References

[ArrasK] Arras, Kai and Vestli, Sjur, *Hybrid, High-Precision Localisation for the Mail Distributing Mobile Robot System MOPS*, Proc. IEEE Int. Conference Robotics and Automation, pp. 3129-3134, Leuven, Belgium, May 1998.

[ArsenioA] Arsénio, Artur and Ribeiro, M. Isabel, *Active Range Sensing for Mobile Robot Localisation*, Proc. IEEE/RSJ Int. Conference Intelligent Robotic Systems, IROS'98, Victoria, Canada, October 1998.

[ButterfieldS] Butterfield, Stuart, Ng, Kia and Hogg, David, *The RESOLV Texture-Mapping Module*, School of Computer Science, University of Leeds. Report 97.48, December 1997.

[CastroJ] Castro, José, Santos, Vítor and Ribeiro, M. Isabel, *A Multi-Loop Robust Navigation Architecture for Mobile Robots*, Proc. IEEE Int. Conference Robotics and Automation, pp. 970-975, Leuven, Belgium, May 1998.

[CrowleyJ] Crowley, James L, Wallner, F. and Schiele, B., *Position Estimation Using Principal Components of Range Data*, Proc. IEEE Int. Conference Robotics and Automation, pp. 3121-3128, Leuven, Belgium, May 1998.

[DubrawskiA_1] Dubrawski, Artur and Siemiatkowska, Barbara, *A Method for Tracking Pose of a Mobile Robot Equipped with a Scanning Laser Range Finder*, Proc. IEEE Int. Conference Robotics and Automation, pp. 2519-2523, Leuven, Belgium, May 1998.

[DubrawskiA_2] Dubrawski, Artur and Siemiatkowska, Barbara, *A Neural Method for Self-Localization of a Mobile Robot Equipped with a 2-D Scanning Laser Range Finder*, Proc. 5th International Symposium on Intelligent Robotic Systems, 1997, pp. 159-168, Stockholm, Sweden, July 1997.

[GilP] Gil, Pedro, Pereira, João and Lopes, Francisco Martinho, *A Human-Computer Interface for a Robotic System*, 4th International Conference on Virtual Systems and Multimedia, VSMM'98, Gifu, Japan, November 1998.

[LeeversD] Leevers, David *et al*, *An Autonomous Sensor for 3D Reconstruction*, 3rd European Conference on Multimedia Applications, Services and Techniques, ECMAST 98, Berlin, Germany, May 98.

[LopesFM] Lopes, Francisco Martinho, Gil, Pedro and Pereira, João, *Client/Server Software Architecture for a Robotic System*, Proc. of the 3rd Portuguese Conference on Automatic Control, CONTROLO'98 – vol 1. Coimbra, Portugal, September 98.

[NgKia_1] Ng, Kia C., Sequeira, V., Butterfield, V., Hogg, D. C. and Gonçalves, J. M., *An Integrated Multi-Sensory System for Photo-Realistic 3D Scene Reconstruction*, Proc. of ISPRS International Symposium on "Real-Time Imaging and Dynamic Analysis", pp. 356-363, 1998.

[NgKia_2] Butterfield Stuart, Ng, K. C. and Hogg D., *The RESOLV Texture-Mapping Module*, Technical Report 97.48, School of Computer Studies, University of Leeds, UK, 1997.

[SequeiraV_1] Sequeira, Vitor, *Active Range Sensing for Three Dimensional Environment Reconstruction*, *Ph.D. Thesis*, Instituto Superior Técnico, Technical University of Lisbon, Portugal, December 1996.

[SequeiraV_2] Sequeira, Vitor, Ng, Kia, Butterfield, Stuart, Gonçalves, João M. and Hogg, D. C., *Three-Dimensional Textured Models of Indoor Scenes from Composite Range and Video Images*. In R.N. Ellson and J.H. Nurre, editors, Proc. SPIE, Three-Dimensional Image Capture and Applications, Vol. 3313, 1998.

[VestliS] Vestli, Sjur, *Fast, accurate and robust estimation of mobile robot position and orientation, Ph.D Thesis*, Eidgenössische Technische Hochschule Zürich, Switzerland, December 1995.

[WeckesserP] Weckesser, P. and Dillman, R., *Modelling Unknown Environments with a Mobile Robot*, Proc. Symposium on Intelligent Robotic Systems SIRS'97, pp. 69-76, Stockholm, July 1997.

Appendix A

Resumo da dissertação

Nesta dissertação apresenta-se uma nova solução para a localização de robots móveis a duas dimensões em ambientes construídos, baseada em medidas de distância laser e mapas reconstruídos com o mesmo sistema laser. As principais novidades desta solução são um algoritmo de localização que não requer qualquer estimativa inicial de postura e uma estrutura composta, com dois algoritmos em cascata e um módulo de validação, que procura compensar as limitações de cada elemento.

A.1 Introdução

A Localização é um elemento essencial na generalidade das plataformas móveis dado que é necessário ter uma estimativa rigorosa da posição e orientação em relação ao mapa do ambiente onde opera a palataforma para cumprir a maioria das missões. A solução proposta foi desenvolvida no âmbito do projecto RESOLV, cujo objectivo principal é criar modelos reconstruídos tridimensionais (3D) de ambientes interiores estruturados.

Este projecto utiliza um sensor laser com um dispositivo de varrimento (*laser range scanner*) para medir distâncias aos elementos da cena e uma câmara de video para adquirir as texturas da cena e assim criar um ambiente foto-realista. O sistema RESOLV orienta-se para a construção civil, arqueologia, criação de visitas virtuais a museus ou outros monumentos, entre outras possíveis aplicações.

Os sensores de aquisição são embarcados numa plataforma móvel com duas rodas motrizes, que pode deslocar-se autonomamente ou ser tele-operada através de uma ligação Internet rádio. Os sensores disponíveis para localização são: o sensor laser, a câmara de video, um anel de ultra-sons em torno da plataforma móvel para detecção de obstáculos próximos e um sistema de odometria que mede o trajecto percorrido a partir da rotação das rodas. Além da configuração base com uma plataforma móvel, foram criadas outras configurações mais simples, incluindo apenas o sensor laser e a câmara de video. Por isso, a Localização deverá requerer apenas os sensores presentes nestas últimas, de forma a poder ser utilizada em todas as configurações.

Como o mapa está definido em termos de distâncias métricas, o sensor laser foi escolhido como sensor principal para localização. Acessoriamente, a odometria é utilizada

quando disponível. Finalmente criou-se também a possibilidade de um operador manual introduzir uma medida de localização obtida por outra fonte externa. O mapa a duas dimensões usado para localização pode ser calculado a partir dos modelos 3D reconstruídos ou criado a partir de uma planta arquitectónica digitalizada ou criada num programa de CAD (Computer Aided Design).

O problema de Localização proposto é formulado nos seguintes termos: "Calcular a melhor estimativa da posição (x, y) e orientação (θ) para um sistema autónomo num ambiente horizontal por troços usando um sensor Laser e odometria, caso esta esteja disponível. O mapa de referência é baseado numa descrição parcial do ambiente que é incrementada à medida que o robot progride na cena".

A principal novidade da solução proposta em relação a outras soluçõs estudadas é a dispensa de duas condições: não é necessária uma estimativa *a priori* e o mapa do mundo pode ser quase todo desconhecido. A solução proposta combina três módulos. O primeiro é um algoritmo de extracção e emparelhamento de elementos característicos do ambiente que não usa qualquer estimativa inicial. O segundo módulo define uma medida de verosimilhança que afere a precisão das estimativas. O terceiro módulo baseia-se numa estimativa inicial refinada por um algoritmo que usa transformações de coordenadas ou aproximações recursivas.

A.1.1 O projecto RESOLV

O objectivo do projecto **RESOLV**-**RE**construction using **S**canned **L**aser and **V**ideo é criar modelos 3D de ambientes interiores estruturados, metricamente correctos e foto-realistas, usando imagens de distância laser e video. O sistema deve ser tele-operado através da Internet e os resultados devem ser igualmente acessíveis por Internet [LeeversD]. Os modelos, tal como o apresentado na Figura 1a, são expressos na linguagem VRML e podem ser acedidos e explorados com um *browser*. O sensor principal do projecto RESOLV é apresentado na Figura 1b, incluindo o laser à esquerda, a câmara de vídeo à direita e a cabeça motorizada sob os anteriores. Estes elementos estão ligados a um computador que controla os processos mecânicos, faz a reconstrução em tempo real e liga-se à Internet com um servidor Web e uma ligação rádio.

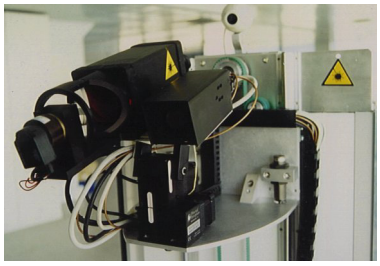


Figura 1– (a) pormenor de um modelo reconstruído, (b) sensor principal com laser e video, (c) sistema autónomo

Para reconstruir um ambiente complexo são necessários vários pontos de aquisição para preencher áreas oclusas e manter um nível de pormenor uniforme. Para tal, o sistema deve ser deslocado entre pontos de aquisição sucessivos. Uma plataforma móvel com rodas (Figura 1c) é o meio ideal para a tarefa. O conjunto do robot móvel e do sistema de reconstrução foi chamado **AEST**- **A**utonomous **E**nvironment **S**ensor for **T**elepresence.

Os dados disponíveis para a localização são um perfil espacial horizontal a duas dimensões (2D) do ambiente medido com o laser e os dados de odometria, caso o sistema use a plataforma móvel, e um mapa do ambiente. A natureza da missão dificulta a Localização pois os mapas são frequentemente muito incompletos em torno do AEST, já que a reconstrução 3D está ainda em curso. O algoritmo de Localização deve calcular uma estimativa de posição e orientação (postura (x, y, θ)) com menos de 0.05m de erro de posição e menos de 0.02rad(1º) de erro de orientação, dentro das especificações do ambiento de trabalho pormenorizadas em [LeeversD] e [SequeiraV_2]. A solução proposta cumpre estes requisitos.

A.1.2 O estado da arte

Diversos sistemas de localização baseados em sensores laser foram desenvolvidos no passado recente. Contudo, as possibilidades e aplicação destas soluções foram condicionadas pelas especificações do projecto RESOLV. Se fosse possível instalar balizas de sinalização ou sensores adicionais, várias soluções conhecidas seriam possíveis. Dadas as condições impostas, o leque de possibilidades estreitou-se. De seguida, algumas das soluções estudadas no curso deste trabalho serão brevemente descritas.

A solução apresentada por [DubrawskiA_1] baseia-se na extracção de linhas de um perfil 2D adquirido com um sensor laser. As linhas extraídas definem um histograma de direcções. Se um novo perfil fôr adquirido na vizinhança do primeiro a rotação do sensor pode ser estimada por correlação dos histogramas enquanto a translação pode ser estimada a partir da variação dos parâmetros que definem as rectas extraídas.

Uma classe de soluções admite que o mapa é conhecido *a priori*. Em [CrowleyJ], o ambiente é adquirido deslocando o sensor laser sobre uma grelha regular para criar uma descrição sistemática e completa do ambiente. Cada medida num perfil de N medidas é representada como uma dimensão num espaço de dimensão N. A expansão de Karhunen-Loève é utilizada para reduzir este espaço a um subespaço linear de dimensão mínima. Durante a operação, um perfil adquirido no ambiente poderá ser expresso como combinação linear da base do subespaço e, usando uma transformação inversa, pode calcular-se a postura do sensor.

Um método mais próximo das características do projecto RESOLV é apresentado em [VestliS]. O perfil de distância 2D adquirido com o sensor laser é transformado em segmentos de recta e arcos circulares através de uma tranformação geométrica associada a uma rede neuronal. Os elementos extraídos do ambiente são utilizados como balizas naturais. A partir de uma estimativa da posição inicial é possível emparelhar um

elemento do mapa com o elemento correspondente no perfil laser. Emparelhando múltiplos elementos do mapa e do perfil laser é possível estimar a postura do sensor com grande precisão.

Um trabalho semelhante [ArrasK] acrescenta a identificação de balizas artificiais e uma análise estatístic aturada dos dados laser, com ênfase na reflectância. O método proposto baseia-se no Filtro de Kalman Estendido (EKF) que inclui balizas naturais e artificiais. Estas duas abordagens ([VestliS] e [ArrasK]) implicam a localização contínua, quase em tempo real, um modelo do mundo de alta qualidade e um sistema de odometria para estimar os movimentos incrementais.

Em [WeckesserP] apresenta-se uma solução para o mapeamento em três dimensões (3D). Este trabalho persegue objectivos semelhantes aos do RESOLV mas usa um sensor formado por três câmaras de vídeo para extracção dos dados de distância. O método baseia-se na extracção das arestas das superfícies, sendo essas arestas comparadas com o modelo do mundo e um critério baseado na distância de Mahalanobis é usado para encontrar uma correspondência entre os dados de distância e o mapa.

Embora as soluções propostas em [ArrasK], [VestliS] e [WeckesserP] pudessem ser adaptadas ao projecto RESOLV, será demonstrado que a solução ora proposta cumpre as especificações do projecto e permite relaxar alguns dos requisitos das soluções propostas, nomeadamente a necessidade de uma estimativa inicial e odometria.

A.1.3 O módulo de Localização

O módulo de Localização é caracterizado pelo seu interface e conteúdo interno. O primeiro é determinado pelo contexto do projecto RESOLV, enquanto o segundo é de elaboração livre, desde que compatível com o interface.

Os dados de entrada do módulo de Localização são: um perfil 2D de distância obtido com um sensor laser, uma estimativa da postura do sistema obtida por odometria, disponível apenas no caso da plataforma móvel e um mapa do mundo. O perfil laser tem uma amplitude angular variável, dependendo do laser e também da construção mecânica do dispositivo. Nas plataformas actuais, o ângulo de visão oscila entre os 3.8rad (220º) e os 4.4rad (250º). A estimativa de odometria inclui os valores de (x, y, θ) calculados incrementalmente a partir da última estimativa de postura calculada com recurso aos dados laser. A princípio o mapa era constituído pela descrição das arestas das superfícies reconstruídas pelos módulos de reconstrução 3D. Contudo, durante o curso do projecto, constatou-se que era vantajoso diferir a fase de reconstrução até ao fim da aquisição. Nestes casos o mapa reconstruído foi substituído por um mapa a duas dimensões (2D), definido *a priori*.

Os dados de saída do módulo de Localização são uma estimativa da postura do sistema e uma indicação da condição de terminação do algoritmo. A solução proposta adiciona a estes dados uma lista de posturas possíveis com o erro associado e diagramas que ilustram a localização do sistema face ao mapa.

Após o estudo das referências citadas, concluiu-se que um só algoritmo não bastaria para resolver os requisitos do projecto RESOLV. A solução proposta baseia-se em dois algoritmos diferentes, aplicados em cascata, associados a um teste de validação que quantifica a verosimilhança da estimativa (Figura 2).

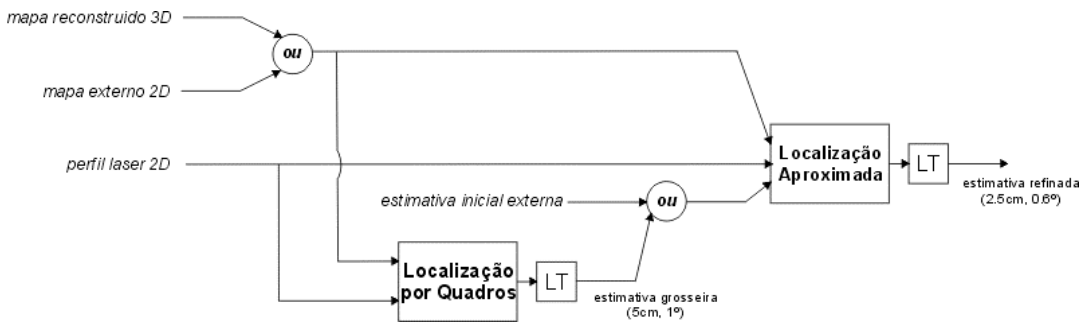


Figura 2 – A sequência dos algoritmos

De acordo com a figura, os dois algoritmos trabalham os mesmos dados. O primeiro algoritmo, chamado Localização por Quadros (*Frame Localisation*), calcula uma estimativa "grosseira" da postura sem uso de estimativa inicial. Este algoritmo pode ser substituído por uma estimativa inicial externa, fornecida pela odometria ou por outro meio qualquer, quando tal fôr possível. O segundo algoritmo, chamado Localização Aproximada (*Approximate Localisation*) baseia-se numa estimativa inicial para calcular uma estimativa refinada.

Nos dois casos a qualidade das estimativas é aferida por um teste de verosimilhança (Likelihood Test, LT) que indica a distância entre o perfil de laser e o mapa correspondente assumindo que o sistema se encontra na postura estimada.

A.2 Localização por Quadros

A Localização por Quadros baseia-se na análise do perfil do laser e do mapa dos quais se extraem segmentos de recta (ver Figura 3). De seguida, estes segmentos são agrupados dois a dois de modo a formar balizas naturais, chamadas quadros. O processo é repetido até que um número suficiente de quadros tenha sido criado a partir do mapa e a partir do perfil laser. Uma vez identificados os dois conjuntos de quadros, procura-se estabelecer uma correspondência entre um quadro do mapa e um quadro do perfil laser. Quando se establece essa correspondência, obtém-se uma postura que traduz uma possível relação entre os referenciais do mapa (associado ao quadro do mapa) e do robot (associado ao quadro do perfil laser). Repetindo o processo para um número suficiente de quadros cria-se uma constelação de posturas possíveis. A última fase do algoritmo é o agrupamento das posturas vizinhas e a sua substituição por uma postura equivalente.

