

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Localização para *Smart Devices* tirando partido de *iBeacons*

Marcos Brandão Duarte



Mestrado Integrado em Engenharia Informática e Computação

Orientador: Armando Jorge Miranda de Sousa

Co-orientador: António Miguel Pontes Pimenta Monteiro

29 de Fevereiro de 2016

Localização para *Smart Devices* tirando partido de *iBeacons*

Marcos Brandão Duarte

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo Júri:

Presidente: Carlos Manuel Milheiro de Oliveira Pinto Soares

Arguente: Helena Rodrigues

Vogal: Armando Jorge Miranda de Sousa

29 de Fevereiro de 2016

Resumo

O estado atual da tecnologia no ano de 2016 não oferece ainda uma solução sem limitações para a localização de bens ou pessoas.

O presente trabalho tira partido da tecnologia Bluetooth v4.0 também conhecido como “*Bluetooth Low Energy*”, concretamente dos iBeacons. E pretende caracterizar a qualidade da localização que é possível alcançar num ambiente com diversos destes dispositivos. Este tipo de beacons ativos para localização são considerados como tendo custo acessível e baixo consumo elétrico, sendo uma opção válida quando é necessário acrescentar elementos de localização ao ambiente.

Começou-se por caracterizar experimentalmente a curva de distância com relação a força do sinal recebido (RSSI - Received Signal Strength Indicator) para os iBeacons da empresa Kontakt.io. Esta anuncia custos de cerca de 20 euros por beacon, que são detetáveis a aproximadamente 50 m, com a respetiva pilha durando no mínimo um ano.

Esta dissertação propõe uma estratégia de localização que tira partido da informação de RSSI e da identificação dos iBeacons detetados num dado instante para produzir uma estimativa de localização (dado que se considera que todos os iBeacons estão em posições conhecidas).

A implementação utilizou o Sistema Operativo Android e dois Smartphones de gama de consumo para a validação das propostas.

O sistema foi ainda testado a bordo do robô para a agricultura AGROB V14 que se destina a trabalhar em zonas onde o GPS não estará disponível.

A experiência realizada com o referido robô e um Smartphone Samsung Galaxy S4 num percurso de cerca de 100 m com utilização de dados de 3 iBeacons detetados em simultâneo revelaram um precisão em torno de 1.5 m.

Abstract

The current state of technology in the year 2016 does not offer a solution without limitations to the location of assets or people.

This work takes advantage of Bluetooth v4.0 technology also known as “*Bluetooth Low Energy*” present in the iBeacons. And tries to characterize the quality of the location that can be achieved in an environment with several of these devices. This type of active beacons for location are considered as having affordable cost and low power consumption, being a valid option when you need to add location elements to the environment.

It was started to experimentally characterize the distance curve related to the Received Signal Strength Indicator (RSSI) of the iBeacons from Kontakt.io company. This company announces costs of about 20 euros per beacon, that can be detectable from approximately 50 m, with respective battery lasting at least one year.

This dissertation proposes a localization strategy that takes advantage of the RSSI information and the identification of iBeacons detected in a given time to produce an estimated location (as it is considered that all iBeacons are at known positions).

The implementation used the Android operating system and two consumer range mobile phones to validate the proposals.

The system was also tested on board the agriculture robot AGROB V14 that is intended to work in areas where GPS is not available.

The experiment done with said robot and a Samsung Galaxy S4 in a course of about 100 m and using data of 3 simultaneously detected iBeacons showed an accuracy around 1.5 m.

Agradecimentos

Aos meus pais que me deram a possibilidade de prosseguir os meus estudos até o nível ao qual estou hoje. À minha esposa que me apoia e acompanha desde 2001 em todos os momentos de dificuldade e também nos de felicidade. À minha irmã que me apoiou em todo o percurso académico. À toda a minha família em geral e aos meus amigos mais próximos e aos mais distantes, mesmo àqueles que já não estão entre nós. À todos os Professores e comunidade FEUP, em especial aos meu orientadores, Prof. Armando Jorge Miranda de Sousa e Prof. António Miguel Pontes Pimenta Monteiro, ao Dr. Filipe Neves dos Santos que deu um grande apoio na publicação de um artigo relacionado com esta dissertação. Enfim, à todos aqueles que me ajudaram a ser quem sou hoje.

