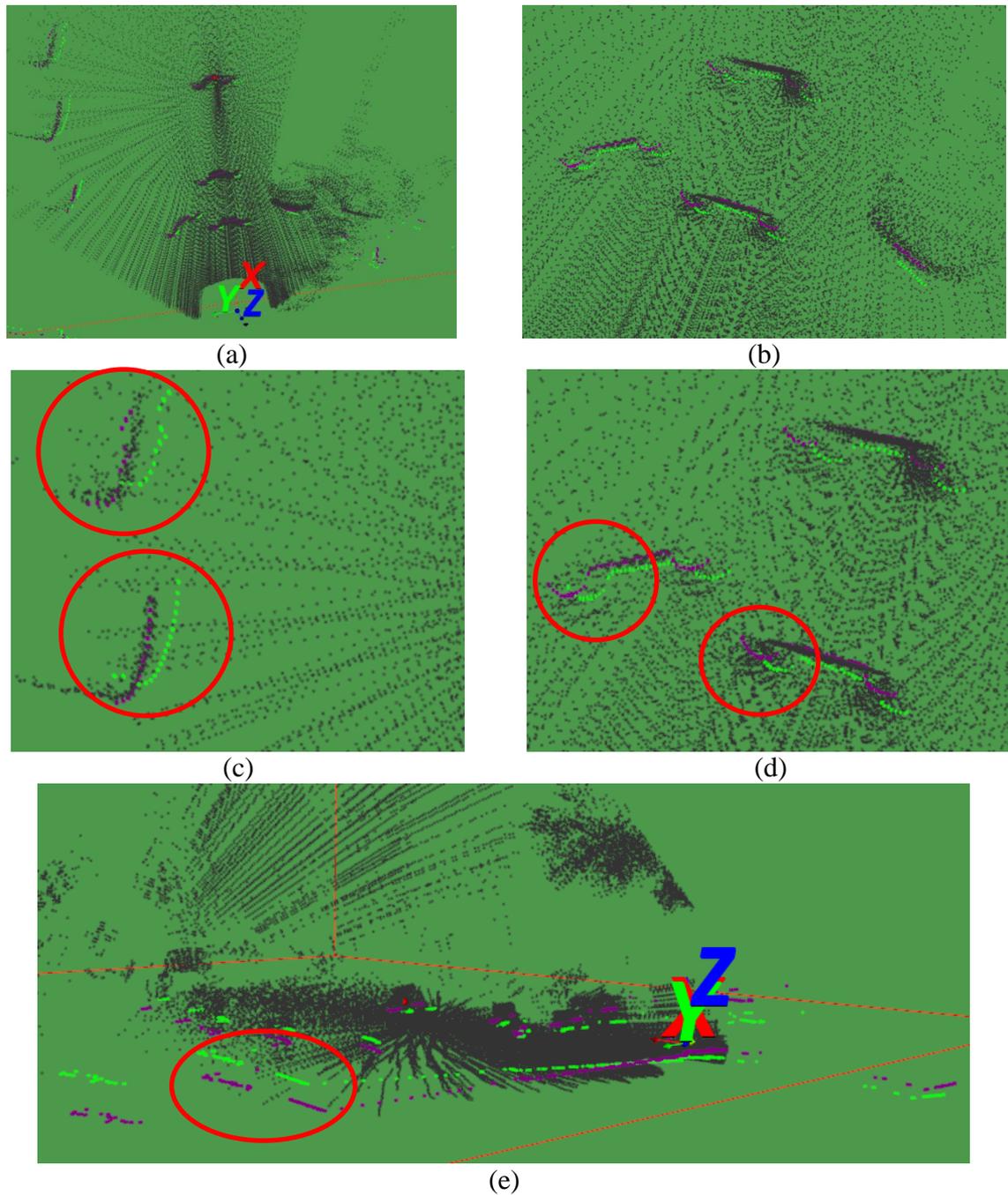


Fig. IV.38 - Resultado da calibração em dados reais usando quatro objectos de calibração. A roxo os dados após calibração e a verde claro os dados originais. Em (a) está presente uma visão geral, mostrando o problema na rotação em torno do eixo X, em (b) o melhoramento conseguido através da calibração, (c) e (d) mostram em maior pormenor a precisão da calibração realizada e em (e) o problema da rotação em torno do eixo X.