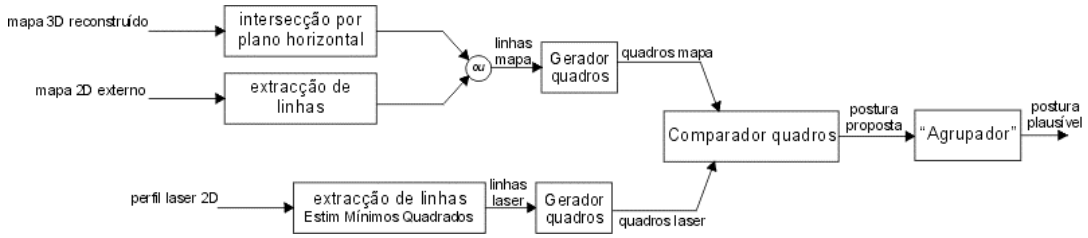
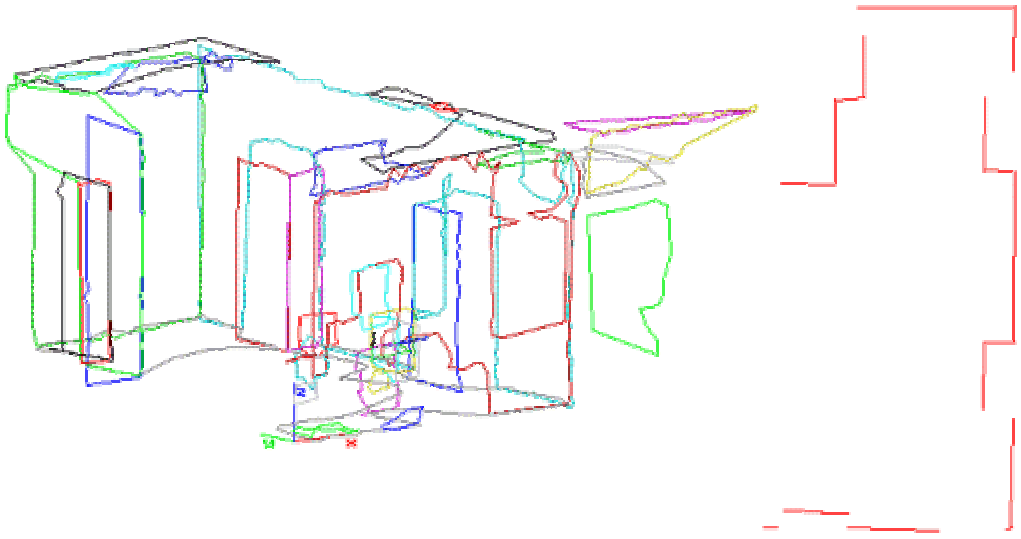


Figura 3 – A sequência dos algoritmos

O método para a extracção de linhas do mapa depende da natureza do mapa. Quando a reconstrução 3D é efectuada durante a missão de aquisição de dados, o mapa usado é fornecido pelos módulos de Reconstrução 3D e consiste na descrição dos contornos das superfícies tridimensionais calculadas no processo (Figura 4a). Para extrair um mapa a duas dimensões a partir destes dados basta intersectar o conjunto de contornos tridimensionais por um plano horizontal definido à altura do sensor laser.



(a) (b)

Figura 4 – A partir dos contornos 3D (a) são extraídas as linhas do mapa (b)

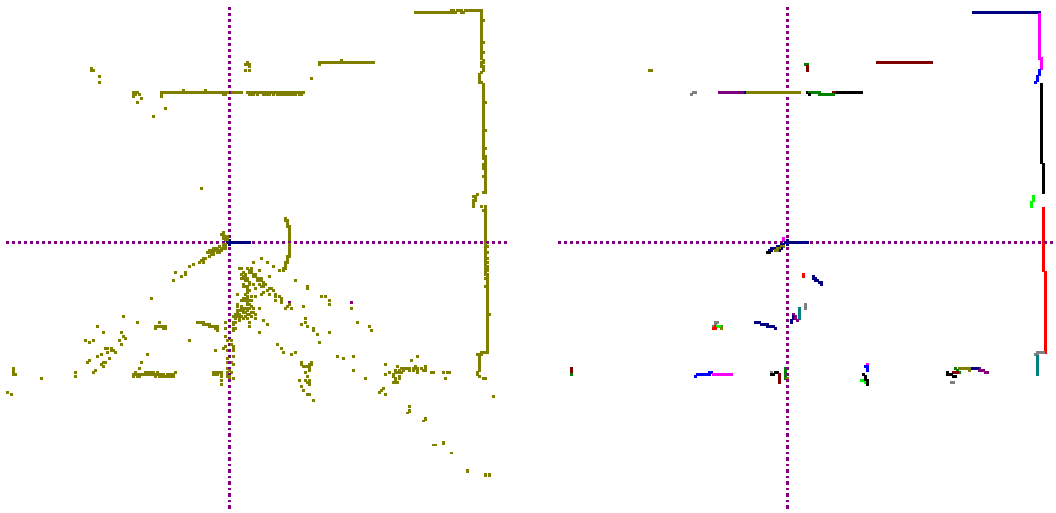
Nesta intersecção considera-se que as superfícies são aproximadamente planas, o que pode não ser exacto caso o ambiente inclua superfícies biquadráticas de curvatura significativa. A intersecção entre os contornos e o plano horizontal exprime-se por um conjunto de segmentos de recta definidos entre os pontos onde cada contorno intersecta o plano horizontal (Figura 4b). Esta lista de segmentos de recta constitui o conjunto de linhas do mapa.

Quando a reconstrução é efectuada após a aquisição dos dados, não existe um mapa de contornos disponível. Em sua substituição, utiliza-se um mapa obtido previamente a partir de uma planta arquitectónica ou por medição do espaço. Neste formato regista-se um corte horizontal do espaço a reconstruir, de preferência à altura do sensor laser. Na prática, o mapa é descrito por um conjunto de segmentos de recta, tal como no caso

anterior. A maior diferença reside, contudo, na completude e exactidão dos mapas. Os mapas baseados em plantas são normalmente muito incompletos pois representam o espaço antes da ocupação com mobiliário e outros elementos de ocupação do espaço. Por outro lado, os mapas criados a partir das medidas laser representam o ambiente circundante tal como o sensor laser o capturou minutos antes da localização.

A extracção das linhas do perfil laser baseia-se num algoritmo clássico de estimação de parâmetros, o estimador de mínimos quadrados (*LSE – Least Square Estimator*). A variante implementada restringe a estimação a segmentos de recta, que descrevem a maioria dos elementos presentes nos mapas e nos perfis laser. Esta solução foi adoptada pois permite uma grande simplicidade na implementação por computador, que foi feita de forma iterativa e recursiva, com ganhos significativos na eficiência de cálculo.

O perfil laser é constituído por uma sequência de medidas de distância (*range*) e brilho (*reflectance*), ordenadas por ângulo de varrimento horizontal (Figura 5a). O perfil é inicialmente dividido em segmentos de acordo com as descontinuidades entre amostras vizinhas e variações no brilho. Estas variações traduzem, normalmente, descontinuidades ou alterações das superfícies detectadas. De seguida, os segmentos do perfil ora definidos são percorridos de forma a detectar segmentos de recta, usando critérios muito exigentes. Os segmentos são assim fraccionados em múltiplos segmentos de recta, normalmente de pequena dimensão. A fase final do algoritmo de extracção de rectas é o agrupamento dos pequenos segmentos de recta consecutivos e paralelos em segmentos maiores (Figura 5b).



(a) (b)

Figura 5 – (a) exemplo de um perfil laser; (b) as linhas extraídas do perfil

Na Figura 5 o sensor laser é representado no centro da imagem associado a um referencial local. Note-se que neste exemplo muitas amostras não contribuem para qualquer recta, sobretudo no lado inferior do perfil. Isto resulta da presença de vidros em toda a parede inferior da sala.

A segunda fase do algoritmo de Localização por Quadros é a construção dos quadros (*frame*), uma nova entidade geométrica. Um quadro é definido por dois segmentos de recta concorrentes, chamados **eixos**, e o ponto onde os dois eixos se intersectam, a **origem**. Os quadros são definidos por dois tipos de parâmetros: cinco parâmetros locais, independentes do sistema de coordenadas em que os seus eixos se expressam (Figura 6a) e três parâmetros globais, que associam a origem do quadro ao referencial de coordenadas dos eixos (Figura 6b).

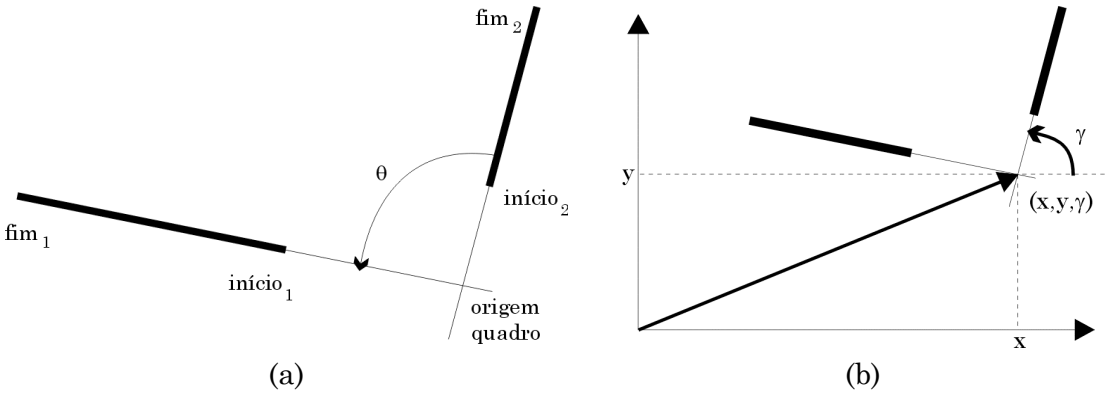


Figura 6 – Definição de um quadro: (a) parâmetros locais; (b) parâmetros globais

Os parâmetros $início_1$ e $início_2$ representam a distância da origem ao vértice mais próximo dos segmentos de recta enquanto fim_1 e fim_2 representam a distância ao vértice mais distante. O ângulo θ representa o ângulo interno entre os dois eixos.

Os parâmetros globais, (x, y, γ), definem a posição do quadro no referencial que lhe deu origem: os quadros do mapa estão referenciados nas coordenadas do mapa enquanto os quadros do scan estão referenciados a um sistema de coordenadas localizado no sensor laser, tal como o indicado na Figura 6. A partir dos parâmetros globais e locais é possível reconstituir a localização do segmento no referencial externo correspondente a cada quadro.

Para gerar os conjuntos de quadros, as linhas do mapa e as linhas do laser são ordenadas por tamanho decrescente, dado que os segmentos maiores são mais raros e encerram informação mais complexa sobre o ambiente. Adicionalmente, as linhas do laser mais compridas foram definidas a partir de um número maior de amostras e por isso a estimação de parâmetros é, à partida, mais rigorosa. A combinação de duas linhas concorrentes para formar um quadro é restrita aos casos em que as linhas são claramente concorrentes. Assim, só se cria um quadro quando o ângulo interno θ cumprir a condição $|\sin(\theta)| > 0.5$. Com esta condição procura-se eliminar os casos de eixos quase paralelos que conduzem a maiores erros no cálculo da origem. A geração das listas de quadros procede iterativamente e é sucedida pelo emparelhamento de quadros de mapa e de laser. Esta implementação permite acrescentar novos quadros a cada lista e continuar o algoritmo até se determinar um resultado fiável, sem prejuízo dos valores já estimados.

O algortimo de mapeamento parte do princípio que se uma determinada superfície faz parte do mapa então deve também ser visível no perfil laser. Como o mapa representa uma descrição tão completa quanto possível do cenário e o perfil laser está sujeito a oclusões e limitações mecânicas é razoável admitir que uma linha do laser esteja compreendida entre os limites da linha do mapa correspondente. Transpondo este princípio para os quadros, diz-se que dois quadros estão emparelhados quando os limites dos eixos do quadro do laser estão dentro dos limites dos eixos do quadro do mapa e os ângulos internos são aproximadamente iguais (Figura 7).

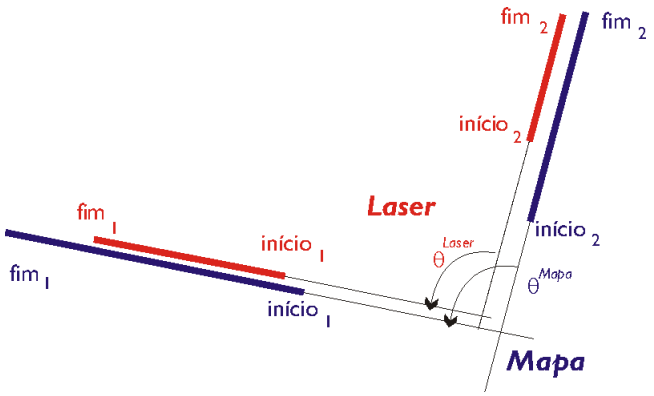


Figura 7 – Emparelhamento de quadros

As condições de emparelhamento são definidas por (A.1), onde δ_m e δ_{ang} são limiares de tolerância definidos pelo utilizador. A escolha dos parâmetros ideais depende do tipo de superfície e das características do sensor laser. Não obstante, as variações dos resultados em função de δ_m e δ_{ang} são pequenas. Por omissão, podem ser definidos $\delta_m = 0.03m$ e $\delta_{ang} = 0.005rad$ que satisfazem a generalidade das condições de operação encontradas.

$$\begin{cases} início_1^{Mapa} - \delta_m \leq início_1^{Laser} < fim_1^{Laser} \leq fim_1^{Mapa} + \delta_m \\ início_2^{Mapa} - \delta_m \leq início_2^{Laser} < fim_2^{Laser} \leq fim_2^{Mapa} + \delta_m \\ \left|\theta^{Mapa} - \theta^{Laser}\right| < \delta_{ang} \end{cases} \tag{A.1}$$

Quando se estabelece um emparelhamento entre um quadro do mapa e um quadro do laser, recorre-se aos parâmetros globais dos quadros para calcular a transformação de coordenadas que relaciona a origem dos dois quadros e que traduzirá a postura do referencial do sensor laser, expressa em coordenadas do mapa, (A.2) e (A.3).

$$\Delta\theta = \gamma^{Laser} - \gamma^{Mapa} \tag{A.2}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{Mapa} = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) & \Delta x \\ \sin(\Delta\theta) & \cos(\Delta\theta) & \Delta y \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{Laser} \tag{A.3}$$

A origem dos quadros do mapa está associada ao referencial do mundo enquanto a origem dos quadros do laser está associada ao referencial do robot. Resolvendo (A.3) em ordem a Δx e Δy, obtém-se o vector de transformação de coordenadas que representa a postura do sensor laser em coordenadas do mapa, (A.4).

$$(\Delta x, \Delta y, \Delta\theta) \Leftrightarrow {}^{Mapa}(x, y, \theta)_{Laser} . \tag{A.4}$$

O processo repete-se para todos os elementos das listas de quadros do mapa e do laser, dando origem a uma nuvem de posturas propostas no espaço de soluções (x, y, θ). A fase final do algoritmo de Localização por Quadros é um procedimento de classificação ponderada, ou agrupamento, que reúne todas as posturas que se encontram numa dada vizinhança, substituindo-as por uma postura equivalente. O peso atribuído a cada postura proposta é função do comprimento dos eixos do quadro do laser que lhe está associado e, em menor grau, da reflectância média associada aos segmentos de recta representados nos eixos.

O peso de um quadro é calculado pelo produto do peso atribuído a cada um dos seus eixos. Procura-se assim privilegiar os quadros com eixos proporcionados em prejuízo dos quadros com um eixo muito longo e outro muito curto, pois os eixos curtos estão mais sujeitos a erros na determinação dos parâmetros que definem as rectas.

O processo de agrupamento reúne todas as posturas numa dada vizinhança e substitui-as por uma postura equivalente, calculada em função do peso. O peso da postura equivalente é a soma dos pesos dos seus constituintes. O agrupamento de posturas é muito rigoroso, isto é, só as posturas muito próximas (tipicamente menos de 5cm, 0.005rad) são agregadas. Esta opção favorece os grupos muito homogéneos a expensas de uma lista de candidatos mais longa. Mesmo que a melhor solução não seja óbvia porque há dois ou mais grupos próximos que não se fundiram, a melhor estimativa será encontrada pelo Teste de Verosmilhança, aplicado a seguir.

O algoritmo de Localização por Quadros é intrinsecamente exponencial. Por isso, é necessário conservar o número de entidades geradas a partir dos dados laser e o esforço computacional tão baixos quanto possível. Para tal, todos os conjuntos são ordenados e o algoritmo é construído de forma iterativa. Só os algoritmo de conversão dos dados laser e dos mapas em linhas são executados exaustivamente. Uma vez transformados os dados em segmentos de recta, estes são ordenados por comprimento decrescente.

O gerador de quadros selecciona todas as linhas com comprimento acima de um dado limiar, para gerar os quadros de mapa e de laser. Os dois conjuntos de quadros são

ordenados por comprimento decrescente do seu eixo maior. O algoritmo de comparação e emparelhamento de quadros começa pelos primeiros elementos do conjunto, aqueles que, pelas suas dimensões representam uma maior fracção do ambiente. O algoritmo repete-se até que um padrão de distribuições surja no espaço de posturas. Se os quadros disponíveis se esgotarem antes de se detectar uma ou várias soluções dominantes, então o algoritmo de geração de quadros é relançado, agora com as rectas imediatamente a seguir às que foram usadas na primeira iteração. Atendendo à forma como o algoritmo está construído, os dados calculados anteriormente são conservados e acrescidos dos resultados associados aos novos quadros.