Marcos Brandão Duarte

*“Aprender é a única coisa de que a mente nunca se cansa,
nunca tem medo e nunca se arrepende.”*

Leonardo da Vinci

Conteúdo

1	Introdução	1
1.1	Contexto/Enquadramento	1
1.2	Motivação e Objetivos	1
1.3	Estrutura da Dissertação	2
2	Revisão Bibliográfica	3
2.1	Introdução	3
2.2	Tecnologias para localização	3
2.3	Problemas na recepção dos sinais	4
2.4	Algoritmos e correção de erros	4
2.5	Resumo e Conclusões	4
3	Definição do problema	5
3.1	Localização	5
3.2	<i>Smart Devices</i>	6
3.2.1	Dispositivo com SO Android	6
3.2.2	<i>iBeacons</i>	6
3.2.3	Aplicação para o SO Android	8
3.3	Resumo e Conclusões	8
4	Implementação	9
4.1	Aplicação para o SO Android	9
4.1.1	Modo de Administração	10
4.1.2	Modo de Caracterização de <i>iBeacon</i>	10
4.1.3	Modo de Teste de Localização	11
4.2	Recolha de dados e validação de resultados	12
4.2.1	Caracterização dos <i>iBeacons</i>	12
4.2.2	Algoritmo de multilateração	14
4.2.3	Teste estático	15
4.2.4	Testes dinâmicos	17
4.3	Resumo e Conclusões	23
5	Conclusões e Trabalho Futuro	25
5.1	Satisfação dos Objetivos	25
5.2	Trabalho Futuro	25
	Referências	27
A	Artigo - Agricultural Wireless Sensor Mapping for Robot Localization	31

CONTEÚDO

Lista de Figuras

4.1	Ecrãs utilizados no modo de administração.	10
4.2	Principais ecrãs do modo de caracterização.	11
4.3	Principais ecrãs do modo de testes de localização.	12
4.4	Valores médios de RSSI para caracterização com Samsung Galaxy S4.	13
4.5	Valores médios de RSSI para caracterização com Vodafone Smart Ultra 6.	13
4.6	Valores estimados da distância com Samsung Galaxy S4.	14
4.7	Valores estimados da distância com Vodafone Smart Ultra 6.	15
4.8	Local de teste de multilateração.	16
4.9	Resultados dos testes de multilateração.	18
4.10	Local dos testes de passagem em região.	19
4.11	Resultados dos testes de passagem em região.	20
4.12	Local de teste com robô agricultor AGROB V14.	21
4.13	Resultados obtidos.	22
4.14	Matriz de localização para o <i>iBeacon</i> com Minor 64416	22

LISTA DE FIGURAS

Lista de Tabelas

4.1	Constantes para o modelo referido na Equação 4.1	14
-----	--	----

LISTA DE TABELAS

Abreviaturas e Símbolos

API	Interface de Programação de Aplicações
BLE	Bluetooth de Baixo Consumo
BT	Bluetooth
GATT	Atributo Genérico
GLONASS	Sistema de Navegação Global por Satélite
GPS	Sistema de Posicionamento Global
GSM	Sistema Global para Comunicações Móveis
IMU	Unidade de Medição Inercial
IP	Protocolo de Internet
RF	Rádio Frequência
RFID	Identificação por Rádio Frequência
RSSI	Indicador da Força do Sinal Recebido
SDK	Kit de Desenvolvimento de Software
SLAM	Localização e Mapeamento Simultâneos
SO	Sistema Operativo
UDP	<i>User Datagram Protocol</i>

Capítulo 1

Introdução

1.1 Contexto/Enquadramento

Hoje em dia, a localização de objetos e pessoas ou de quaisquer alvos em geral traz imensas vantagens como a autonomia e independência de quem ou do que beneficia com a sua utilização. Encontra-se largamente testada e experimentada através de diversas tecnologias. Talvez, a tecnologia mais conhecida seja a de localização por satélite, como é o caso do GPS¹/GLONASS².