IV.5.4. Conclusões

Os resultados apresentados demonstram que visualmente a calibração do laser 2D relativamente ao laser 3D é mais precisa do que aquela já usada no AtlasCar. Os dados 3D não são tão sensíveis aos erros pois não é efectuada nenhuma operação de reconstrução mas apenas de minimização de erros com base num modelo cujas propriedades são conhecidas *à priori*. Quanto ao laser 2D, é efectuada uma reconstrução de elipses com um número muito reduzido de pontos (entre 6 e 9) e com a presença de ruído, torna o processo de cálculo da elipse mais difícil. Este problema é reflectido na rotação em torno do eixo X, em que o tamanho das elipses influencia directamente esta variável. Quanto à translação os resultados obtidos são mais promissores porque o centro da elipse que é calculada situa-se muito próximo do centro do cone.

A Fig. IV.35 (a) contém um exemplo de uma elipse cujos resultados não fazem sentido, na medida em que o parâmetro de comprimento do eixo maior é muito acentuado, comprometendo o resultado do encaixe dessa elipse num cone.

Devido à dificuldade enfrentada na aquisição de dados com o AtlasCar e a dificuldade em termos de espaço para as aquisições, não foi possível efectuar mais testes com diferentes posições e/ou orientações do objecto de calibração no espaço de tempo dedicado nesta tese.

IV.6. Conclusões

Com o estudo teórico e aplicação dos algoritmos e metodologias em dados simulados, verificou-se que a calibração de um laser 2D relativamente a um laser 3D é possível e demonstrada. Com dados reais não foi possível demonstrar a mesma eficácia verificada com os dados simulados mas foi possível observar que melhora a calibração já existente, com apenas a rotação em torno do eixo X por resolver e melhorar.

O fabrico artesanal do objecto de calibração não permite obter uma precisão de milímetros como verificado em dados simulados mas tal como referido, esta tese é um primeiro passo para atingir a calibração integrada de lasers e câmaras. Este projecto demonstra que é possível a calibração de um laser 2D a partir do laser 3D de referência do AtlasCar.

O aumento da resolução do laser 2D poderá aumentar a eficácia no cálculo das elipses mas não foi possível realizar novas aquisições com o AtlasCar por problemas de disponibilidade e facilidade na aquisição de novos dados e configuração do laser 2D.

V. Conclusões e trabalho futuro

V.1. Conclusões

Nesta tese foram atingidos os dois objectivos propostos:

- Na área da reconstrução 3D, foram estudados os métodos já existentes, desenvolvendo uma aplicação que aplica o método escolhido aos dados do AtlasCar. O desenvolvimento da aplicação de reconstrução 3D levou primeiramente à criação de uma plataforma de leitura de dados dos sensores e lasers do AtlasCar integrada sobre a plataforma VTK, que permite a integração de futuras ferramentas sobre a aplicação já desenvolvida. Foram desenvolvidas ferramentas de interacção que acrescentam funcionalidades que não existiam, numa plataforma escalável e robusta.

Com o estudo realizado na reconstrução 3D, foi escolhido o algoritmo que melhor se adaptava a dados não estruturados, em ambientes abertos ou fechados. A conversão e integração na plataforma VTK permite o uso deste algoritmo a diversos dados nas diversas plataformas do projecto Atlas, não se restringindo ao AtlasCar.