O algoritmo de Localização por Quadros produz uma pequena lista de posturas plausíveis, ordenadas por pesos decrescentes. Na maioria dos casos a primeira postura reúne mais de 50% do peso total. As excepções ocorrem sobretudo em ambientes irregulares, com mapas incompletos ou com vários eixos e orientações simétricos.

As estimativas calculadas pelo algoritmo de Localização por Quadros são baseadas em análises locais dos dados. O Teste de Verosimilhança faz uma comparação global dos dados, testando cada uma das soluções propostas. Se o teste validar mais do que uma solução, o resultado de mais alta verosimilhança será publicado como a estimativa da postura do robot.

A.3 Teste de Verosimilhança

As soluções calculadas pelo algoritmo de Localização por Quadros e por um dos algoritmos de Localização por Aproximação baseiam-se na comparação e emparelhamento de fracções dos dados. Trata-se de reunir os resultados de várias análises locais conducentes a um resultado coerente. Todavia, é necessária uma avaliação do mérito da solução, fazendo uma comparação extensiva do mapa e do perfil laser.

Para aferir o mérito das posturas propostas, usa-se como critério de verosimilhança a distribuição da distância entre cada um dos pontos do perfil laser e o mapa do mundo a duas dimensões, admitindo que o robot se localiza na postura proposta.

Para tal, o primeiro passo é converter o mapa do mundo, definido em segmentos de recta, num formato que o torne comparável ponto a ponto com o perfil laser. Para cada postura proposta, $p_i = (x_i, y_i, \theta_i)$, é criado um *perfil de laser simulado*, admitindo que o sistema de aquisição está localizado exactament em (x_i, y_i, θ_i). O princípio do perfil simulado será igualmente usado na Localização por Aproximação.

A.3.1 Perfil de Laser simulado

O perfil de laser simulado procura replicar as características principais de um

verdadeiro *laser scanner*. Descreve um varrimento horizontal circular, centrado no ponto indicado para a localização do sensor laser real; o varrimento é feito à altura estimada do sensor laser e tem a mesma resolução e amplitude angular (Figura 8).

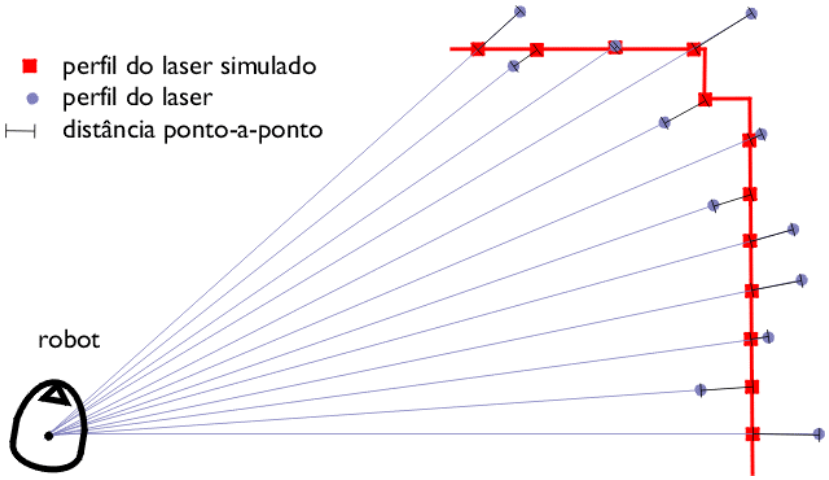


Figura 8 – Distância ponto-a-ponto

Calculando a diferença ponto a ponto entre a distância estimada das superfícies do mapa à postura do laser e a distância medida pelo sensor laser real, pode calcular-se um histograma das diferenças, agrupando as diferenças em classes discretas.

O perfil de Laser simulado apresenta limitações de duas naturezas diferentes: estatísticas da medida e correcção dos mapas.

As primeiras resultam da insuficiente modelação dos erros, nomeadamente a exclusão da estatística dos erros de distância na modelação das medidas do laser, o que poderia ter sido feito na sequência de ensaios de calibração de sensores laser reais e da exclusão da estatística dos erros associados à incidência nos materiais alvos. Neste caso é possível estudar estatisticamente os erros associados ao ângulo de incidência nos alvos mas não é possível incorporar a dependência das cores e texturas dos alvos, visto que eles são desconhecidos *a priori*. Sendo estes últimos os mais significativos, julgou-se de importância reduzida incluir os restantes.

O segundo tipo de limitações decorre da natureza e completude dos mapas. Se os mapas estiverem incompletos, o perfil simulado resultará alterado e impossível de compatibilizar com o perfil real. Este constrangimento é especialmente grave quando os mapas são calculados *offline*, baseados em plantas arquitectónicas. Nestes casos, é comum existirem diferenças muito extensas entre o conteúdo dos mapas e aquilo que o sensor laser mede no ambiente real.

O perfil de laser simulado baseia-se num modelo simplificado do sensor laser. Quando calculado na posição óptima, tem uma distribuição do erro face às medidas reais que se aproxima da característica de erro de distância do sensor laser.

O histograma típico de erro de distância de um laser, a sua "assinatura", é apresentado na Figura 9, onde cada coluna representa o número de amostras em cada ensaio com erro de distância no intervalo de $[x, x+1[$ unidades. O erro concentra-se nas pequenas distâncias, isto é, a maioria das amostras cai nas classes de 0-1cm e de 1-2cm, reduzindo-se progressivamente até aos 10cm. Acima destes limiares (válidos para os sensores usados no projecto RESOLV e outros semelhantes) o número de amostras é residual. A última classe concentra todas as amostras com um erro superior a um limiar fixo (neste caso, >19cm).

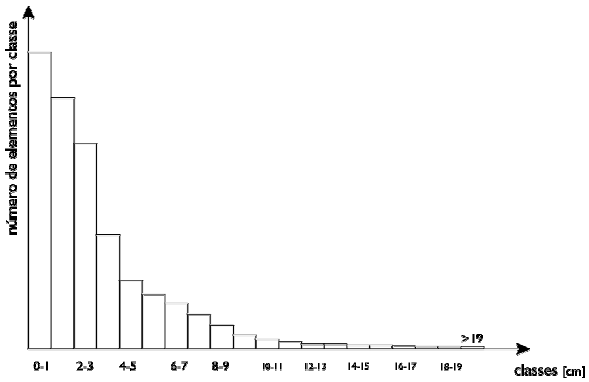


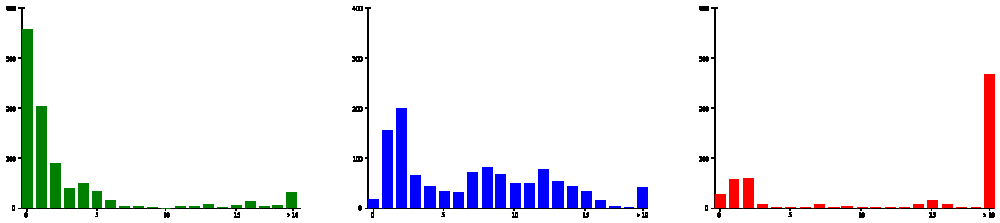
Figura 9 – Assinatura de um laser

A.3.2 Análise do erro e validação das soluções

Se a localização proposta coincidir com a localização real do robot, se o mapa fôr exacto e se os efeitos de variação das superfícies alvo (materiais, côr e textura) forem negligenciáveis, o histograma das diferenças entre os dois perfis traduzirá a distribuição de erro de distância do sensor laser. Nos casos reais, esta característica traduz um limite óptimo e define-se a melhor estimativa de localização na postura que se apresentar mais próxima dele.

Na Figura 10, apresentam-se três histogramas de verosimilhança definidos pelas diferenças entre perfis de laser reais e perfis simulados, admitindo três localizações diferentes para o sistema de aquisição. Na Figura 10a, o histograma coincide praticamente com a assinatura do laser, traduzindo uma localização bastante precisa; só o cálculo dos três momentos e a comparação com outras posturas candidatas permitirá concluir se se trata da estimativa de máxima verosimilhança. Na Figura 10b a maioria dos elementos encontra-se já em limites de erro reduzidos (< 15cm) mas o resultado pode ainda melhorar. Este é um resultado típico da Localização por Quadros: os elementos nas primeiras classes correspondem aos elementos constituintes dos quadros e que por isso apresentam uma diferença reduzida, enquanto os restantes elementos representam os elementos ignorados pelo algoritmo de Localização por Quadros. Neste cenário típico, a Localização por Aproximação conduzirá a um histograma de diferenças análogo ao da Figura 10a. Finalmente, a Figura 10c representa o histograma de uma localização errada.

Mais uma vez há alguns elementos próximos da origem mas a maioria dos pares encontram-se na última classa, com erros acima dos 19cm. Este histograma surge normalmente associado a falsas soluções calculadas pelo algoritmo de Localização por Quadros. Apenas uma pequena amostra dos perfis coincide e os restantes pares estão distantes. Os três histogramas são claramente diferentes, ilustrando a eficiência do Teste de Verosimilhança como discriminador das soluções propostas pelos algoritmos de localização.



(a) (b) (c)

Figura 10 – Três histogramas de verosimilhança

Ao comparar extensivamente os dois perfis, encontra-se alguns pares não comparáveis. Ora porque não existe mapa, ora porque o perfil laser tem um valor inválido, ora porque a distância entre eles é demasiado elevado, há pares que não são "emparelhados".

Este último critério é importante porque permite moderar o peso atribuído aos pontos distantes. Por exemplo, se num mapa faltar um objecto ou estiver um objecto a mais, a distância entre uma amostra do perfil laser e a sua correspondente no perfil isolado é muito grande e sem significado estatístico pois os dois perfis não são localmente comparáveis. Antes de calcular quaisquer momentos do histograma, torna-se conveniente restringir a análise aos segmentos dos perfis comparáveis. Isso é feito a partir da definição convencional de "pares emparalhados", que são aqueles cuja distância está abaixo de um limiar definido pelo operador.

A análise dos histogramas baseia-se nos seus momentos estatísticos, restritos aos pares emparelhados. No Teste de Verosimilhança usam-se os três momentos de ordem inferior. O momento de ordem 0 do histograma de distâncias associado à postura p_n representa o número de elementos, (A.5), onde NS é o número de classes (*slots*) do histograma. O momento de ordem 1 representa o valor esperado do histograma, (A.6), onde Δ_{slot} é a extensão de cada classe (tipicamente 1cm) e $\Delta_{\max}$ o valor equivalente da classe correspondente à maior distância.

$$ParesEmparelhados(p_n) = \sum_{s=0}^{NS-1} t_k \qquad \text{(A.5)}$$

$$ValorEsperado(p_n) = \frac{\sum_{k=0}^{NS-2} t_k \cdot \left(\Delta_{slot} \cdot \left(k + \frac{1}{2} \right) \right) + t_{NS-1} \cdot \Delta_{\max}}{ParesEmparelhados(p_n)} \tag{A.6}$$

$$Dispersão(p_n) = \frac{\sum_{k=0}^{NS-2} t_k \cdot \left(\Delta_{slot} \cdot \left(k + \frac{1}{2} \right) \right)^2 + \left(t_{NS-1} \cdot \Delta_{\max} \right)^2}{ParesEmparelhados(p_n)} \tag{A.7}$$

O momento de ordem 2, (A.7), representa a dispersão do histograma. Os três momentos permitem validar e classificar os histogramas. As soluções são validadas se o número de pares emparelhados, (A.5), ultrapassar um limiar mínimo de pares e o valor esperado, (A.6), não ultrapassar um limiar máximo de erro.

A comparação entre posturas candidatas recorre a um conjunto de testes comparativos dos momentos, seguindo uma sequência decrescente de relevância. Primeiro a comparação é feita ao nível do número de pares emparelhados, (A.5). Se o número de pares emparelhados fôr semelhante para diferentes posturas, procede-se à análise do valor esperado, (A.6). Se ainda assim não fôr possível classificar e ordenar claramente as posturas, procede-se à análise da dispersão, (A.7).

Para o processamento numérico de dados (como o algoritmo de Descida do Erro, descrito na próxima secção) foi criado um funcional de custo, (A.8), que combina o valor esperado com a fracção de pares válidos que foi possível emparelhar (o coeficiente α tem a dimensão de uma distância e representa o peso desta componente). Este funcional discrimina entre posturas que tenham valores esperados do erro semelhantes, mas representem fracções diferentes dos perfis a comparar, porque é mais verosímil uma postura com um valor esperado ligeiramente superior, se fôr baseada numa extensão significativamente maior dos perfis.

$$custo(p_n) = ValorEsperado(p_n) + \alpha \cdot \frac{ParesValidos(p_n) - ParesEmparelhados(p_n)}{ParesValidos(p_n)} \tag{A.8}$$

A principal limitação do Teste de Verosimilhança resulta da igualdade do peso atribuído à diferença entra cada par de pontos. Um único par de pontos distantes pesa muito mais na definição dos momentos do que dezenas de pares de pontos próximos. Como os pares distantes resultam não só dos erros dos algoritmos mas também de limitações dos mapas e dos sensores, a igualdade do peso dos pares conduz a uma sensibilidade excessiva àqueles pares distantes que não podem ser corrigidos pelos algoritmos. Corre-se então o risco de o Teste de Verosimilhança definir como estimativa óptima aquela que minimiza o custo associado a estes pares.

Embora seja impossível classificar *a priori* todos os pares, pode-se eliminar alguns dos pares de grande distância, com probabilidade mínima de erro. Se num ensaio há 90% dos pares com menos de 10cm de distância e a fracção de pares acima dos 30cm é inferior

a 2%, é provavelmente correcto ignorar estes últimos no cálculo de verosimilhança. Cria-se assim um segundo critério de validação que, afastando os pontos demasiado distantes, confere sensibilidade e ponderação estatística ao Teste de Verosimilhança. Novas implementações poderão incluir uma variação de peso em função da distância, para evitar descontinuidades no limiar de aceitação/recusa de um par de pontos.

Uma análise fina do Teste de Verosimilhança, em particular quando aplicado a experiências reais, revela o elevado grau de sensibilidade deste teste às variações de posturas, sendo possível discriminar nitidamente posturas candidatas situadas a menos de 1cm de distância. O Teste de Verosimilhança torna-se assim a bitola de aferição da qualidade das estimativas, substituindo as estatísticas tradicionais associadas à incerteza dos parâmetros estimados, de que se tinha abdicado no início do trabalho.

A.4 Localização por Aproximação

Após uma estimativa inicial de localização calculada pela Localização por Quadros ou obtida de fonte externa, validada e seleccionada pelo Teste de Verosimilhança, há que determinar qual a melhor estimativa possível numa vizinhança da estimativa conhecida. Foram desenvolvidos dois algoritmos para cumprir este objectivo: o algoritmo de Transformação de Referenciais, baseado na identificação de pontos notáveis nos perfis de laser (dados reais) e laser simulado e o algoritmo de Descida do Erro, onde se procura uma sequência de posturas que ofereça erros (ou custos) constantemente decrescentes.

Os dois algoritmos recorrem ao perfil de laser simulado, tal como foi apresentado na secção anterior.

Embora desempenhem a mesma função, os dois algoritmos seguem caminhos muito diferentes e oferecem um nível de desempenho diferente.

O algoritmo de Transformação de Referenciais calcula num intervalo de processamento muito curto uma estimativa actualizada a partir de um sub-conjunto muito limitado das amostras e, ao ser aplicado iterativamente, conduz a uma solução óptima ao fim de duas ou três iterações. As suas principais limitações são a restrição a pequenas correcções angulares, e a robustez reduzida na presença de perfis incompletos ou mapas incompletos, ou dados laser de baixa qualidade. A determinação da qualidade e precisão da postura actualizada é feita a partir do funcional de custo do Teste de Verosimilhança, baseado nos momentos de ordem inferior do histograma de diferenças.

O algoritmo de Descida do Erro calcula uma estimativa actualizada partindo da estimativa inicial e procurando todas as posturas vizinhas com um erro menor, de acordo com o Teste de Versomilhança. O algoritmo expande-se em todas as direcções que ofereçam melhorias incrementais no funcional de custo e detém-se quando não fôr possível encontrar vizinhos de custo inferior. Pela sua natureza de busca sistemática e de análise

extensiva dos perfis, é um algoritmo robusto, que nunca diverge. Contudo, obriga a um esforço computacional muito superior.

Em implementações automáticas o algoritmo de Descida do Erro é preferido pela sua elevada robustez e porque o poder computacional disponível actualmente permite efectuar todos os cálculos necessários em poucos segundos.

A.4.1 Transformação de Referenciais

O perfil do sensor laser é definido em coordenadas locais, tendo a origem do referencial na localização real do sistema de aquisição, medida em coordenadas do mundo, $(x_{robot}, y_{robot}, \theta_{robot})$. O perfil de laser simulado é calculado a partir do mapa do espaço pelo que está definido em coordenadas locais com a origem do referencial na estimativa inicial de postura, (x_I, y_I, θ_I). Se fôr possível associar um ponto do perfil real com o ponto correspondente do perfil simulado, a diferença das coordenadas traduz a relação entre os dois referenciais (Figura 11), expressa no vector de transformação, (x_P, y_P, θ_P).