Contudo existem muitas outras tecnologias, como a utilização dos pontos de acesso de uma rede sem fios, das redes de endereços IP, da rádio frequência, das redes GSM, da visão, do laser, do Bluetooth, sem prejuízo de outras.

Para a escolha de cada uma dessas tecnologias, é preciso que se analise quais os objetivos pretendidos e se verifique para cada caso qual será a mais adequada.

1.2 Motivação e Objetivos

Com a massificação dos dispositivos móveis de última geração, consegue-se ter acesso a cada vez mais capacidade de processamento e com uma variedade de componentes que frequentemente trazem novas e aprimoradas funcionalidades.

No final de Junho de 2010[Blu10], surgiu o Bluetooth v4.0, também conhecido como “*Bluetooth Low Energy*” (BLE)³ e com isto, a mundialmente conhecida empresa de tecnologia – Apple, lançou e divulgou estar presente nos seus dispositivos os *iBeacons*. Embora não tenha surtido logo o impacto esperado, outras empresas começaram a fabricar e disponibilizar estes *iBeacons* colocando a disposição dos seus clientes, API’s, SDK’S e aplicações para o uso dos mesmos.

Assim, nesta dissertação, na tentativa de clarificar as expectativas relativas a localização utilizando *iBeacons*, são estudados métodos que recorrem a utilização da tecnologia existente nos dispositivos móveis mais atuais e nos *iBeacons*, aproveitando as suas apelativas características

¹Sistema de localização por satélite dos EUA.

²Sistema de localização por satélite da Rússia.

³Uma breve descrição e algumas das suas vantagens podem ser consultadas em: <https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics/low-energy>.

Introdução

de baixos consumo e custo. Mais concretamente, através de uma aplicação para o SO Android⁴, criada para o efeito, obtêm-se os sinais emitidos pelos *iBeacons* e captados por um dispositivo móvel, recolhendo dados como o RSSI que permitem calcular a localização de um qualquer alvo em um mapa determinado.

Este tipo de solução permite a localização mais precisa ou menos, conforme a necessidade de cada caso e em diversas áreas de aplicação. Uma delas é a área da agricultura, em específico o caso das vinhas na região do Douro em Portugal, onde as características do terreno íngreme e em socacos impedem a utilização do GPS, pois a qualidade do sinal é reduzida devido ao seu bloqueio que o torna frequentemente indisponível. Portanto, foram feitos diversos testes para caracterização do funcionamento dos *iBeacons*, em particular, com a ligação a um robô de agricultura.

1.3 Estrutura da Dissertação

Para além desta introdução, esta dissertação contém mais 4 capítulos. No Capítulo 2 é descrito o estado da arte e são referidos os estudos existentes no âmbito da localização. No Capítulo 3 define-se o problema e apresentam-se os conceitos-chave do tema, a localização, os *Smart Devices* e os *iBeacons*. No Capítulo 4 explica-se como se implementaram os conceitos que foram sendo descritos nos capítulos anteriores e como foram recolhidos, testados e validados os resultados obtidos. No capítulo 5 conclui-se aquilo que foi o objetivo final desta dissertação e o seu cumprimento, fazendo uma análise do trabalho futuro.

⁴<http://www.android.com/>

Capítulo 2

Revisão Bibliográfica

2.1 Introdução

A localização pode ser realizada com recurso a diversas tecnologias, dentre as quais: a localização baseada em RF [BP00, OA14], RFID [LL06, HBF⁺04, NLLP04, KHLH03], Bluetooth [Oks14, MnOJD12, Zhu12, ABFB10, Apa08, Yip05], Wi-Fi [Zhu12, TgStMHj10, Apa08, KHNF05], sem prejuízo de outras.

Claro que todas estas tecnologias estão suscetíveis às interferências sofridas pelos sinais transmitidos o que implica a necessidade da correção de erros, como é o caso da utilização dos filtros de kalman [FWZJ14, ODAA13, OA14, OLAA14, TgStMHj10].

Nesta dissertação está em causa a utilização do RSSI, pelo que é necessário também o seu estudo [RHYHHW⁺08, ZZZZ08, ODAA13, OLAA14].