- Na área da calibração, houve uma contribuição com um algoritmo de *fitting* de uma elipse a um cone de forma unívoca através da combinação da reconstrução de dados 2D com dados 3D. Foi integrado um conjunto de ferramentas de interacção com os dados que facilitam o processo de calibração. Foi provado teoricamente que o processo de calibração atinge o objectivo proposto. A limitação encontrada teve a ver com a precisão dos dados devolvidos pelos lasers, que com os erros associados, interferem no seu pleno funcionamento.

O módulo de calibração possui as estruturas para suportar diversos objectos de calibração, podendo ser diferentes daquele proposto nesta tese e possui ferramentas úteis ao projecto AtlasCar que podem ser integradas aos módulos já existentes assim que seja validado com dados reais.

V.2. Trabalho futuro

Durante o desenvolvimento da tese, surgiram vertentes que poderão ser exploradas futuramente, entre elas:

Melhoramentos:

O módulo de visualização dos dados dos lasers e sensores poderá suportar funcionalidades de pós-processamento dos dados obtidos para diminuir o processamento exigido a estruturas ou módulos que usem os dados devolvidos. A nível da câmara estéreo, esta possui um ruído considerável nos dados que o torna difícil de usar em projectos que exigem precisão.

No módulo de reconstrução, o uso do algoritmo BPA é versátil quanto ao tipo de dados que suporta, mas a adaptação do tamanho do raio da esfera às diferentes densidades de pontos na aquisição levará à criação de uma malha triangulada mais completa.

Aplicação de textura com sobreposição de imagens de diversas câmaras

Um avanço nesta área é a inclusão de diversas imagens de câmaras para aplicar textura ao modelo de superfícies. Desde a inclusão como a fusão de diversas imagens, com tratamento das zonas comuns das imagens para produzir uma superfície homogénea e mais abrangente.

Detecção automática dos objectos de calibração

A estrutura actual necessita da interacção directa do utilizador para detecção dos objectos de calibração na cena. A detecção automática dos objectos minimiza a interacção e torna o processo de calibração mais automático. Uma possível solução passará pela análise da nuvem de pontos por camadas, identificando os objectos de calibração com base na sua geometria conhecida *a priori*.

Fitting das elipses aos dados do laser 2D

O uso de restrições ou a definição de um tamanho inicial de uma elipse para efectuar a reconstrução das elipses nos dados do laser 2D poderá minimizar os erros introduzidos na calibração produzindo melhores resultados.

Calibração de câmaras

O objecto de calibração existente permite a adição de ferramentas de calibração de câmaras como o uso de padrões como o *chess board*. O módulo de calibração actualmente desenvolvido é uma base para uma calibração integrada de todos os dispositivos do AtlasCar.

A. Apêndice A – Alterações na Makefile

Em ambiente Linux, é de prática comum o uso de uma Makefile para compilação de ficheiros de programação. Uma Makefile não é mais do que um *script* onde estão descritas quais as bibliotecas usadas pelos diversos ficheiros que compõem a aplicação, o caminho local onde estas se localizam, regras de como criar o executável da aplicação e qual o compilador que será usado para efectuar a compilação.

No caso específico da Makefile utilizada no desenvolvimento de módulos no AtlasCar, foi necessário proceder a alterações para permitir a integração do VTK no desenvolvimento das plataformas.

Em primeiro lugar, é necessário especificar onde se localiza a biblioteca do VTK e armazenar os dados em variáveis para mais fácil referência (Fig. V.1)

```
16 VTKLIBPATH = /usr/local/lib/vtk-5.6
17 VTKINCPATH = /usr/local/include/vtk-5.6
18 IFLAGS += -I./ -I$(VTKINCPATH)
19
```