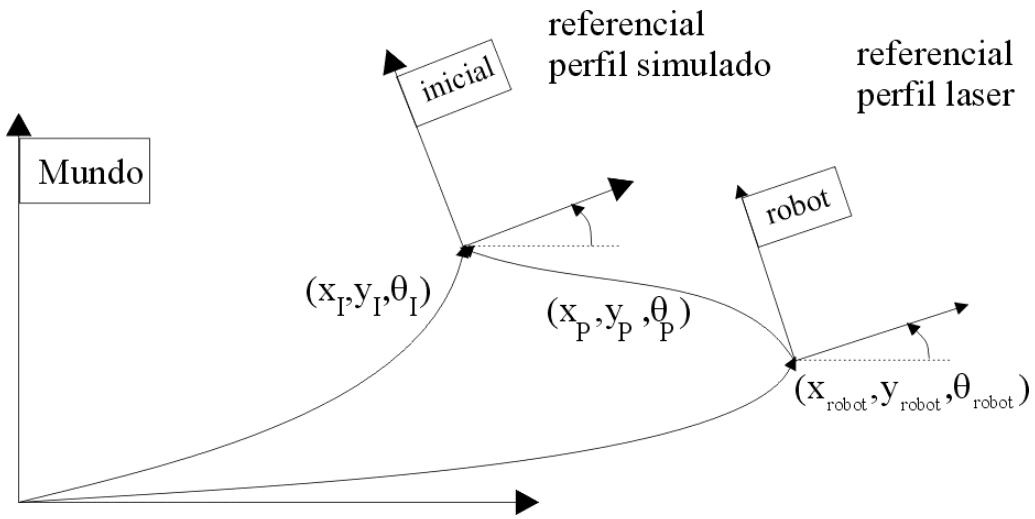


Figura 11 – Transformação de Referenciais

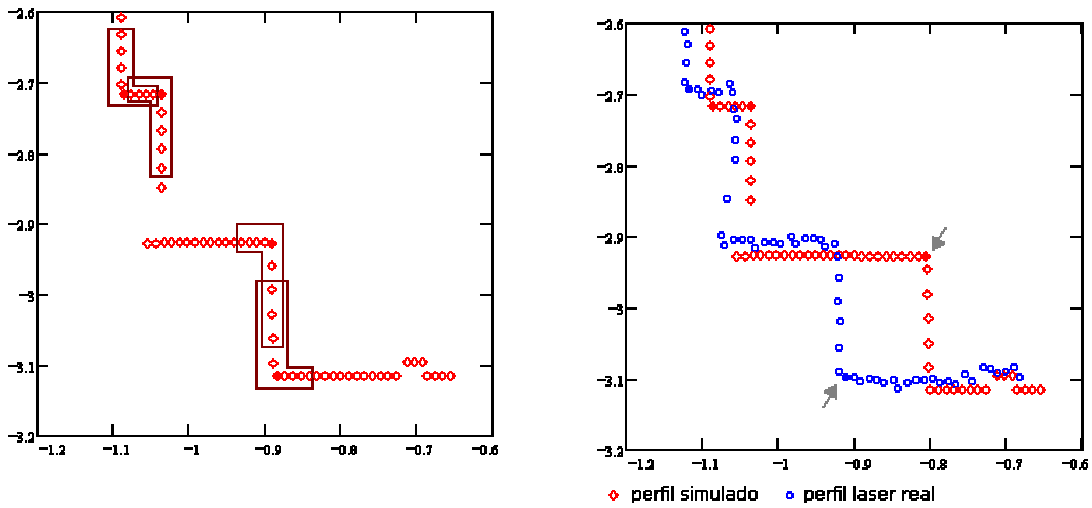
As incógnitas do problema são três, (x_P, y_P, θ_P). Dado que cada ponto dos perfis bidimensionais é caracterizado pelas suas duas coordenadas, $(x, y)^{robot}$ para o perfil laser e $(x, y)^{inicial}$ para o perfil simulado, um par de pontos notáveis correspondentes estabelecerá duas equações independentes, (A.9). São necessários um mínimo de dois pares de pontos notáveis para calcular a transformação de coordenadas completa.

$$\begin{bmatrix} x \\ y \end{bmatrix}^{robot} = \begin{bmatrix} \cos(\theta_P) & -\sin(\theta_P) \\ \sin(\theta_P) & \cos(\theta_P) \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}^{inicial} + \begin{bmatrix} x_P \\ y_P \end{bmatrix} \qquad \text{(A.9)}$$

Para estabelecer uma correspondência entre pontos dos dois perfis é necessário

identificar pontos notáveis correspondentes. Deve notar-se que, dada a natureza discreta dos perfis, os pontos não serão exactamente coincidentes. A um ponto notável do perfil simulado corresponderá o ponto contido no perfil real que mais próximo se encontra do ponto correspondente no perfil simulado. É importante também salientar que a comparação de perfis só é válida se a transformação entre os referenciais fôr pequena quando comparada com as distâncias às superfícies. Em particular, se o erro de orientação fôr significativo, o método proposto torna-se quase impossível.

Os melhores candidatos a pontos notáveis num perfil bidimensional são as intersecções entre as arestas, que correspondem normalmente aos cantos e esquinas do ambiente. Estes pontos podem ser facilmente detectados no perfil simulado recorrendo a uma janela móvel que percorre cada um dos perfis e calculando o produto interno de $2N+1$ amostras adjacentes, (Figura 12a), usando a amostra central como eixo de rotação.



(a) (b)

Figura 12 – Detecção de pontos notáveis: (a) janela móvel;
(b) pares de pontos notáveis

Se o produto interno apresentar um mínimo local, a amostra central corresponde a um ponto de intersecção de superfícies. Repetindo o método para todo o perfil simulado, encontrar-se-á um conjunto de pontos notáveis, que serão ordenados por produto interno crescente. De seguida, aplica-se o mesmo método ao perfil real, restrito à vizinhança dos pontos notáveis já encontrados. O resultado está ilustrado na Figura 12b, que representa um pormenor dos dois perfis comparados. Note-se a maior irregularidade do perfil real, porque o perfil simulado foi construído a partir de uma planta arquitectónica. O uso de uma planta obtida previamente e desactualizada é também responsável pela diferença dos perfis indicada pelas setas. Por coincidência, foi possível estabelecer uma correspondência nesta área apesar da diferença dos perfis. Se este par fôr incluído nas equações de determinação das coordenadas, os resultados serão mal condicionados, levando a uma má estimativa. Na maioria dos casos há um número de pares notáveis superior aos dois

necessários e por isso é possível escolher aqueles que conduzirão aos resultados melhor condicionados.

Uma vez seleccionados NL pares notáveis (com $NL \geq 2$), define-se um sistema sobre-dimensionado de 2NL equações a 3 incógnitas. Reordenando os termos obtém-se o sistema de equações (A.10).

$$\begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \\ x_{NL} \\ y_{NL} \end{bmatrix}^{robot} = \begin{bmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ x_2 & -y_2 & 1 & 0 \\ y_2 & x_2 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_{NL} & -y_{NL} & 1 & 0 \\ y_{NL} & x_{NL} & 0 & 1 \end{bmatrix}^{inicial} \cdot \begin{bmatrix} \cos(\theta_P) \\ \sin(\theta_P) \\ x_P \\ y_P \end{bmatrix} \tag{A.10}$$

Admitindo que o erro de orientação é pequeno ($\theta_P \approx 0$), o sistema pode ser linearizado em torno da origem. Reordenando os termos, o sistema linearizado exprime-se como (A.11) . A solução do sistema linearizado (A.11) pode ser obtida expeditamente por métodos algébricos convencionais.

$$\begin{bmatrix} x_1^{robot} - x_1^{inicial} \\ y_1^{robot} - y_1^{inicial} \\ x_2^{robot} - x_2^{inicial} \\ y_2^{robot} - y_2^{inicial} \\ \vdots \\ x_{NL}^{robot} - x_{NL}^{inicial} \\ y_{NL}^{robot} - y_{NL}^{inicial} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -y_1^{inicial} \\ 0 & 1 & x_1^{inicial} \\ 1 & 0 & -y_2^{inicial} \\ 0 & 1 & x_2^{inicial} \\ \vdots & \vdots & \vdots \\ 1 & 0 & -y_{NL}^{inicial} \\ 0 & 1 & x_{NL}^{inicial} \end{bmatrix} \cdot \begin{bmatrix} x_P \\ y_P \\ \theta_P \end{bmatrix} \tag{A.11}$$

O sistema (A.11) é do tipo (A.12), e pode ser resolvido na forma (A.13), correspondente a um sistema de três equações lineares a três incógnitas, com as matrizes de coeficientes definidas em (A.14) e (A.15).

$$M_{(3x2NL)} \cdot \begin{bmatrix} x_P \\ y_P \\ \theta_P \end{bmatrix} = Q_{(2NLx1)} \tag{A.12}$$

$$M^T_{(3\times 2NL)} \cdot M_{(2NL\times 3)} \cdot \begin{bmatrix} x_P \\ y_P \\ \theta_P \end{bmatrix} = M^T_{(3\times 2NL)} \cdot Q_{(2NL\times 1)} \tag{A.13}$$

$$M^T M = \begin{bmatrix} NL & 0 & -\sum_{i=1}^{NL} y_i^{inicial} \\ 0 & NL & \sum_{i=1}^{NL} x_i^{inicial} \\ -\sum_{i=1}^{NL} y_i^{inicial} & \sum_{i=1}^{NL} x_i^{inicial} & \sum_{i=1}^{NL} \left[\left(x_i^{inicial}\right)^2 + \left(y_i^{inicial}\right)^2 \right] \end{bmatrix}_{(3\times 3)} \tag{A.14}$$

$$M^T Q = \begin{bmatrix} \sum_{i=1}^{NL} (x_i^{robot} - x_i^{inicial}) \\ \sum_{i=1}^{NL} (y_i^{robot} - y_i^{inicial}) \\ \sum_{i=1}^{NL} \left[x_i^{inicial} \cdot (y_i^{robot} - y_i^{inicial}) - y_i^{inicial} \cdot (x_i^{robot} - x_i^{inicial}) \right] \end{bmatrix}_{(3\times 1)} \tag{A.15}$$

O resultado é o conjunto das três variáveis de transformação de coordenadas, (x_P, y_P, θ_P). Para obter a estimativa actualizada da postura do robot $(x_{robot}, y_{robot}, \theta_{robot})$, a partir da estimativa inicial, (x_I, y_I, θ_I), basta aplicar a transformação (A.16).

$$\begin{gathered} \theta_{robot} = \theta_I + \theta_P \\ \begin{bmatrix} x_{robot} \\ y_{robot} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta_P) & -\sin(\theta_P) & x_P \\ \sin(\theta_P) & \cos(\theta_P) & y_P \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_I \\ y_I \\ 1 \end{bmatrix} \end{gathered} \tag{A.16}$$

A.4.2 Descida do Erro

O algoritmo de Descida do Erro parte de um princípio muito simples: dado o custo, (A.8), da estimativa inicial, procurar na vizinhança outras posturas de custo inferior. Se tais posturas existirem, iterar o processo a partir dos novos candidatos até que não haja quaisquer posturas vizinhas de posturas candidatas com custos inferiores.

Ao contrário do algoritmo de Localização por Quadros e de Transformação de Referenciais, é um algoritmo que implica um processamento maciço de dados: para cada postura candidata é necessário efectuar um teste de verosimilhança, comparando extensivamente os dois perfis, e calcular os momentos associados ao histograma das diferenças. Para reduzir o esforço computacional, é crucial escolher uma distância óptima entre posturas vizinhas. A solução adoptada implementa um passo variável: começa-se com um passo grande e quando já não é possível obter reduções de custo, divide-se o passo por dois, recomeçando o algoritmo a partir dos candidatos anteriores.

Os pontos fortes do algoritmo de Descida do Erro são a robustez e a precisão. Ao procurar uma redução de custo o algoritmo nunca diverge – no sentido em que nunca dá piores resultados do que aqueles que recebeu. Por outro lado, como não são extraídas quaisquer características geométricas dos mapas ou dos ambientes reconstruídos e/ou do perfil laser, é especialmente adaptado para condições de operação difíceis, onde a caracterização do ambiente é deficiente. O algoritmo é especialmente preciso porque

percorre uma vizinhança da postura candidata, usando uma resolução mais fina nas áreas mais promissoras. Se a estimativa inicial fôr boa e o número de iterações suficiente, a distância entre duas posturas candidatas vizinhas é muito pequena, oferecendo por isso grande precisão.

O principal óbice do algoritmo de Descida do Erro é o custo computacional. Corre-se também o risco de sobre especialização em mínimos locais quando a postura estimada é conduzida para um ponto onde a redução do custo associado a um número reduzido de pares de pontos distantes se sobrepõe à degradação da estimação de dezenas de pares de pontos próximos. A melhor forma de evitar este erro é reduzir o limiar de aceitação dos pares emparelhados ou modular o peso da distância entre pares em função dessa mesma distância, reduzindo-o para as maiores distâncias.

O algoritmo de Descida do Erro desenvolve-se como uma busca orientada num grafo cujos nós são as posturas candidatas. A raiz do grafo é a estimativa inicial, (x_I, y_I, θ_I). De seguida, define-se um conjunto de posturas vizinhas de acordo com uma grelha regular num espaço a três dimensões independentes, X, Y, θ, com espaçamentos entre células de $\delta_X, \delta_Y = \delta_X$ e δ_θ. Na Figura 13 apresentam-se duas grelhas possíveis. A primeira corresponde à variação de apenas uma coordenada entre posturas vizinhas e a segunda à variação de uma ou duas coordenadas entre posturas vizinhas. Por razões de eficiência computacional, optou-se pela primeira.

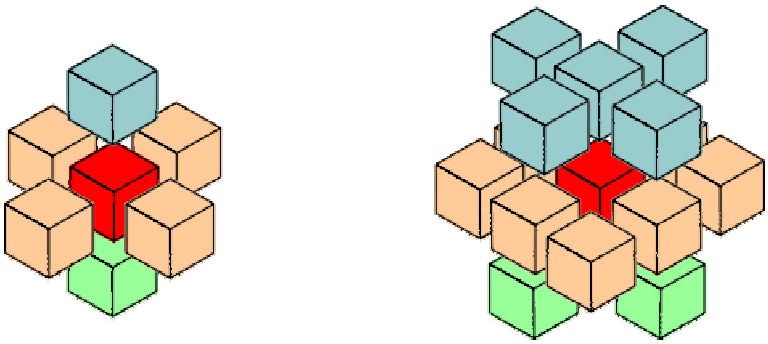


Figura 13 – Definição de vizinhanças

Após a definição da grelha o algoritmo segue a sequência indicada abaixo, para cada uma das posturas candidatas contidas no grafo.

1. Calcular o custo, (A.8), associado à postura candidata, $p_k = (x_k, y_k, \theta_k)$.
2. Determinar o conjunto de vizinhos.
3. Verificar se algum dos vizinhos ora considerados já foi incluído na lista de posturas candidatas anteriormente calculadas. Nestes casos, o vizinho repetido é ignorado.
4. Calcular o custo, (A.8), associado aos vizinhos restantes.
5. Adicionar todos os candidatos com um custo inferior a $custo(p_k)$ à lista de

posturas candidatas e marcar as novas posturas como candidatas a estudar.

6. Se nenhuma postura vizinha tiver um custo inferior a $custo(p_k)$, marcar a postura p_k, como um terminal do grafo.
7. Prosseguir para a postura candidata seguinte.

Quando o grafo de posturas candidatas foi completamente explorado para uma dada resolução do passo $\delta_X, \delta_Y = \delta_X$ e δ_θ, dividir o passo por dois e reiniciar o algoritmo, usando como raízes do grafo os terminais da iteração anterior.

O curso do algoritmo pode ser representado graficamente e em coordenadas inteiras (número de passos) como na Figura 14. Nesta figura representa-se apenas os caminhos que conduzem a mínimos locais e os nós terminais. O valor típico dos passos é de $\delta_X = \delta_Y = 0.001m$ e $\delta_\theta = 0.0001rad$.

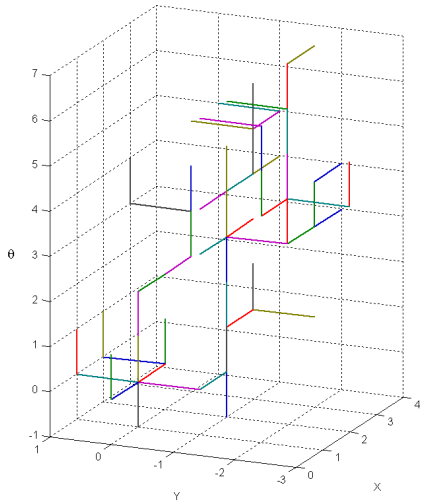


Figura 14 – Exemplo do algoritmo de Descida do Erro

O resultado do algoritmo de Descida do Erro obtém-se a partir da lista de nós terminais do grafo. De entre estes, a melhor estimativa da localização corresponde ao nó com menor custo.

Os dois algoritmos de Localização por Aproximação seguem abordagens opostas e por isso são adaptados a situações opostas: enquanto a Transformação de Referenciais analisa, interpreta e condensa os dados num conjunto reduzido de variáveis que alimentam um algoritmo de custo computacional modesto, a Descida do Erro trabalha extensivamente os dados dos perfis do sensor laser e do mapa e repete numerosas vezes uma sequência de processamento dos dados em bruto, sem interpretações topológicas ou geométricas.

A Transformação de Referenciais adapta-se a problemas bem condicionados, com boas medidas de laser e mapas completos e a Descida do Erro aos problemas mal condicionados em ambientes incompletamente descritos ou de descrição difícil. Dado que a Descida do Erro oferece grande robustez com um custo computacional que, sendo centenas

de vezes superior ao exigido para a Transformação de Referenciais, é aceitável para a missão de reconstrução, foi escolhido este último algoritmo na implementação das aplicações práticas da Localização.