2.2 Tecnologias para localização

Todas as tecnologias mencionadas anteriormente têm um fator comum, utilizam sinais de rádio emitidos por pontos de referência. Estes, são captados por outros dispositivos que os usam da maneira que necessitam. Para a localização, existem diversos métodos que podem ser utilizados para que se possa tentar calcular a posição de um alvo determinado. Alguns deles, são:

- **ângulo de chegada** - obtenção do ângulo de chegada de um sinal de um determinado ponto;
- **tempo de chegada** - cálculo do tempo de chegada de um sinal como resposta a um pedido feito a determinado ponto;
- **força do sinal recebido** - receção de dados que indiquem a força do sinal recebido de um determinado ponto.

Estes dados, podem ser utilizados em algoritmos que calculam a posição de um determinado alvo.

2.3 Problemas na recepção dos sinais

Um dos maiores obstáculos existentes para a utilização destes dados é a variação sofrida no caminho percorrido desde a origem do sinal, até o dispositivo que o capta. Os ambientes onde se utilizam estes sinais são sempre suscetíveis de causar interferências que trazem problemas como:

- **reflexão** - obstáculos existentes no ambiente fazem com que o sinal seja refletido e redirecionado;
- **difração** - o sinal é atenuado, atravessando o obstáculo;
- **propagação em múltiplos caminhos** - devido aos dois anteriores, o sinal segue diferentes caminhos.

2.4 Algoritmos e correção de erros

Todos os algoritmos que possam ser utilizados terão sempre de prever erros nos dados. Assim, todos os dados podem e devem ser muito bem analisados para sejam corrigidos conforme necessário.

Alguns dos algoritmos utilizados são:

- **multilateração** - baseia-se na obtenção de sinais de diversas origens e faz o cálculo da posição de cada um com base num círculo centrado no dispositivo que capta o sinal, tentando encontrar uma área comum dos mesmos;
- **triangulação** - semelhante ao anterior, mas utiliza também ângulos de chegada dos sinais;
- **impressão digital de RSSI** - divide a área onde se pretende fazer a localização por diversas células e mede, em cada uma, os RSSI's das diversas origens, construindo uma base de dados com esses valores, para que posteriormente se possam comparar com os dados recebidos por um alvo.

Para correção do erro, um dos métodos utilizados são os filtros de Kalman. Numa análise muito superficial, este método, utiliza uma amostra suficientemente larga dos dados que necessitam de alguma correção e aprimora-os de modo a que se aproximem dos valores esperados.

2.5 Resumo e Conclusões

A literatura existente no âmbito do tema desta dissertação é vasta, abordando o problema da localização e diversas técnicas para a tornar o mais precisa possível. Cada uma deve ser analisada e escolhida de forma a tentar manter a estabilidade necessária para a solução pretendida.

No caso concreto desta dissertação, utilizaremos o Bluetooth, o RSSI e a sua caracterização baseada em modelos de *path loss*.

Capítulo 3

Definição do problema

Existem 3 conceitos-chave que são utilizados, o de localização, *Smart Devices* e *iBeacons*. Adiante faz-se uma breve explicação do significado de cada um deles e como se interligam para o fim proposto.

3.1 Localização

A localização teve a sua popularização com a massificação dos dispositivos que utilizam as tecnologias de localização por satélite (GPS/GLONASS). E é amplamente utilizada em sistemas que necessitam de autonomia, nomeadamente na área da robótica. Existem muitas outras tecnologias para a localização de pessoas ou coisas, que têm de ter em conta um aspeto essencial, o ambiente onde se encontra o alvo da localização. Para isto, definem-se dois tipos de ambiente:

- **fechado** - interior de edifícios;
- **aberto** - fora de edifícios.

Esta distinção tem de ser feita, pois cada tecnologia utilizada pode ter o seu funcionamento influenciado conforme os ambientes em que estamos.