Fig. V.1 - Definição do caminho da biblioteca VTK

De seguida é necessário incluir as bibliotecas como especificado na Fig. V.2.

```
25 LFLAGS+= -lglobal\
26          -lipc\
27          -lparam_interface\
28          -lipc_wrapper\
29          libreadcommandlineparameters.a\
30          libxb3_interface_64bit.a\
31          libglwindow_wrapper.a\
32          liblaser3d_interface.a\
33          libhokuyo_utm_interface.a\
34          -L$(VTKLIBPATH) \
35          -lMapReduceMPI -lvtkGenericFiltering -lvtkpng\
36          -lmpistubs -lvtkGeovis -lvtkproj4\
37          -lvtkalglib -lvtkGraphics -lvtkRendering\
38          -lvtkCharts -lvtkHybrid -lvtksqlite\
39          -lvtkCommon -lvtkImaging -lvtksys\
40          -lvtkDICOMParser -lvtkInFovis -lvtktiff\
41          -lvtkexoIIC -lvtkIO -lvtkverdict\
42          -lvtkxpat -lvtkjpeg -lvtkViews\
43          -lvtkFiltering -lvtklibxml2 -lvtkVolumeRendering\
44          -lvtkfreetype -lvtkmetaio -lvtkWidgets\
45          -lvtkftgl -lvtkNetCDF -lvtkzlib \
46          /usr/lib/libXt.so.6 \
47          -ldl \
48          -lGLU\
```

Fig. V.2 - Inclusão das bibliotecas necessárias à compilação

Por fim, a definição dos binários de *linkagem* na compilação (Fig. V.3).

```

97 laser_subscribe_example: laser_subscribe_example.o laser_subscribe_example_functions.o liblaser_subscr
98 $(LAR_HOME)/lib/libreadcommandlineparameters.a \
99 $(LAR_HOME)/src/sensors/camera/flea2/libflea2_interface.a\
100 $(LAR_HOME)/src/sensors/camera/flea2/libxb3_interface_64bit.a\
101 $(LAR_HOME)/src/sensors/camera/firei/libfirei_interface.a\
102 $(VTKLIBPATH)/libvtkIO.a $(VTKLIBPATH)/libvtkGraphics.a $(VTKLIBPATH)/libvtkzlib.a \
103 $(VTKLIBPATH)/libvtkFiltering.a $(VTKLIBPATH)/libvtkCommon.a $(VTKLIBPATH)/libvtkImaging.a \
104 $(VTKLIBPATH)/libvtksys.a $(VTKLIBPATH)/libvtkpng.a $(VTKLIBPATH)/libvtklibxml2.a \
105 $(VTKLIBPATH)/libvtktiff.a $(VTKLIBPATH)/libvtkjpeg.a $(VTKLIBPATH)/libvtkWidgets.a \
106 $(VTKLIBPATH)/libvtkVolumeRendering.a $(VTKLIBPATH)/libvtkRendering.a

```

Fig. V.3 - Definição dos ficheiro binário para *linkagem*

Com isto é possível compilar uma aplicação VTK com o comando “*make*” numa consola em ambiente Linux.

VI. Referências

- Amenta, N., Bern, M., Kamvyselis, M., 1998. A new Voronoi-based surface reconstruction algorithm, Proceedings of the 25th annual conference on Computer graphics and interactive techniques. ACM, pp. 415-421.
- Amenta, N., Choi, S., Dey, T.K., Leekha, N., 2000. A simple algorithm for homeomorphic surface reconstruction, Proceedings of the sixteenth annual symposium on Computational geometry. ACM, Clear Water Bay, Kowloon, Hong Kong, pp. 213-222.
- Baker, M.J., 2011. Maths - Euler Angles. [Consult. January 2011] . Disponível em <http://www.euclideanspace.com/maths/geometry/rotations/euler/index.htm>
- Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., Taubin, G., 1999. The ball-pivoting algorithm for surface reconstruction. Visualization and Computer Graphics, IEEE Transactions on 5, 349-359.
- Besl, P.J., McKay, H.D., 1992. A method for registration of 3-D shapes. Pattern Analysis and Machine Intelligence, IEEE Transactions on 14, 239-256.
- Bouguet, J.-Y., 2010. Camera Calibration Toolbox for Matlab.
- Bradski, G., Kaehler, A., 2008. Learning OpenCV: computer vision with the OpenCV library. O'Reilly.
- CARMEN-Team. [Consult. September 2010]. Disponível em <http://carmen.sourceforge.net/home.html>
- Dias, J., 2009. Distanciômetro 3D baseado num unidade laser 2D em movimento contínuo, Departamento de Engenharia Mecânica. Universidade de Aveiro, Aveiro.
- Dias, P., Matos, M., Santos, V., 2006. 3D Reconstruction of Real World Scenes Using a Low-Cost 3D Range Scanner. Computer-Aided Civil and Infrastructure Engineering 21, 486-497.