A.5 Conclusões

O objectivo principal deste trabalho de investigação era o desenvolvimento de uma solução de localização baseada em medidas de distância laser que não requeresse uma estimativa inicial de posição e estivesse adaptada a mapas incompletos ou em construção. Este objectivo foi cumprido com o algoritmo de Localização por Quadros. Para avaliar a qualidade de estimativa foi desenvolvido o Teste de Verosimilhança. O terceiro passo no trabalho descrito foi o desenvolvimento de duas alternativas para a Localização por Aproximação, que tira partido de uma estimativa inicial para calcular uma estimativa refinada da postura do sistema de reconstrução RESOLV.

Os quatro algoritmos descritos cumprem na totalidade as especificações originais do projecto RESOLV e cumprem ainda quase todas as especificações adicionais, que surgiram no decurso do projecto. Os requisitos iniciais de erro de localização estipulavam um limite de 0.1m em posição e 0.017rad (1º) em orientação enquanto os valores típicos de erro se aproximam de 0.02m e 0.0034rad (0.2º), sendo possível obter menos de 0.01m e 0.001rad de erro em ambientes bem condicionados e com mapas correctos.

Entre as questões por resolver, há problemas em aberto externos ao enunciado original do problema, tais como a localização com o robot em movimento, a localização baseada em mapas gerados a partir de medidas laser bidimensionais e a localização em planos inclinados ou superfícies irregulares. Todavia, há dois problemas não tratados que se incluem no enunciado original: a representação e manipulação de superfícies não planas, mormente biquadráticas e a prova formal de que a estimativa encontrada pelos algoritmos de localização é a melhor estimativa global.

O primeiro problema foi relegado para segundo plano porque a fracção do ambiente descrita por superfícies biquadráticas de extensão modelizável é diminuta e a consideração de biquadráticas implicaria um aumento significativo do esforço computacional. Não obstante, o algoritmo de Localização por Quadros é compatível com curvas biquadráticas e os restantes algoritmos também, desde que os mapas incluam a descrição das superfícies biquadráticas.

O segundo problema não foi abordado e, formalmente, os algoritmos detectam apenas mínimos locais do erro de localização. Da inspecção visual dos diagramas conclui-se facilmente se o mínimo é local ou global, mas a prova formal implicaria a definição completa do perfil laser e do mapa do ambiente – o que é impossível nos termos do enunciado do problema ou, em alternativa, a inspecção exaustiva de uma grelha de posturas infinitesimalmente próximas. Por outro lado, os algoritmos de Localização por

Quadros e Descida do Erro calculam listas de mínimos locais, ordenados por verosimilhança decrescente, o que é de grande interesse prático em casos de ambientes mal definidos ou simétricos.

As direcções de pesquisa futura na área da Localização incluem os problemas em aberto mencionados atrás mas também a generalização do problema proposto. Por exemplo, seria interessante estender o algoritmo à localização a três dimensões (seis variáveis). Se se abandonasse a condição do sensor laser como sensor único ou primordial, novas possibilidades se abririam, nomeadamente com recurso a GPS para localização no exterior, ou combinando laser com vídeo ou ultransons para criar mapas de visibilidade e espaço livre e preencher as áreas que o laser não consegue adquirir.

A experiência do projecto RESOLV ensinou ao autor que as direcções de investigação nos projectos aplicados são motivados pelas especificações das missões e não pelas expectativas iniciais dos investigadores.

Este trabalho foi encarado num espírito de "cliente-servidor": o módulo de Localização é o servidor e os utilizadores e restantes módulos do sistema RESOLV são os clientes. A Localização deve cumprir os requisito dos seus clientes, adaptando-se aos sensores e equipamentos disponíveis para atingir um desempenho óptimo, sem acrescentar novos condicionamentos ao projecto. Não obstante, foi necessário acrescentar a restrição de deslocação em superfícies horizontais, o que foi considerado aceitável pelos restantes parceiros.

A solução de Localização adaptou-se à substituição de mapas reconstruídos actuais por mapas calculados a partir de mapas anteriores (normalmente insuficientes e desactualizados), adaptou-se às exigências de precisão acrescida, à restrição do campo de visão do sensor laser, à operação em cenários descritos de forma muito incompleta e de dimensões superiores ao alcance de medida do laser. A extensão das capacidades da Localização a ambientes mal estruturados deve-se sobretudo à robustez do Teste de Verosimilhança e da Descida do Erro.

Os algoritmos propostos foram submetidos a ensaio em condições de operação difíceis e o seu desempenho foi aferido pelos membros do projecto RESOLV. Fiel ao espírito "cliente-servidor", o esforço do autor prolongou-se até que estes estivessem completamente satisfeitos.

Appendix B

Line extraction from Map data

The method used to generate map data for localisation purposes depends on the map source. The primary map source is the 3D Reconstruction Algorithms mentioned in Chapter 1, Section 3 and documented in [SequeiraV_1] and [SequeiraV_2]. However, during the project, the convenience of operating without the 3D reconstruction modules arose as a major enhancement.

The reconstruction of a site, or some rooms in a site, is usually preceded by a detailed analysis and preparation. Sometimes, some type of site plans are available, produced by architects or other technicians. Therefore, it should be possible to translate the information in the plans to digital format, and use them for navigation purposes.

In addition, the Localisation solution is able to read plain text files with very little formatting, a necessary feature for the development and debugging phase. According to these requirements, a common format was developed to represent the map data.

The common format is described in the next Section. Section B.2 describes the extraction of map lines from the 3D Reconstruction data. Section B.3 describes the extraction of map lines from offline maps.

B.1 The Localisation Map format

The current implementation of Localisation only uses line segments on two dimensions. Therefore, the map description needs only to store line segment definitions. Should the Localisation be extended to other types of curves, such as biquadratic curves, the format must be extended as well to encompass the new types.

The proposed format is supported on a single text file for the whole environment description. The lines are described by their vertexes in the world coordinate reference. In usual environments, line segments are not isolated but linked to other neighbouring elements. Hence, it seemed appropriate to associate the lines in contours, both open or closed, encompassing nl lines or equivalently, $nl+1$ vertexes. In this format, an isolated line is represented by a contour with two vertexes only.

The data is stored on the map file according to the sequence:

1. Write the total number of contours, nl.

2. Write the number of vertexes in each contour in sequence. There is no particular order to sort the contours
3. Write the *(x,y)* vertexes of a contour in sequence. If the contour is closed, the last vertex has the same coordinates of the first one. There is no other method to discriminate between open and closed contours.

A sample of the line map file is presented in Table 1. It is part of the description of the room presented in Chapter 2, Section 5 and illustrated in Figure 1 below. All measurements are given in meters.

Data	Legend
5	Number of contours
61 5 5 5 5	List with the number of vertexes in each contour
…	(first contour omitted)
4.42 5.31 5.37 5.30 5.37 4.90 4.42 4.91 4.42 5.31	Second contour
3.36 5.31 4.34 5.31 4.34 4.91 3.36 4.91 3.36 5.31	Third contour
2.36 5.31 3.34 5.31 3.34 4.91 2.36 4.91 2.36 5.31	Fourth contour
0.125 0.125 0.475 0.125 0.475 0.475 0.125 0.475 0.125 0.125	Fifth contour

Table 1 – Sample of a Localisation Map file

No reflectance data is included. In addition, no confidence or data quality estimate is associated to the lines or the contours.

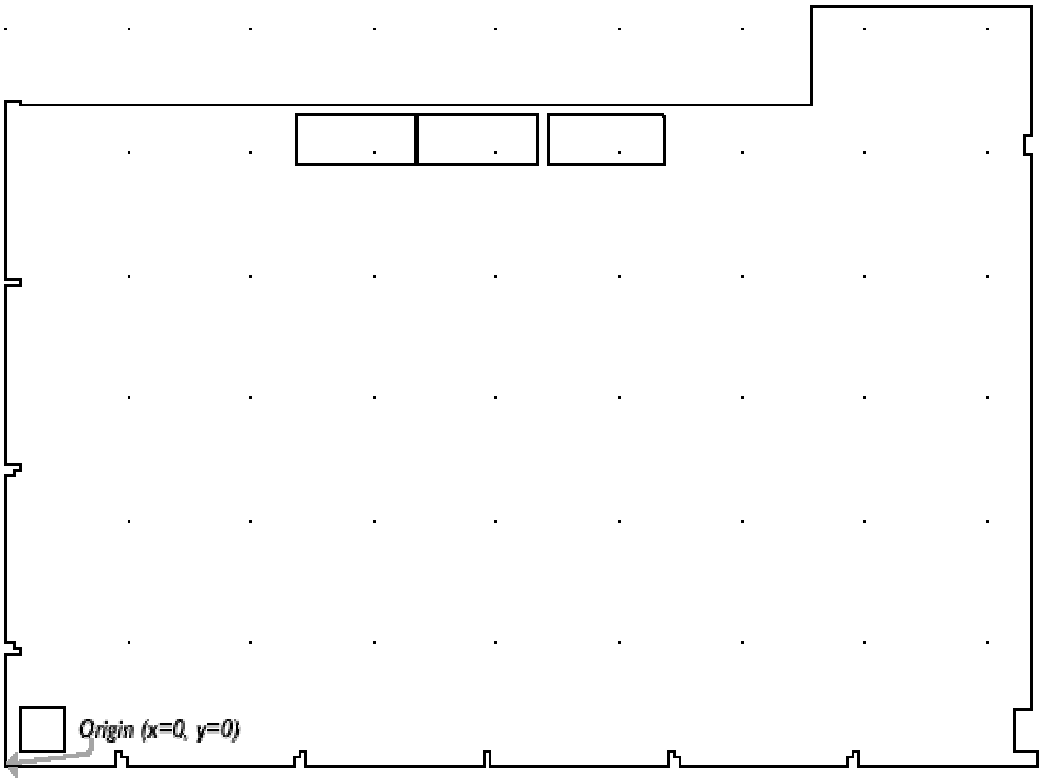
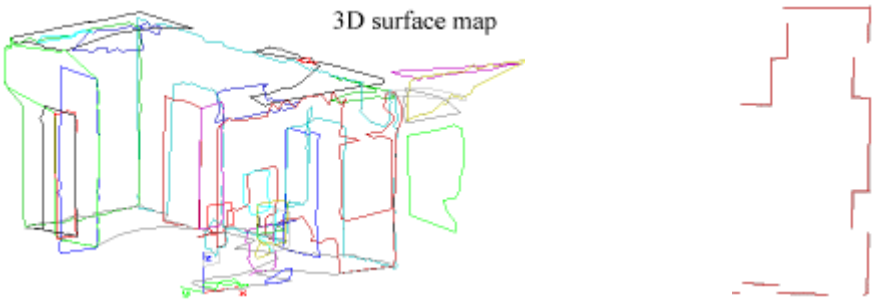


Figure 1 – Sample floor plan (based on Table 1 data)

B.2 Extraction of map lines from 3D Reconstruction data

The 3D Reconstruction data is represented in different forms, depending on the final application. For Localisation purposes, the reconstructed surfaces are described by their contour in three dimensions. Each surface is a set of (x, y, z) vertexes, which define its boundary. If the surfaces are nearly flat, the reconstructed surface may be approximated by its boundary.

Assuming this simplification, the 3D contour description is translated to two dimensions in order to meet the laser scan data characteristics. The two dimensional profile of the room is found at the intersection of the reconstructed surfaces with a horizontal plane located at the laser scanner height (Figure 2). In general, the surfaces are not convex. Hence, the boundary of a single surface may intersect the laser scanner plane multiple times, always in even numbers if the singular case of the vertex located exactly at the plane height is ignored.



(a) (b)

Figure 2 – A sample reconstructed room and its profile at $z = 1.2m$

The algorithm starts with a pre-processing phase to minimise the number of surfaces considered. Given a room description with ns nearly flat surfaces, each surface is enclosed in a contour with np (x, y, z) vertexes defined in the world coordinates. Consider also a laser scanner taking a horizontal profile at height $z = h$. The minimum and maximum height for each surface is computed from the z-coordinates. If z-maximum is below the scanning height h, or z-minimum is above h, the surface does not intersect the laser scanner plane. Therefore, it is ignored in the map generation algorithm.

The line extraction algorithm determines the locus of the intersection between the 3D surface and an horizontal plane located at the laser scan height. This will be the 2D map used in the Localisation. The algorithm performs the following loop on all the surfaces crossing height h:

1. Consider a contour with np vertexes and start at the first vertex, $p_i = (x_i, y_i, z_i)$ with $i = 1$.
2. If $((z_i < h) \wedge (z_{i+1} > h)) \vee ((z_i > h) \wedge (z_{i+1} < h))$, the line segment $[p_i, p_{i+1}]$ intersects the laser scanner plane. The intersection coordinates are computed from the line equations defined as $x(z)$ and $y(z)$, in equations (B.1), and computed at $x(h)$ and $y(h)$.

$$\begin{cases} x(z) = x_i + (z - z_i) \cdot \dfrac{x_{i+1} - x_i}{z_{i+1} - z_i} \\ \\ y(z) = y_i + (z - z_i) \cdot \dfrac{y_{i+1} - y_i}{y_{i+1} - y_i} \end{cases} \qquad \text{(B.1)}$$

3. The intersection coordinate is added to the two-dimension vertex list as $v_j = (x(h), y(h))$.
4. The algorithm resumes at step 2, until all vertexes are processed (the last iteration considers p_{np} and p_1 to close the contour).

When the contour analysis is complete, the algorithm proceeds to the next step, where the line segments that build the 2D line map are computed (see Figure 3):

5. The nv vertexes in the two-dimension vertex list are searched to compute the pair of vertexes that are located at maximum distance, v_A and v_Z. All the remaining vertexes must be comprehended in the line segment between the two ends, v_A and v_Z.
6. The vertex v_A and its the closest vertex, v_k, define a line segment which is comprehended in the map surface and located at $z = h$ (see Figure 3). Thus, the elementary line segment $[v_A, v_k]$ is part of the locus of the intersection between the 3D surface and the horizontal plane located at the scan height and should be included in the map description.
7. Next consider v_k and its nearest vertex – excluding the vertexes already considered -, v_{k+1}. Since v_k ends a segment that belongs to line map, $[v_k, v_{k+1}]$ defines a line segment that does not belong to the line map.
8. Defining v_{k+2} as the closest vertex to v_{k+1} yet to be considered, and repeating the analysis leads to the fact that the elementary line segment $[v_{k+1}, v_{k+2}]$ is part of the map description.
9. The algorithm is iterated to all vertexes in the list. The closest vertex to the most recent vertex considered is searched in the remaining of the vertex list. Each two vertexes define a line segment that switches from being in the map description and outside the map description.

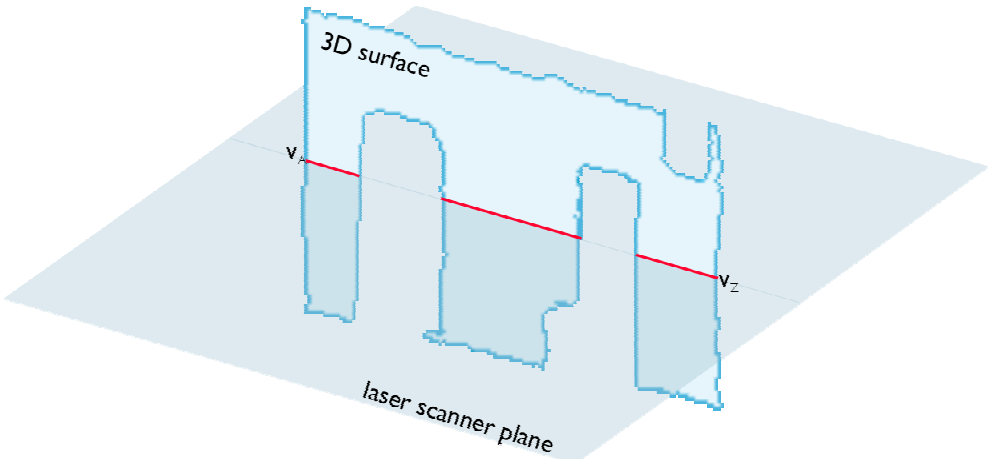


Figure 3 –The 2D map as the intersection between the plane at laser height and the 3D surfaces

The algorithm assumes the surfaces are flat. However, this is an abusive approximation if the surface is a biquadratic. In order to eliminate this hazard, there is a parameter on the surface description to indicate the flatness of the surface.

Notwithstanding, if the surface is not exactly flat the 2D vertexes do not match the line segment between v_A and v_Z, but are located at its vicinity. This effect is considered in the implementation and the elementary line segments are defined at the actual location of the 2D vertexes even though there is some misalignment.

The 3D-reconstruction process [SequeiraV_1] expands the surfaces form the interior to the boundary. Hence, the links between adjacent surfaces are often mismatched, whether the surfaces are co-planar or concurrent. The boundaries of adjacent surfaces should match exactly, i.e., there should be a portion of the contour common to both surfaces. However, in most cases, the two contours are separated by a few centimetres or, alternatively, cross each other by some centimetres too. These errors range from zero to two centimetres in case of a well defined flat surface scanned at short range to a maximum of five centimetres in case the laser scan for 3D reconstruction was taken at a larger distance. In the latter case, the distance between adjacent laser samples can be as large as five centimetres, constraining the reconstruction process.