Outros fatores que devem ser levados em consideração para a localização:

- interferências causadas por coisas que pertencem ao ambiente onde está o alvo;
- precisão que se quer obter;
- algoritmos a utilizar no cálculo da localização, seus tempos de execução e cargas de processamento;
- necessidade de um mapa com pontos de referência, que pode ser predefinido ou construído à medida que seja necessário com base em algoritmos de Mapeamento e Localização Simultâneos (SLAM);

3.2 *Smart Devices*

A utilização desta nomenclatura está relacionada com a que é utilizada pelo Bluetooth SIG (Grupo de Interesse Especial do Bluetooth). De acordo com as suas definições, um *Smart Device*¹, nada mais é, do que um dispositivo que tem por base a utilização do protocolo recente da especificação Bluetooth (a versão 4.0 ou BLE).

Em específico, estarão disponíveis dois tipos destes dispositivos:

- **dinâmicos** - que podem se mover no espaço, dispositivos móveis com SO Android;
- **estáticos** - que estarão fixos no espaço, os *iBeacons*.

3.2.1 Dispositivo com SO Android

Inicialmente, um raspberry-pi² com um adaptador Bluetooth v4.0 por razões baseadas nas características de baixo custo, mobilidade, devido ao fato de ser um dispositivo pequeno e com baixo consumo de energia e com a possibilidade de instalação do SO Android. Esta configuração teve de ser abandonada pois o suporte pelo Android ao “*Bluetooth Low Energy*” só é dado a partir da sua versão 4.3, que ao tempo do início deste trabalho não era suportada pelo raspberry-pi.

Com isto, tomou-se a decisão de utilizar dois dispositivos móveis, ambos *Smartphones*. As características comuns são a presença do Bluetooth v4.0 e do SO Android sempre em versão acima da 4.3, que como dito, são condições necessárias para a ligação aos *iBeacons*. Os dispositivos são os seguintes:

- **Samsung Galaxy S4**³
- **Vodafone Smart Ultra 6**⁴, que tem como base o ZTE Blade S6⁵

3.2.2 *iBeacons*

Foram lançados pela Apple e entretanto, diversas empresas começaram a fabricá-los de acordo com as suas especificações.

São simples emissores Bluetooth, que enviam pacotes de dados com determinadas informações.

Em específico serão usados os *iBeacons* fornecidos pela empresa Polaca Kontakt⁶.

Como já referido anteriormente, estes dispositivos são baseados na versão 4.0 do Bluetooth.

¹<http://www.bluetooth.com/Pages/Bluetooth-Smart-Devices.aspx>

²<http://www.raspberrypi.org/>

³Especificações disponíveis em: <http://www.samsung.com/pt/consumer/mobile-devices/smartphones/galaxy-s/GT-I9505DKYTPH>

⁴Especificações disponíveis em: <https://loja.vodafone.pt/telemoveis/vodafone/smart-ultra-6-cinza?src=ON>

⁵Especificações disponíveis em: http://www.ztedevice.com/product/smart_phone/af87142a-2581-43fd-814b-cc0c50462ef7.html

⁶<http://www.kontakt.io>

Definição do problema

O funcionamento desta versão do Bluetooth [Blu10], específica a utilização do perfil GATT que funciona sobre o Protocolo ATT. Este define operações comuns e uma *framework* para utilização dos dados guardados e transportados.

O GATT funciona numa arquitetura cliente/servidor, definindo-se estes papéis a seguir:

- **servidor** - guarda os dados que serão transportados pelo Protocolo ATT e aceita pedidos, comandos e confirmações de clientes GATT;
- **cliente** - envia pedidos, comandos e confirmações através do Protocolo ATT para os servidores GATT.

Portanto, o GATT está dividido em Atributos, que por sua vez contêm Serviços com determinadas Características. Estas têm um determinado valor e uma descrição.

Algumas das principais características disponibilizadas por estes *iBeacons* são [Kon14]:

- **“TX Power Level”** - permite a definição da potência do sinal de transmissão para os seguintes valores -30, -20, -16, -12, -8, -4, 0 ou 4 dBm, o valor predefinido são -12 dBm;
- **“Battery Level”** - informa o nível da bateria do *iBeacon*, facilitando a tarefa de monitorização e de troca das mesmas;
- **“Advertising Interval”** - permite definir o tempo de intervalo entre os pacotes de anúncio que serão adquiridos pelos clientes GATT para valores entre os 20 ms e 10,240 ms sempre em intervalos de 625 μ s, o valor predefinido são 350 ms.