Howe, N.R., Leventon, M.E., Freeman, W.T., 1999. Bayesian reconstruction of 3d human motion from single-camera video, *Advances in Neural Information Processing Systems*. MIT Press, pp. 820-826.

Kazhdan, M., 2005. Reconstruction of solid models from oriented point sets, *Proceedings of the third Eurographics symposium on Geometry processing*. Eurographics Association, Vienna, Austria, p. 73.

Kazhdan, M., Bolitho, M., Hoppe, H., 2006. Poisson surface reconstruction, *Proceedings of the fourth Eurographics symposium on Geometry processing*. Eurographics Association, Cagliari, Sardinia, Italy, pp. 61-70.

Lim, C.-W., Tan, T.-S., 2006. Surface reconstruction by layer peeling. *Vis. Comput.* 22, 593-603.

Lisca, G., Pangyu, J., Nedevschi, S., 2010. Automatic one step extrinsic calibration of a multi layer laser scanner relative to a stereo camera, *Intelligent Computer Communication and Processing (ICCP)*, 2010 IEEE International Conference on, pp. 223-230.

Núñez, P., Drews, P., Rocha, R., Dias, J., 2009. Data Fusion Calibration for a 3D Laser Range Finder and a Camera using Inertial Data. *Camera*.

Oliveira, M., Stein, P., Almeida, J., Santos, V., 2009. Modular Scalable Architecture for the Navigation of the ATLAS Autonomous Robots, *9th Conference on Autonomous Robot Systems and Competitions*, Castelo Branco.

Qilong, Z., Pless, R., 2004. Extrinsic calibration of a camera and laser range finder (improves camera calibration), *Intelligent Robots and Systems, 2004. (IROS 2004)*. *Proceedings. 2004 IEEE/RSJ International Conference on*, pp. 2301-2306 vol.2303.

Robótica, L.d.A.e., 2010. Projecto AtlasCar. [Consult. September 2010]. Disponível em <http://atlas.web.ua.pt/>.

Santos, V., Almeida, J., x, vila, E., Gameiro, D., Oliveira, M., Pascoal, R., Sabino, R., Stein, P., 2010. ATLASCAR - technologies for a computer assisted driving system on board a common

automobile, Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on, pp. 1421-1427.

Schroeder, W., Martin, K., Lorensen, B., 2004. The visualization toolkit: an object-oriented approach to 3D graphics. Kitware.

Sequeira, V., Goncalves, J.G.M., 2002. 3D reality modelling: photo-realistic 3D models of real world scenes, 3D Data Processing Visualization and Transmission, 2002. Proceedings. First International Symposium on, pp. 776-783.

VCGLib, 2010. VCGLib. [Consult. February 2011]. Disponível em http://vcg.sourceforge.net/index.php/Main_Page.

Zhang, Z., 2000. A flexible new technique for camera calibration. Pattern Analysis and Machine Intelligence, IEEE Transactions on 22, 1330-1334.

Zhengyou, Z., 1999. Flexible camera calibration by viewing a plane from unknown orientations, Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, pp. 666-673 vol.661.