Thus, it is convenient to adjust the line segments in order to *restore* the right environment description and *correct* these minor reconstruction inaccuracies. This is performed in two ways:

1. Reduce the lines which cross each other to their common point (Figure 4a). This is usually a safe operation since the laser can not scan beyond the foreground surface planes (unless the surface is partially transparent). To reduce the line segments, the intersection point is computed. The segments are redefined, replacing the vertexes that are beyond the intersection by the intersection coordinates, while preserving the other vertex.
2. Extend the lines that are close but do not intersect until a common vertex located at the intersection of their directions (Figure 4b). This extrapolation is subject to errors because there may be surfaces that do not have a common edge although they are close to each other. To extend the line segments, the point defined by the intersection of the two lines is computed. The segments are redefined, replacing the vertexes that are closest to the intersection by the intersection coordinates, while preserving the other vertex.

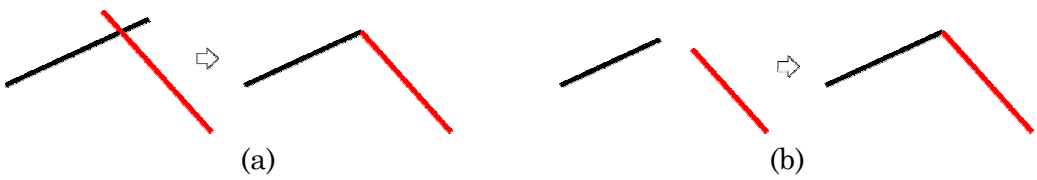


Figure 4 – Matching adjacent line segments

In Figure 4, the corrections were exaggerated for the benefit of illustration. This correction is bounded to a maximum of three to five centimetres, according to a parameter

that may be tuned by the operator.

This operation is an exception to the rule of making no assumptions about the environment topology. Some generality is lost in the way since there is no topological description indicating that any given surface is tangent to any other surface. Nevertheless, the advantages perceived compensate the possible errors, which seldom occurred in the field trials.

The Localisation Map is a list of line segments defined in the XY plane, which is only accurate if the laser height (z-coordinate) is correct. The line segments are defined in the world coordinates. For Localisation convenience, the line list is sorted by decreasing size.

B.3 Extraction of map lines from offline maps

The generation of a Localisation Map in the proposed format is very simple if the map is produced before the mission.

If the room is measured by a human operator, by using a tape or other measurement device, the vertex coordinates are entered manually after some basic calculations. However, the operator should pay attention to the measuring height. While the sensor head is fixed in some acquisition prototypes (the EST), it may be raised and lowered in others (the AEST), as described in Chapter 1, Section 3. Hence, it is important to take the room dimensions approximately at the scanning height and not on the ground level, for instance. Although Localisation copes effectively with some differences in the perceived environment and the proposed map, the map errors should be minimised since the map is assumed to be the *true representation of the scene*, and Localisation must adhere to it.

If the room plan is available from an architectural plan, the scene features must be transferred from the plan media to the proposed format. If the plan is available on paper, a digitizer table can be used to accurately generate the coordinates of the vertexes. The translation to the real dimensions is a simple task that can be performed by any simple digitizer software. If the data is already available in some type of digital format produced by a CAD program, there are usually some filters to extract numeric data from it.

The final task is to compute the number of vertexes on each contour, and the number of contours to fill in the header of the Localisation Map file.

The architectural plans are often a poor estimate of the scene features. First, they sometimes lack detailed vertical information, and only the projections of the surfaces at ground level are available. Second, the scene to be reconstructed is filled with objects and furniture, which are not usually mentioned in the plan.

For instance, the experiment described in Chapter 2, Section 5, was based on an

architectural plan, depicting the contour of the room and the fifth contour in Table 1, near the origin (see Figure 1). The remaining three contours are three file cabinets located close to the top wall in Figure 1, which were measured and appended manually to the file after the floor plan had been digitised.

Regardless of the map generation method, it is clear that some of the error endorsed to the localisation algorithms is in fact due to an incomplete or inadequate Localisation Map.

Appendix C

Line extraction from Laser data

The line extraction procedure is decomposed in three phases: scan segmentation, elementary line extraction and line merge. The three phases will be described in detail including the practical issues, which arose during the development phase, and constrained the proposed solutions.

C.1 Laser scan segmentation

The goal of scan segmentation is to alleviate the line extraction procedure. It is possible to identify some clear surface changes with a simple analysis. Hence, the whole scan may be cut into scan segments and the line extraction is more efficient if applied to a reduced number of surfaces, especially if the segment cuts coincide with the surface changes. In Figure 1, a sample laser scan is shown with some candidate cut points marked. Based on range and reflectance analysis, there are two criterions for the segmentation of the whole laser scan.

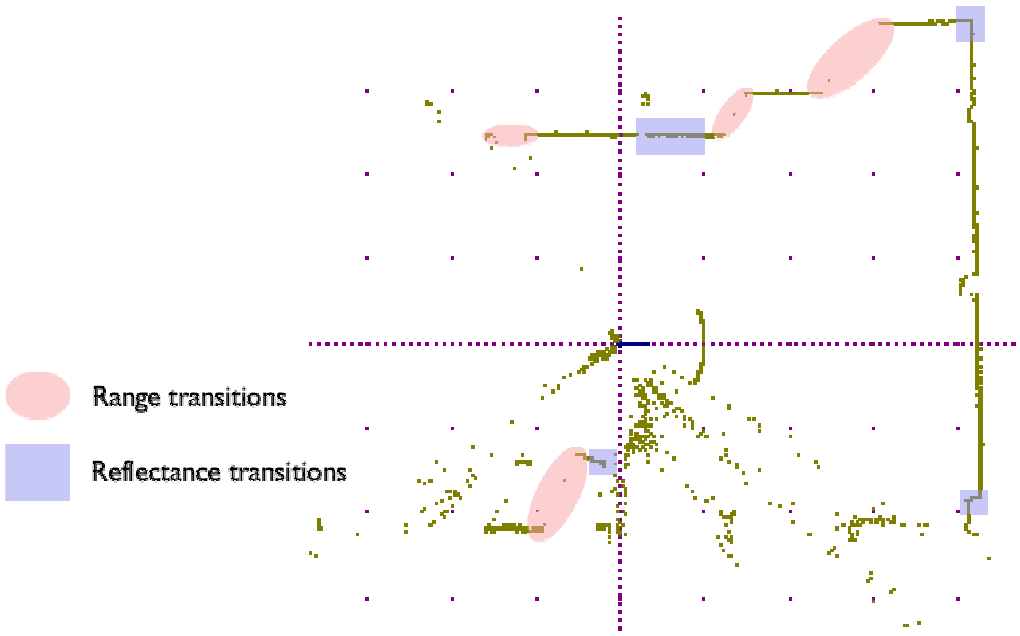


Figure 1 – Segmentation of a laser range scan

C.1.1 Range data analysis

The range information is used to cut the whole scan into contiguous scan segments, *i.e.*, the segment is a sequence of adjacent samples with a smooth distance variation along it. A sudden distance variation between adjacent samples usually indicates a surface transition when one element of the scene is occluding another element. Exceptions occur when the laser beam is almost parallel to the scan surface and adjacent surface hits are too distant. It may also occur that a surface transition is masqueraded by the mixed footprint effect described in Chapter 2, Section 5.2.

The minimum range difference between consecutive samples in a common segment is defined by a maximum range threshold. This parameter is available to the human operator for tuning according to the operating conditions. Four of these transitions are shown in Figure 1.

A second aspect of the range analysis is the detection of invalid measurements, which are often marked with a flag value (for instance, -1.0). If a single or a very small number of invalid values occur within a segment, the invalid values are discarded and the remaining points are treated as a common segment. On the other hand, if there is a group of invalid points inside a segment, the invalid points are eliminated and the segment is partitioned in two segments.

In the current implementation, the minimum size of a segment is four samples. The segment is partitioned in two if two consecutive samples are invalid or four non-adjacent invalid samples are found within 10 consecutive samples. These parameters may be tuned by the operator to suit different environment conditions and laser equipment. However, it should be emphasised that relaxing the parameters to accommodate more invalid measurements degrades the Localisation performance severely. The laser scanner suffices to the Localisation solution because of its accuracy. Otherwise, more sensors would be required.

C.1.2 Reflectance data analysis

Although the range data is the most important source of information, the analysis of the reflectance data proved to be valuable in the statistical analysis of the laser data. Originally, reflectance was added to overcome the difficulties created by partially mirrored windows, which are present in all rooms used for development trials.

It is common in modern office and residential buildings to face large extensions of a wall, if not an entire wall, made of reflecting glass. When the laser beam hits such surfaces, a fraction of the energy passes through but a sufficient amount of light is reflected, according to the reflection laws, and proceeds to the next surface (Figure 2).

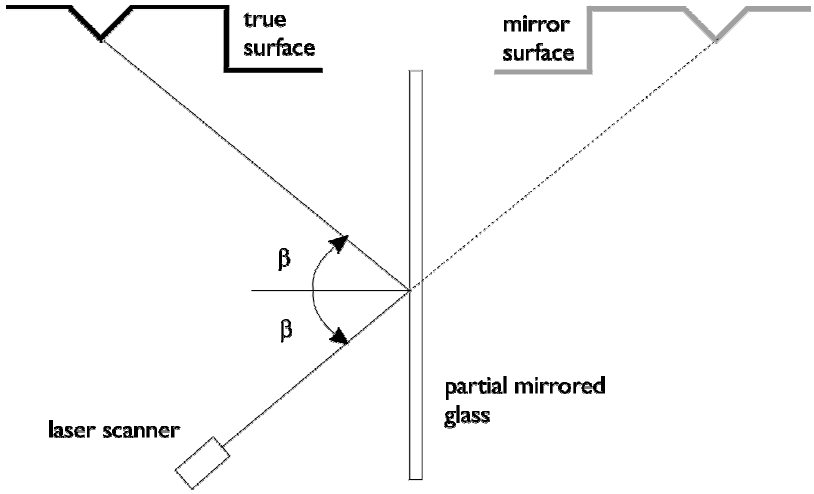


Figure 2 - the effect of partial mirrors

The return path is similar and, depending on the window reflectance degree, the angle of incidence, and to a lesser extent, the thickness of the material, a "visible echo" is received at the laser range finder. In such cases, the image beyond the glass is a mirrored version of the real room. The reflectance data allows the detection of the phenomenon since the data in the mirrored surfaces has a lower reflectance value than the surfaces in the direct path.

In addition, dark or textured surfaces absorb a large fraction of the incident laser beam and the intensity of the returned signal is comparable to the reflection by a partial mirror. Moreover, in the presence of full-reflective mirrors the reflectance test fails since nearly all the energy follows the reflection path and the glass only absorbs a minimal fraction of it. To overcome these difficulties a refined analysis is required, based on the reflectance variation between neighbouring samples.

Given these constraints, the optimisation of the reflectance thresholds described in Chapter 2, Section 5, recommends some careful experiments on field prior to any reconstruction trials.

If the reflectance value is above the minimum threshold, the analysis may be extended to the reflectance variation between adjacent data samples. A strong and apparently random variation suggests a metallic surface, a highly textured feature or a partially mirrored window. In all cases, the line extraction will perform poorly. In presence of these features, the segmentation procedure will separate the ill-conditioned data from the remaining samples, creating new segments.

Beyond these difficulties, the reflectance data is used to detect surface changes, especially corners. Given the fact that reflectance depends on the angle of incidence, as described in Chapter 2, Section 2.5, when the sweeping laser changes the target surface there is a sudden change of incidence angle. In Figure 3 two consecutive laser hits are illustrated (the angular scale is much exaggerated and the linear scale is much reduced for clarity sake).

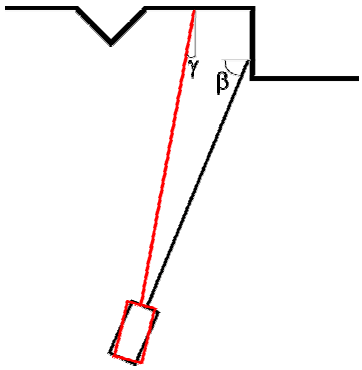


Figure 3 – Corner detection

It is apparent that angle β is different from angle γ. In fact, if the two adjacent surfaces are close to orthogonal, the two angles should be close to complementary, *i.e.*,

$$\gamma \approx \frac{\pi}{2} - \beta \qquad \text{(C.1)}$$

If the angle of incidence is different of $\pi/4$ (45º), there is a distinct and consistent change in the reflectance at the surface intersection. Thus, the original scan segment is divided in two segments at the transition point.

However, many factors may contribute to induce false surface transitions, whereas other factors contribute to conceal some true transitions, although less often. The most common source of false transitions is a change in colour or texture of a given surface. This may occur in painted walls, with objects on the walls, such as posters or paintings, but also by the change of the materials the surface is made of. The electrical wiring fixed to the corners of the walls and the decoration features used to "round up" the corners in some architectural styles are the most common reasons for concealed surface transitions.

The set of criterions, based on range and reflectance analysis, used to divide the original laser scan into a list of segments generates a large number of unintended cuts. However, this procedure is only an "educated guess" that helps the line extraction procedure. If the segmentation algorithm can handle the association of the data samples to the segments without losing information, the final line merge will be capable of reconstructing the whole scene features, based on the elementary lines computed from each segment.

C.1.3 Segmentation algorithm

This sub-section provides a formal presentation of the segmentation algorithm, based on the aforementioned considerations on range and reflectance.

The laser scan is a set of **n** points, equally spaced in the radial co-ordinate in a two dimensional polar space. Each data point, p_i, is a triplet containing the range, r_i, the reflectance (or brightness), b_i, and the pan angle, θ_i. Due to the hardware characteristics, in some configurations it is not possible to measure the angle directly. Instead, there is a flag to mark the beginning and the completion of a revolution, and an encoder provides an approximate measure of the angle.

The list of points, P, is defined as:

$$P = \{p_i = (\theta_i, r_i, b_i),\ 1 \le i \le n\} \tag{C.2}$$

The scan points must comply with condition (C.3) to be validated, where $Rng_{\min}$ is the minimum valid range (positive value, close to 0.0, *e.g.*, 0.1m) and $Rng_{\max}$ is the maximum range value which is hardware dependent (*e.g.*, 20.0m). These two parameters may be tuned by the operator to accommodate different laser devices or operating conditions.

$$Rng_{\min} \le r_i \le Rng_{\max} \tag{C.3}$$

Each segment, S_m, is a set of np points, defined in cartesian coordinates. Given the whole laser scan, the segment list is generated according to the following procedure:

1. Start with p_1. Let $i = 1$.
2. If p_i is invalid – condition (C.3) fails – proceed with the following points, p_{i+1}, p_{i+2}, … until a valid point is found. The first valid point, p_i, starts the next segment S_m.
3. Proceed to the next point $p_j = p_{i+1}$.
4. If p_j is invalid while the previous points $p_k, i \le k < j$ are valid, the segment is discarded in case condition (C.4) fails. This means that segments with fewer points than a user defined threshold, $NP_{\min}$, are ignored. If (C.4) holds, the segment S_m is defined with $np_m = i - j$ points. Each point p_k is translated to p_k^R, defined in the cartesian frame of reference associated to the robot origin (C.5). Proceed to step 2.

$$j - i \ge NP_{\min} \tag{C.4}$$

$$p_k^R = \begin{bmatrix} x_k \\ y_k \\ b_k \end{bmatrix} = \begin{bmatrix} r_{ki} \cdot \cos(\theta_k) \\ r_k \cdot \sin(\theta_k) \\ b_k \end{bmatrix} \tag{C.5}$$

5. If p_j is valid, the range difference between adjacent points is tested. If the condition (C.6) holds, the algorithm proceeds to the next step. Otherwise, S_m is defined with $np_m = i - j - 1$ points, encompassing the points from p_i to p_{j-1}, translated to the cartesian reference, according to equation (C.5). Point p_j initiates a new segment, S_{m+1}, as p_i. Proceed to step 3.

$$\left| r_j - r_{j-1} \right| \leq DifRng_{\max} \qquad \text{(C.6)}$$

6. If p_j is valid and (C.6) holds the brightness difference is measured between adjacent samples. If the condition (C.7) holds, the algorithm proceeds to the next step. Otherwise, S_m is defined with $np_m = i - j - 1$ points, encompassing the points from p_i to p_{j-1}, also defined in cartesian coordinates. Point p_j initiates a new segment, S_{m+1}, as p_i. Proceed to step 3.

$$\left| b_j - b_{j-1} \right| \leq DifBgt_{\max} \qquad \text{(C.7)}$$

7. Given that p_j is valid and the conditions (C.4), (C.6) and (C.7) are met, advance to the next point, p_{j+1} and return to step 4.

The segmentation procedure keeps the angular sequence of the samples, *i.e.*, the segment sequence follows the angular distribution of the samples, regardless of the range distribution. This will prove valuable when merging lines from adjacent segments.

The raw laser data measures the distance between some fixed point in the laser sensor and the target. With some simple calculations, the raw laser data is referred to a circular reference associated to the laser device. This transform depends on the laser sensor geometry (see Chapter 1, Section 3).

To reduce the computation effort, all the subsequent line extraction operations are referred to this system of coordinates. Only after the line merge procedure, the lines are translated from the reference associated to the laser beam rotation into the robot reference. The translation involves a cascade of coordinate transform, whose parameters are stored in a common blackboard available to all software modules. In case there are moving part (*e.g.*, the elevator shelf in the AEST) the transforms are time variant, otherwise they are based on a set of physical parameters measured by the operator.

Once the whole laser scan is cut into segments, the elementary line extraction algorithm handles each independent segment in sequence.