Uma lista completa dos Atributos, Serviços e Características disponibilizados pelos *iBeacons* da Kontakt podem ser vistos na sua *Datasheet* [Kon14].

O funcionamento dos *iBeacons* está baseado na emissão de pacotes de dados por via do Bluetooth. Existem dois tipos de pacotes de dados que são emitidos por estes dispositivos:

- **“Pacotes de Anúncio”** - pacotes de dado que estão a ser emitidos com a frequência definida pela característica *Advertising Interval*;
- **“Pacotes de Resposta à Procura”** - são pacotes de dados que são emitidos caso um dispositivo que leia um dos pacotes anteriores, requeira mais dados.

Assim, os *iBeacons* estão sempre a emitir os pacotes de anúncio, ao que podem vir a emitir um pacote de resposta à procura, caso seja requerido e voltando a emitir os primeiros.

Esta versão dos *iBeacons* da Kontakt também possui uma característica de segurança que é a necessidade de colocação de uma senha quando se tenta conectar ao *iBeacon* para alteração ou leitura dos seus dados.

Uma parte fundamental da estrutura dos pacotes dos *iBeacons* e que não pode ser esquecida são os seus identificadores, através dos quais podemos de forma fácil verificar qual o *iBeacon* que estamos a visualizar ou ao qual queremos tentar conectar. São os seguintes dados:

Definição do problema

- **UUID** - 32 caracteres hexadecimais representados em 32 bytes;
- **Major** - valor inteiro representado em 2 bytes;
- **Minor** - valor inteiro representado em 2 bytes;

Para além destes dados constantes dos pacotes de dados, há também um ID fixado numa etiqueta na parte inferior do *iBeacon*.

3.2.3 Aplicação para o SO Android

Para interligação dos elementos anteriores, foi criada uma aplicação Android recorrendo ao desenvolvimento no IDE Eclipse e à linguagem Java.

Esta aplicação torna dos dispositivos móveis em clientes GATT que acessam os dados dos *iBeacons* que são os servidores GATT. Fazendo toda a recolha de dados necessários para análise.

3.3 Resumo e Conclusões

Portanto, foram utilizadas as tecnologias Bluetooth mais recentes presentes nos dispositivos apresentados, para se proceder a localização de um alvo móvel. Procurou-se ir de encontro com os métodos já utilizados para a estimação da posição de dispositivos em redes de sensores que utilizam a rádio frequência e também da utilização de meios para correção de erros.

Capítulo 4

Implementação

4.1 Aplicação para o SO Android

A aplicação é uma parte fundamental de todo o sistema, e onde está presente o algoritmo para cálculo da localização.

Esta aplicação torna o dispositivo num cliente GATT que comunica diretamente com os servidores GATT, os *iBeacons*.

Essencialmente, a aplicação tem os seguintes modos de funcionamento:

- **administração** - serve para verificar ou alterar alguns dos dados disponibilizados pelos Atributos GATT dos *iBeacons*.
- **caracterização de *iBeacon*** - recolhe dados de RSSI de um *iBeacon* ao qual o dispositivo móvel está conectado, guardando-os em ficheiro para posterior análise.
- **teste de localização** - utilizado para a localização em si, apenas descobre os *iBeacons* e mostra ao utilizador informação sobre o essencial para a localização, gravando dados da recolha do RSSI dos *iBeacons* visíveis em ficheiros para análise posterior;

Há um conjunto de boas práticas[[BHX13](#)] que são respeitadas para a utilização do Bluetooth pela aplicação:

1. saber se o dispositivo tem Bluetooth;
2. se sim, saber se o Bluetooth está ligado;
3. estando ligado, ficar num estado de espera, procurando por dispositivos BLE;
4. se requerido, ligar então ao dispositivo para descobrir os serviços que tem disponíveis;
5. ler/escrever nas características pretendidas;
6. se não pretender mais alterações, voltar ao passo [3](#) ou avançar para o passo [7](#);

Implementação

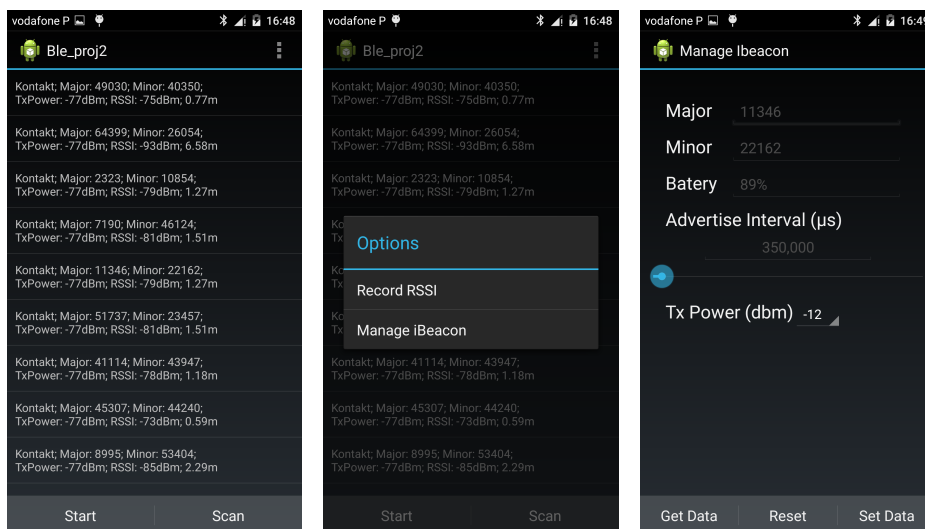
7. terminado o processo, não sendo mais necessário, desligar o Bluetooth.

Note-se que se estiver no modo para testes de localização, apenas é preciso chegar ao passo 3, já que neste ponto já se consegue obter o RSSI, que é o dado essencial para a localização, não sendo necessárias ligações aos dispositivos.

Adiante, especificam-se mais aprofundadamente os modos de funcionamento da aplicação.

4.1.1 Modo de Administração

Neste modo, a aplicação é aberta normalmente. No ecrã inicial, clicando-se no botão “Scan”, localizado no canto inferior direito (ver Figura 4.1(a)). Surge então uma lista dos dispositivos que estão próximos e com um clique longo sobre o item da lista, poderemos seleccionar a opção “Manage iBeacon” que surge no ecrã (ver Figura 4.1(b)), sendo então direccionado para o ecrã seguinte (ver Figura 4.1(c)), onde se pode configurar a potência do sinal dentre as disponibilizadas na opção “TxPower” e o intervalo entre os pacotes de anúncio na opção “Advertising Interval” e visualizar dados como o Major, Minor e a percentagem de bateria.



(a) Ecrã principal

(b) Ecrã com opções para o *iBeacon*.

(c) Ecrã de gestão do *iBeacon*.

Figura 4.1: Ecrãs utilizados no modo de administração.

4.1.2 Modo de Caracterização de *iBeacon*

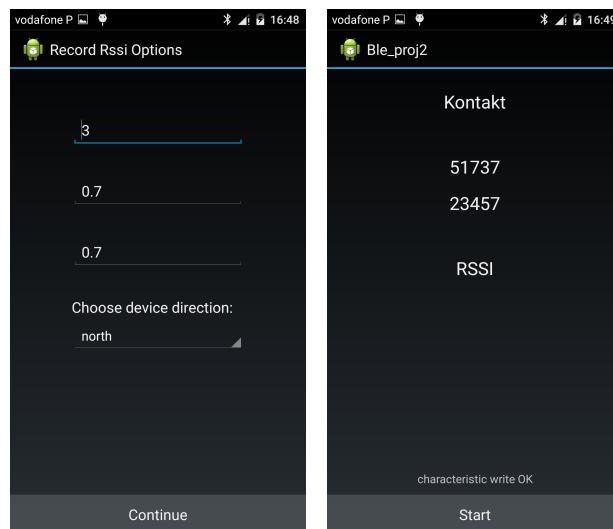
Aqui, seguem-se os passos do modo de administração, mas ao invés de seleccionar a opção “Manage iBeacon”, selecciona-se a opção “Record RSSI”. Após esta escolha, é apresentada um ecrã (ver Figura 4.2(a)) onde se pode escolher diversos parâmetros:

- altura do *iBeacon*;

Implementação

- altura do dispositivo móvel;
- distância entre dispositivos;
- orientação do *iBeacon* relativa ao dispositivo móvel.