C.2 Elementary line extraction

C.2.1 Deduction of the line extraction *formulae*

The line extraction procedure is solved in the framework of likelihood estimation. An unknown process $y(x)$ is sampled, generating a set of n points (x, y). The process will be approximated by a known function $y(x_i; p_1, p_2, ..., p_m)$ where $p_1, p_2, ..., p_m$ are m unknown fixed parameters. If the samples are independent and the x and y errors follow a Gaussian distribution with zero mean, the Least Square Estimation (LSE) will provide the maximum likelihood estimator.

The generic LSE equation may be written as in (C.8), in the form of a χ^2 distribution, where x_i and y_i are the individual samples and σ_i^2 is the variance associated to each sample.

$$\min(p_1, p_2, ..., p_m) : \sum_{i=1}^{n} \left[\frac{y_i - y(x_i; p_1, p_2, ..., p_m)}{\sigma_i^2} \right]^2 \tag{C.8}$$

Assuming the data samples fit a linear function, the general model function in (C.8) is defined as $y(x, B, M) = B + M \cdot x$. Under the constraints defined in Chapter 2, Section 5, there isn't a model of the variance of the data. To obviate to the problem, it is assumed the variance is equal for all samples. The χ^2 expression (C.8) is instantiated as (C.9).

$$\min(B, M) : \frac{1}{\sigma^2} \sum_{i=1}^{n} \left[y_i - (B + M \cdot x_i) \right]^2 \tag{C.9}$$

The quadratic error relates to χ^2 by a factor of σ^2. The quadratic error, E^2, is expanded as

$$E^2 = \sum_{i=1}^{n} \left[Mx_i^2 + y_i^2 + 2BMx_i - 2By_i - 2Mx_i y_i + B^2 \right] \tag{C.10}$$

To determine the values of B and M, the partial derivatives of E^2 are computed. Simultaneously, the sums are transferred to the sampled variables, acting like fixed parameters.

$$\frac{\partial E^2}{\partial B} = 2M \sum_{i=1}^{n} x_i - 2 \sum_{i=1}^{n} y_i + 2Bn \tag{C.11}$$

$$\frac{\partial E^2}{\partial M} = 2M\sum_{i=1}^{n} x_i^2 - 2B\sum_{i=1}^{n} x_i - \sum_{i=1}^{n} x_i y_i \tag{C.12}$$

Equalling (C.11) and (C.12) to zero simultaneously generates a system of two linear equations on two unknowns, (C.13), which may be readily solved, (C.14).

$$\begin{bmatrix} \sum_i^n x_i & n \\ \sum_i^n x_i^2 & \sum_i^n x_i \end{bmatrix} \cdot \begin{bmatrix} B \\ M \end{bmatrix} = \begin{bmatrix} \sum_i^n y_i \\ \sum_i^n x_i y_i \end{bmatrix} \tag{C.13}$$

$$\begin{cases} B = \dfrac{\sum_{i=1}^n x_i^2 \sum_{i=1}^n y_i - \sum_{i=1}^n x_i \sum_{i=1}^n x_i y_i}{n\sum_{i=1}^n x_i^2 - \sum_{i=1}^n x_i \sum_{i=1}^n x_i} \\ M = \dfrac{n\sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n\sum_{i=1}^n x_i^2 - \sum_{i=1}^n x_i \sum_{i=1}^n x_i} \end{cases} \tag{C.14}$$

This formulation is optimal in the sense that the iterative implementation of the algorithm requires only the storage of six parameters, which are in turn, computed iteratively: $\sum x$, $\sum x^2$, $\sum y$, $\sum y^2$, $\sum xy$, n, (C.15). If one instance of the problem is to be extended with a new set of samples, it is sufficient to add the two sets of sums and the new line parameters are readily computed from (C.14). Conversely, if one whole set of samples is to be divided in two separate instances, it is sufficient to compute the sums in (C.15) for the shorter one, and subtract them from the original sums to get the new line extraction parameters.

$$\begin{aligned} \sum x &= \sum_{i=1}^n x_i & \sum x^2 &= \sum_{i=1}^n x_i^2 \\ \sum y &= \sum_{i=1}^n y_i & \sum y^2 &= \sum_{i=1}^n y_i^2 \\ \sum xy &= \sum_{i=1}^n x_i \cdot y_i & n &= \sum_{i=1}^n 1 \end{aligned} \tag{C.15}$$

The quadratic error, which measures the overall deviation from the model $y(x, B, M) = B + M \cdot x$, is expressed in equation (C.10), is also re-written in the form of the sums:

$$E^2 = M^2 \sum x^2 + \sum y^2 + 2BM \sum x - 2B \sum y - 2M \sum xy + B^2 n \qquad \text{(C.16)}$$

The proposed line extraction method accommodates a shift in the coordinate system in order to optimise the accuracy of the line parameters. If the data is concentrated in a narrow range in the x-axis, the parameter definition will be ill conditioned, due to the low value of the denominator, (C.17). Therefore, a shift in the problem variables would minimise the extraction error, unless the data is also constrained to a narrow stripe in the y-axis, suggesting an ill-conditioned problem.

The computation of the alternative line $x(y, B', M') = B' + M' \cdot y$ is immediate from the six sums in (C.15), requiring only the additional computation of the parameters, similar to (C.14), exchanging the variable's names, (C.18).

$$n \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i \qquad \text{(C.17)}$$

$$\begin{cases} B' = \dfrac{\sum_{i=1}^{n} y_i^2 \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i y_i}{n \sum_{i=1}^{n} y_i^2 - \sum_{i=1}^{n} y_i \sum_{i=1}^{n} y_i} \\ M' = \dfrac{n \sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i}{n \sum_{i=1}^{n} y_i^2 - \sum_{i=1}^{n} y_i \sum_{i=1}^{n} y_i} \end{cases} \qquad \text{(C.18)}$$

The quadratic error is redefined likewise, (C.19).

$$E^2 = (M')^2 \sum y^2 + \sum x^2 + 2B'M' \sum y - 2B' \sum x - 2M' \sum xy + (B')^2 n \qquad \text{(C.19)}$$

The symmetry with respect to $\boldsymbol{x}$ and $\boldsymbol{y}$ is an essential feature of the current implementation given the strict accuracy requirements of the localisation problem. The switch between the two modes, $y(x, B, M)$ and $x(y, B', M')$, depends on the slope, M (or M'). The switch was designed with a histeresis gap, as explained in Chapter 2, Section 5.

C.2.2 Line extraction algorithm

The line extraction algorithm starts with the laser scan data divided in segments and already translated to cartesian coordinates, expressed in the laser circular reference.

The algorithm is implemented in a recursive manner for each segment. The first iteration considers the whole segment as a single line. In case it succeeds, the algorithm

moves to the next segment. Otherwise, the segment is cut in two pieces that initiate the second iteration of the algorithm now in a recursive manner.

Finally, all subsets of the original segment are transformed in lines or, except the subsets that have a much reduced number of points, which are discarded.

For each segment of laser data, the line extraction follows the steps below:

1. Consider the set of laser scan data as a single line. Compute the (C.15) sums.
2. Compute the line parameters B and M – equations (C.14) – and the quadratic error - equation (C.16). If the average quadratic error E^2/n is below a specified threshold, the set is considered a line and the algorithm continues in step 6.
3. Otherwise, the set of data is divided in two pieces by a moving pointer (see Figure 4). The pointer begins at position q, where q is the minimum number of samples to generate a line. The (C.15) sums are computed only for the first subset, while the second subset sums are obtained by subtracting the newly computed sums from the previous total sums.
4. The pointer sweeps the set, sample by sample computing the B and M parameters and the average quadratic error E^2/n separately at the left and right of the pointer. To optimise the computation effort, the sample pointed by p is subtracted from the first set sums and it is added to the second set sums. The location, $p_{\min}$, and the minimum value of the average quadratic error are stored as the pointer proceeds along the data set until position $n-q$.
5. Split the data set in two subsets. The first subset encompasses data from sample 1 to sample $p_{\min}$ and the second subset is defined from $p_{\min}$ to n. Restart the algorithm at step 1 with the two subsets.
6. Compute the ends of the extracted line segments. The end points of the line segments are computed from the projection of the end samples in the data set on the line of parameters B and M. The point in a line (x_e, y_e) nearest to point (x_p, y_p) is determined by equations (C.21). The line segment is stored in an extracted line list, represented by its vertexes.
7. Consider the next subset of data in the same segment. If there is no more data in the current segment, the algorithm resumes with the next segment.

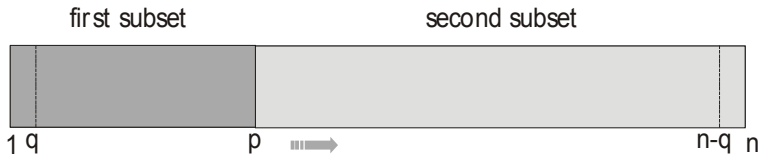


Figure 4 – Sliding pointer along a laser scan set

The vertexes of the extracted line segments are computed from the definition of the distance of a point to a line $y(x) = B + M \cdot x$. Deriving D^2 with respect to x_e results in

(C.21).

$$D^2 = (x_p - x_e)^2 + (y_p - (B + Mx_e))^2 \tag{C.20}$$

$$\begin{cases} x_e = \dfrac{x_p + M(y_p - B)}{1 + M^2} \\ \\ y_e = \dfrac{B + Mx_p + M^2 y_p}{1 + M^2} \end{cases} \tag{C.21}$$

For each extracted line segment an equivalent reflectance is also defined. This parameter associates a confidence value to the line segment. The equivalent reflectance is the average of the reflectance of the individual samples considered in the line segment.

The recursive calls follow a fixed processing rule in order to keep the sequence of line segments. The calling sequence is illustrated in Figure 5. When a data set is split in two, the first subset is immediately processed. The second subset is analysed only after the first subset and its entire tree of subsets is processed.

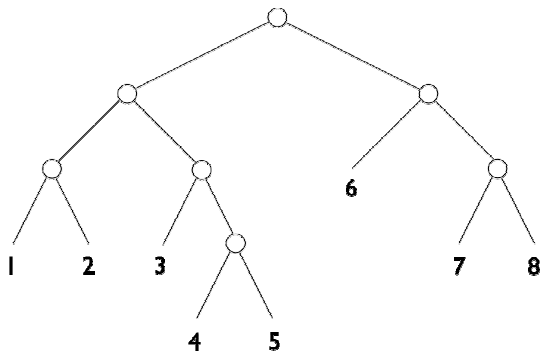
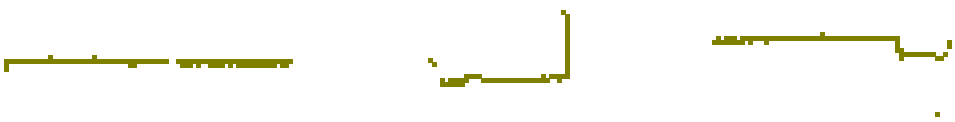


Figure 5 – Recursive call sequence

Sometimes one scan segment contains a single line, as shown in Figure 6a. In such cases the algorithm performs only steps one and two and then proceeds to step 6.



(a) one line (b) two lines (c) three lines

Figure 6 – Examples of data sets with varying number of lines

Most often, though, there is more than one line per segment. This is due to the variations in the environment, which are not followed by threshold adjustments and sometimes to the geometry relations between the surfaces. If the angle of incidence near the corner in Figure 6b is approximately $\pi/4$ (45º), the surface change will be ignored.

The sliding pointer method described in steps 3 and 4 will bear a clear minimum near the corner and the segment will be split in two segments, which are readily solved in the second iteration.

Finally, it may occur that many lines are included in a single segment as in Figure 6c. This event may prevent the optimal detection of the surface transitions, inducing the segment split in sub-optimal points. Under these circumstances, the resulting lines will be over fragmented and the line merge operation will be required to restore the environment structure.

In spite of the relative high quality of the laser data, which makes it best suited for line extraction, errors sometimes occur. If a measurement is valid but too distant from the straight line, it may be removed from the line to reduce the overall quadratic error. This occurs specially at the end of the sets, at the surface transitions. Following this approach will reduce the size of the extracted lines by one or two samples, with benefits on the parameter accuracy. This error may be recovered in the line merge procedure, as it will be shown.

Although the line extraction algorithm is the most resource-consuming module in the localisation solution, it was designed with efficiency concerns at every stage. The efficiency of the line extraction algorithm goes with $\boldsymbol{n}$ in the non-recursive version. Establishing the segmentation as a pre-processing phase is an essential factor for increased efficiency, reducing the number of recursive calls. It is also important for exception handling, since many data errors are removed at an early stage.

In addition, the sums in (C.14) provide a flexible method of manipulating the laser scan samples without losing the previously computed data. It is also quite efficient in the memory management, since only a limited amount of single variables is stored.

C.3 Line Merge

The last stage of the line extraction algorithm is the line merge. The previous stages - segmentation and elementary line extraction - fragment the data into small pieces. Thus, it is convenient to associate the neighbouring short line segments to build longer line segments in order to represent the environment features more accurately and alleviate the computation effort. Since each large feature in the scene map is represented by a single line, the corresponding scan lines should resemble it as close as possible.

It is worth recalling the origins of fragmentation:

- When the data is split in two segments to eliminate a group of invalid samples due to

a change in material or some obstacle which is not acquired, the surface in the background is divided in two.

- Similarly, the data are split in two when there is a sudden reflectance variation or the reflectance falls below the minimum confidence threshold. For instance, a black band (or a cable) on a bright wall will split a wall profile in two segments.
- When the line extraction procedure considers a segment with more than two lines, the minimum quadratic error will seldom occur exactly at a surface transition. In these cases, the subsets resulting from the set division will break a single line in two.
- Finally, when an object is occluding a larger feature in the background the two parts of the background feature will be considered different surfaces, even though they are aligned. However, joining the two background surfaces is a dare step since there is no data about the surface description behind the foreground object. Therefore, this step will not be taken and the line segments will be kept apart.

The line merge procedure begins with a list of lines, sorted by the order of appearance in the laser data sequence. This is important to minimise the computational effort, since line l_i can only be adjacent to lines l_{i-1} and l_{i+1}. This solution proved to be robust and swift.

Given the lines $l_i = \left[(x_i^a, y_i^a) \rightarrow (x_i^b, y_i^b)\right]$ and $l_{i+1} = \left[(x_{i+1}^a, y_{i+1}^a) \rightarrow (x_{i+1}^b, y_{i+1}^b)\right]$ which are adjacent according to the scan sequence, the neighbouring vertexes are (x_i^b, y_i^b) and (x_{i+1}^a, y_{i+1}^a).

The rules for line merging are the following:

1. Two adjacent line segments may be joined if the distance between the vertexes is smaller than a pre-defined threshold, $D_{\min}$, (C.22).

$$(x_i^b - x_{i+1}^b)^2 + (y_i^b - y_{i+1}^a)^2 < D_{\min}{}^2 \qquad \text{(C.22)}$$

2. If the previous condition holds, the inner angle between two lines, θ, measured at the common vertex (xc_i^{i+1}, yc_i^{i+1}) – defined as the middle point of the two neighbouring vertexes, (C.23) – should be close to π (plus or minus a specified threshold), see Figure 7 . The inner angle is computed from the ratio of the inner product with the product of the line segment norms, (C.24). Equivalently, $\cos(\theta)$ is computed instead, to avoid the time consuming operation.

$$\begin{cases} xc_i^{i+1} = \dfrac{x_i^b + x_{i+1}^a}{2} \\ yc_i^{i+1} = \dfrac{y_i^b + y_{i+1}^a}{2} \end{cases} \qquad \text{(C.23)}$$

$$\cos(\theta) = \frac{(x_i^a - xc_i^{i+1}) \cdot (x_{i+1}^b - xc_i^{i+1}) + (y_i^a - yc_i^{i+1}) \cdot (y_{i+1}^b - yc_i^{i+1})}{\sqrt{(x_i^a - xc_i^{i+1})^2 + (y_i^a - yc_i^{i+1})^2} \cdot \sqrt{(x_{i+1}^b - xc_i^{i+1})^2 + (y_{i+1}^b - yc_i^{i+1})^2}} \qquad \text{(C.24)}$$

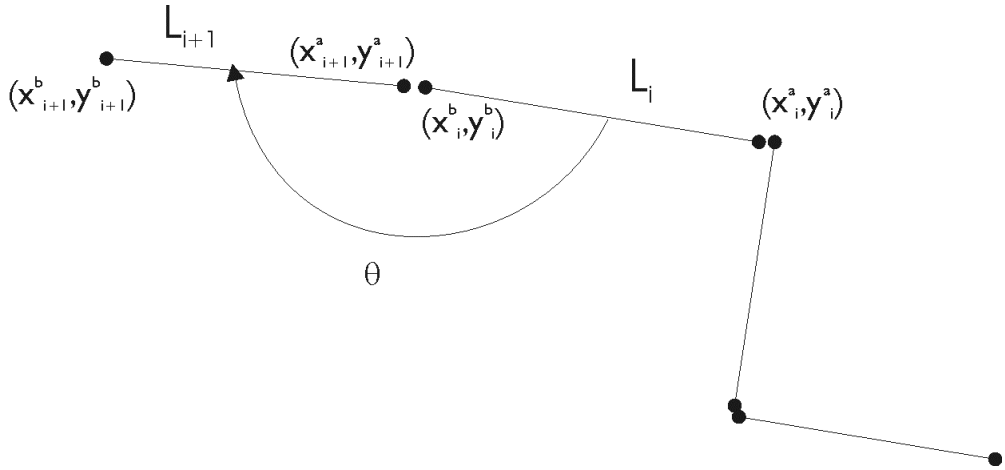


Figure 7 – The merge of two adjacent, nearly aligned line segments

3. If the line segments are close and aligned, they are replaced by a new line segment defined by the end vertexes of the considered pair $l_i' = \left[(x_i^a, y_i^a) \rightarrow (x_{i+1}^b, y_{i+1}^b)\right]$. The new equivalent reflectance is defined as the weighed average of the reflectance of l_{i-1} and l_{i+1}, using the size of the segments as the weighing criterion. Line l_{i+1} is removed from the line list.
4. The loop repeats until the end of the line list.

The final step of the line merge procedure is to remove the lines that are too short to be used for localisation. The minimum line length suitable for localisation is a user-defined parameter. Typical values range from 0.02m to 0.1m.

The outcome of the three-stage Line Extraction algorithm is a list of line segments. The sequence of line segments follows the order of the samples in the laser scan used to build the lines.

The Line Extraction algorithm depends largely on the fine tuning of the multiple thresholds used. There are thresholds to segment the laser scan, to assign confidence levels to the laser samples, to determine the linear characteristics of data sets, and so forth. Nearly all of these parameters are available to the user for tuning. Notwithstanding, tuning so many and often so interdependent parameters requires a deep understanding of the underlying algorithms.

In order to assist the user to understand the inner procedures, a set of log files was created. The main log file records the results of all major operations and the variable outcome. The log file is illustrated by a set of log images, one image per major stage.

There is an image with the whole scan including reflectance data, an image with the segments, an image with the elementary lines and finally an image with the merged lines.

Using these tools, it is expectable that a user can acquire the insights and sensitivity to the issues involved in accurate localisation, which is regarded as a requirement to optimise the process to the vast universe of applications and operation conditions the localisation algorithms are designed for.

Appendix D

Optimal step-by-step clustering

The clustering problem to be solved is stated as follows: "given a cloud of N weighed points defined in the form of (D.1), compute a clustering solution which aggregates all points lying closer than D from their closest neighbour". In addition, it is known that the distance between neighbouring clusters is much higher than the distance between elements of the same cluster.

$$p_i = \begin{bmatrix} x_i \\ y_i \\ \theta_i \\ w_i \end{bmatrix}, \quad i = 1,2,...,N \qquad \text{(D.1)}$$

In order to compute the distance between two points, a specific metrics must be defined to embrace the linear and angular distance in one single measurement. Since this distance is associated to an error, an acceptable criterion is that the weight ratio between the position and orientation variables follows the ratio between the position and orientation covariances. However, given the heterogeneous nature and diversity of the error sources, addressed in Chapter 2, Section 5, a model of covariance errors was not defined.

An alternative way to define the metrics is to weigh the relevance of the angular error. If a candidate posture has an angular error $\delta\theta$ and assuming the position errors are neglected, the system posture, based on the scene elements considered, will be deviated from the correct location by $r(\theta)\cdot\delta\theta$. The distance to the scene surfaces in each direction, $r(\theta)$, depends on the scene characteristics. For typical operation ranges of 1 to 10 meters and small angular errors (less than 0.1rad (≈6º)), the error ratio between angular error and linear error, R, varies between 1 to 10. In order to adapt this parameter to different scenarios, the human operator should be allowed to tune it.

The distance between two points p_i and p_j is defined in equation (D.2), where R is the angular to linear error ratio.

$$\Delta_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + R^2 \cdot (\theta_i - \theta_j)^2} \qquad \text{(D.2)}$$

After the metrics is established, the clustering algorithm may be defined. The proposed solution is optimal in the sense that at each step, only the two closest points are merged. The algorithm works as follows:

1. Create a table with the point to point distance between all points. This table may be created as a matrix. Given the distance definition, it may be reduced to an upper triangular matrix because $\Delta_{ij} = \Delta_{ji}, \quad \forall i, j \in \{1,...,n\}$ and $\Delta_{ii} = 0, \quad \forall i$.

$$T = \begin{bmatrix} 0 & \Delta_{1,2} & \Delta_{1,3} & \cdots & \Delta_{1,N} \\ 0 & 0 & \Delta_{2,3} & \cdots & \Delta_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \Delta_{N-1,N} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{D.3}$$

2. Sweep the table to detect the lowest distance, $\Delta_{\min}$.
3. If $\Delta_{\min} \leq D$, merge the two points according to their relative weights. Assuming the point with higher weight is kept, for instance p_j, the posture p_j is updated according to equations (D.4) and (D.5) and the entries relative to p_i are removed.
   Depending on the implementation this may be achieved by removing one line and one column from the table or, alternatively by setting the entries to zero.

$$\begin{cases} x_j = \dfrac{w_j.x_j + w_i.x_i}{w_j + w_i} \\ y_j = \dfrac{w_j.y_j + w_i.y_i}{w_j + w_i} \\ \theta_j = \dfrac{w_j.\theta_j + w_i.\theta_i}{w_j + w_i} \end{cases} \tag{D.4}$$

$$w_j = w_j + w_i \tag{D.5}$$

4. The distance table must then be updated, to reflect the changes in p_j. Since only one line and one column are affected, the required number of mathematical operations is reduced.
5. The algorithm is resumed to step 2, until $\Delta_{\min} > D$.
6. After the clustering cycle, the points in the cloud are sorted by decreasing weight.

The algorithm proposed in Chapter 2, Section 4, is a simplified version of this optimal step-by-step algorithm. The most important difference is the separation between the angular and linear errors in the simplified version, which might eliminate some points with unbalanced errors. A second difference arises from the creation of clusters, which assume the distances between clusters are larger than the distances between points and clusters. If a point belongs to a given cluster, there is no need to check if it might belong to a different cluster. Finally, it is assumed the distances between neighbouring points are closer than the cluster radius. Hence, it is not necessary to check if the cluster displacement after a posture update brings another point into the cluster radius.

These approximations, although crude, proved to be acceptable under the current operation conditions and they reduce the number of calculations largely. Nevertheless, the computation requirements of the full version are acceptable, should the simplified version prove to be inaccurate.

Appendix E

Natural landmark selection

One of the initial steps of Reference Transform introduced in Chapter 4, Section 1, is the detection of natural landmarks that will establish a correspondence between the simulated scan and the laser scan.

Although the number of natural landmarks detected in the simulated scan is usually small (less than six), it might be high (more than ten) in case the scene map is almost complete or the scene description is very rich (high number of corners, many pieces of furniture in the room, etc.).

In these circumstances, it is convenient to select the most appropriate landmarks among the numerous candidates detected by the normalised inner product procedure. The classroom introduced in Chapter 3, Section 4 and later used also in Chapter 4 might be used as an example to the method.

The simulated scan shown in Figure 1 was based in the off-line map from the classroom. The limits of the panning angle are defined to be equal to the laser scan. Assuming the panning angle was 2π and the robot was located close to the left wall (west wall, see Chapter 3, Section 4), the direction of the detected landmarks is shown.

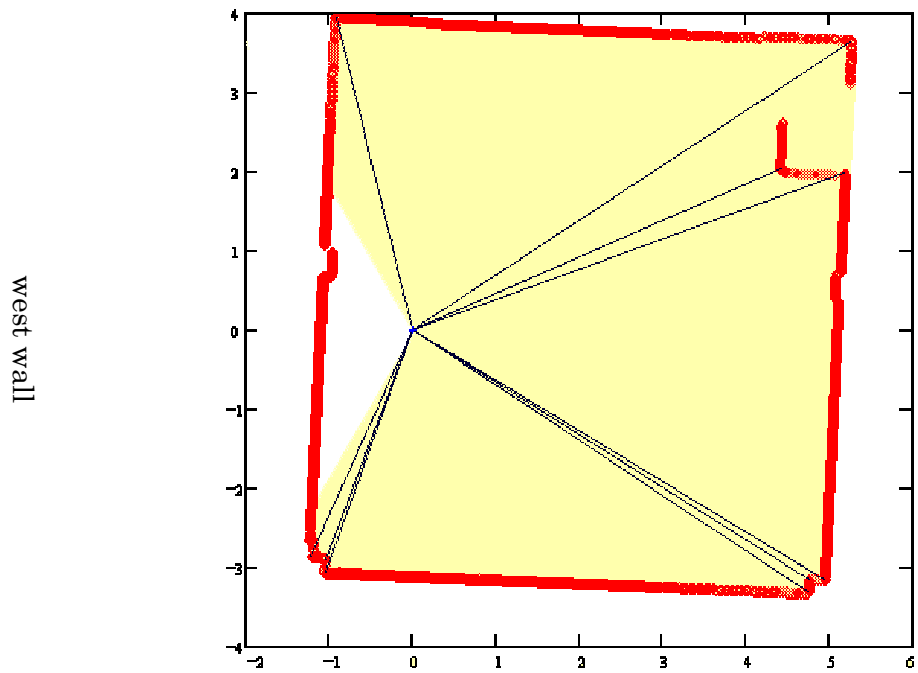


Figure 1 – Natural landmarks detected in a simulated scan

The coordinates in the axes refer to the coordinate system attached to the robot.

The actual field of view of the laser scanner is approximately 4.4rad (252º), which corresponds to the shaded area in Figure 1. Although there are no landmarks outside the field of view, the simulated scan is also computed in these directions in case the laser includes them.

The pan movement starts at the north wall (top) and sweeps the scene in counter-clockwise sense until the same direction in the north wall is reached again. The first landmark is detected on the top left corner, followed by a cluster of three landmarks in the lower left corner, another cluster in the lower right corner, two landmarks in the L-shape and one landmark in the top right corner. The total number of natural landmarks detected is 10. It should be noticed that the landmark sequence is determined by the sweeping movement.

Depending on the scene topology and architecture, the natural landmarks may be grouped in clusters, where a cluster is a set of landmarks located at short distances. In Figure 1, there are two of these clusters, in the lower left and in the lower right corners.

The position and orientation errors in the posture estimate are translated into the point-to-point differences between the simulated scan and the laser scan along the profiles. It is expected that the point-to-point distance vector (E.1) relating the simulated scan landmarks and the laser scan landmarks within a given cluster should be similar, since the landmarks are close to each other. On the contrary, landmarks located on opposite orientations should contribute with different distance vectors.

$$\begin{bmatrix} x_i^{robot} - x_i^{SimScan} \\ y_i^{robot} - y_i^{SimScan} \end{bmatrix} \tag{E.1}$$

Therefore, in case there are too many landmarks with correspondence in the simulated scan and in the laser scan (more than ten landmarks), a subset of the candidates should be chosen. The method chosen to select the candidates was kept very simple, because the occurrence of too many landmarks is quit rare. The method is based only on the angular distance between landmarks and it handles the landmarks regardless of their aggregation in clusters.

The natural landmark selection algorithm is applied only to the pairs with corresponding simulated scan and laser scan landmarks. However, the selection algorithm is applied only to one of the profiles, *e.g.* the simulated scan; the selected elements in the second profile are determined by the landmark correspondence.

The sequence of operation is described below, where the set of *NL* candidate landmarks is ordered by increasing pan angle:

1. The first and last candidate landmarks are always included in the selected landmark set, since they maximise the angular difference between landmarks.

2. The indexes of the first (s_0) and last landmarks (s_1) relative to the simulated scan sequence are recorded. The centre of the interval between the two corner landmarks is computed, (E.2), where the ≈ sign means the closest landmark to the sample defined by the average sum of s_0 and s_1.
3. Two additional landmarks are computed close to half the distance between the three available landmarks. $s_{1/4}$ is defined as (E.3) and $s_{3/4}$ is defined as (E.4). Depending on the landmark distribution one of these landmarks may not exist, in case the candidate landmarks are all below or above $s_{1/2}$.
4. If $s_{1/4}$ exists, the new landmark ($s_{1/8}$) is defined as (E.5). The next landmark ($s_{3/8}$) is defined as (E.6) . If s_{134} exists, add a new landmark ($s_{5/8}$) at (E.7). The final landmark ($s_{7/8}$) is defined as (E.8).
5. In any case, the candidate landmark must lay at least 20 samples away from any landmark already selected.

$$s_{1/2} \approx \frac{s_0 + s_1}{2} \tag{E.2}$$

$$s_{1/4} \approx \frac{s_0 + s_{1/2}}{2} \tag{E.3}$$

$$s_{3/4} \approx \frac{s_{1/2} + s_1}{2} \tag{E.4}$$

$$s_{1/8} \approx \frac{s_0 + s_{1/4}}{2} \tag{E.5}$$

$$s_{3/8} \approx \frac{s_{1/4} + s_{1/2}}{2} \tag{E.6}$$

$$s_{5/8} \approx \frac{s_{1/2} + s_{3/4}}{2} \tag{E.7}$$

$$s_{7/8} \approx \frac{s_{3/4} + s_1}{2} \tag{E.8}$$

This algorithm produces a minimum of five landmarks – if the proximity condition is not applied - and a maximum of nine landmarks aiming at the maximum angular dispersion of the landmarks.

In the classroom example, the selected landmarks are presented in Figure 2.

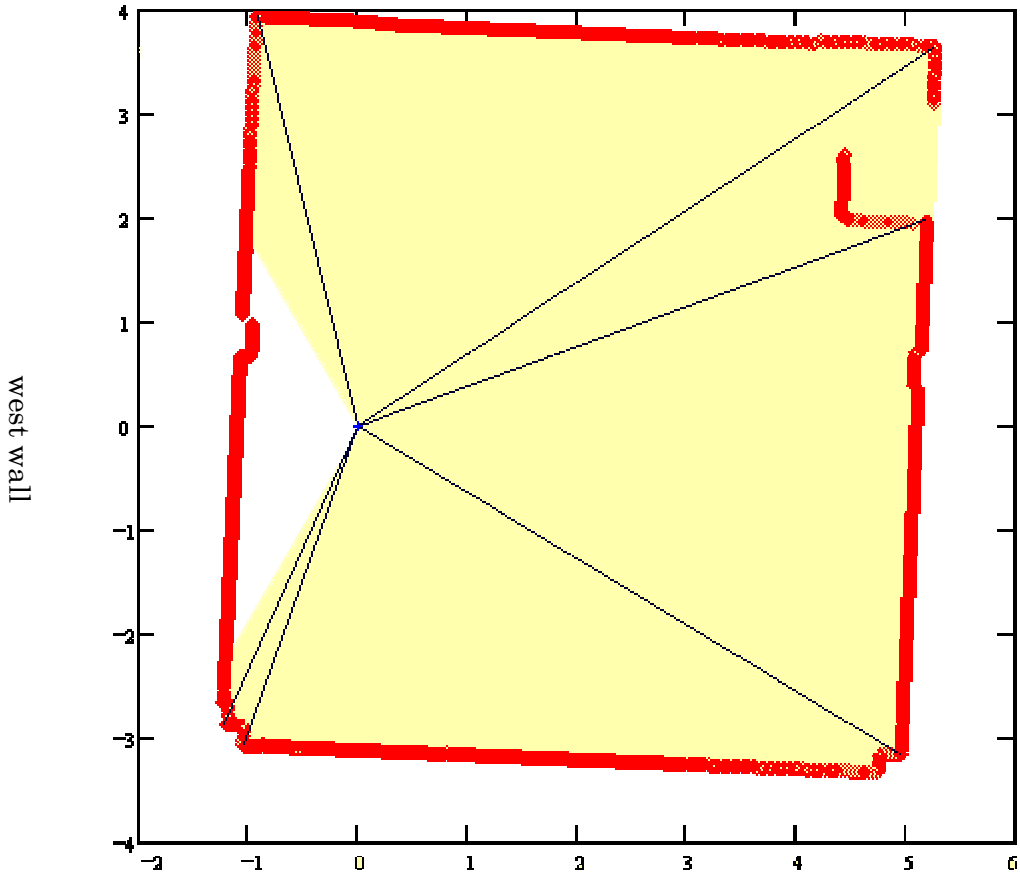


Figure 2 – Selected landmarks from the simulated scan

The benefits of the landmarks selection are the reduction of the computation effort and increased robustness in the results because the landmarks in close vicinity impose similar equations, although not equal, and it is impossible to satisfy them simultaneously.

Appendix F

Companion CD contents

The dissertation includes a companion CD where the author takes advantage of the Web browsing facilities to present the algorithms in an interactive manner.

The CD-ROM is formatted in DOS/Windows format. To browse the CD contents an Internet browser such as Netscape Navigator or Microsoft Internet Explorer. The 4.0 versions or later of these browsers or compatible is recommended for optimal experience.

F.1 The dissertation in Adobe PDF format

0. Prologue
1. Introduction
2. Frame Localisation Algorithm
3. Likelihood Test
4. Approximate Localisation
5. Conclusions
6. References and Appendices

F.2 Dynamic tutorials:

- Frame Localisation
- Likelihood Test
- Error Descent

F.3 Other publications by the author in Adobe PDF format

- *A Multi-Layer Robot Localisation Solution using a Laser Scanner on Reconstructed 3D Models*, João Gomes-Mota, Maria Isabel Ribeiro, 6th

Symposium on Intelligent Robotic Systems, Edinburgh, United Kingdom, July 1998.

- *Localisation of a Mobile Robot using the Laser Scanner on Reconstructed 3D models*, João Gomes-Mota, Maria Isabel Ribeiro, 3rd Portuguese Conference on Automatic Control, Coimbra, Portugal, September 1998.

- *RESOLV: da Robótica à realidade virtual*, João Gomes-Mota, Maria Isabel Ribeiro, V Jornadas Tecnico-Científicas, DEE, Escola Superior Tecnologia, Instituto Politécnico de Setúbal Portugal, Abril 1999

F.4 Web sites

- The author's site
- The RESOLV site
- The Mobile Robotics Lab at ISR/IST site

The web sites are dated of November 2001.