Após esta escolha, o dispositivo móvel conecta-se ao *iBeacon* gravando o seu RSSI com uma certa frequência (preferencialmente igual à dos pacotes de anúncio), durante um tempo determinado (ver Figura 4.2(b)). Estes dados são guardados em um ficheiro para análise posterior.



(a) Ecrã com opções da caracterização. (b) Ecrã para recolha do RSSI.

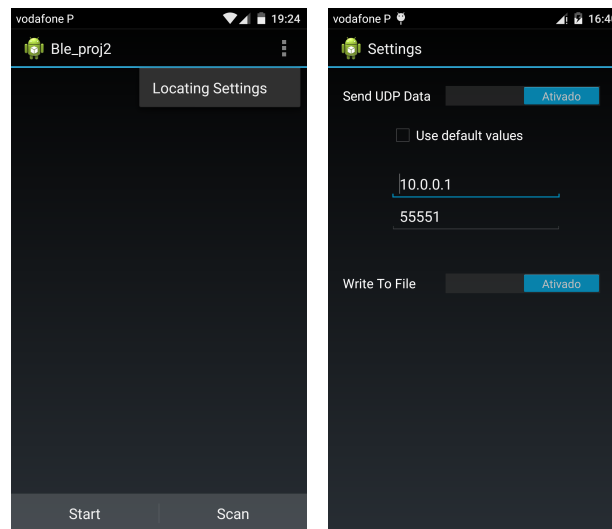
Figura 4.2: Principais ecrãs do modo de caracterização.

4.1.3 Modo de Teste de Localização

Com a abertura da aplicação, ao invés de utilizar o botão “Scan” como nos modos anteriores, utiliza-se o botão “Start” (ver Figura 4.3(a)). A aplicação ficará sempre ligada e a recolher dados dos *iBeacons* visíveis, até que o utilizador interrompa carregando no botão “Stop”. Neste modo, existem duas configurações extras que devem ser realizadas consoante os testes que se esteja a fazer (mais a frente aprofundar-se-á os tipos de testes realizados). Carregando no canto superior direito, no menu de opções, surge a opção “Locating Settings” (ver Figura 4.3(a)). Esta leva-nos a um ecrã (ver Figura 4.3(b)) que permite a seleção de uma opção para envio de dados para um outro dispositivo através de UDP¹, para um socket (conjunto de endereço IP e porta) configurável. Uma segunda opção é se queremos que os dados da localização sejam guardados em ficheiro ou não.

¹Protocolo simples para envio de dados através de pacotes IP.

Implementação



(a) Ecrã com menu de opções. (b) Ecrã com configurações para a localização.

Figura 4.3: Principais ecrãs do modo de testes de localização.

4.2 Recolha de dados e validação de resultados

Para confirmação e validação do funcionamento dos *iBeacons* e da sua utilidade na localização foram realizados diversos testes que se descrevem nas secções seguintes.

4.2.1 Caracterização dos *iBeacons*

Como se pode depreender das características de funcionamento dos dispositivos Bluetooth, há uma ampla variação nos valores de RSSI que se utiliza. Mas ainda há mais aspetos que alteram estes valores e os podem fazer variar de acordo com cada um dos dispositivos móveis utilizados. Conforme os tipos de antenas que utilizam e o seu posicionamento dentro da estrutura física do dispositivo, poderá haver alterações ao modo como devemos utilizar os dados recebidos. Quanto aos *iBeacons*, como são todos iguais, em princípio as suas características mantêm-se semelhantes entre todos os que são utilizados.

Assim, para se poder definir uma função que relacione o RSSI com a distância, é preciso caracterizar o funcionamento dos *iBeacons* com cada dispositivo móvel (esta relação é também sugerida pela AltBeacon na sua *Android Beacon Library* que fornece API's de interação com *iBeacons* ², de modo a que se obtenha melhores resultados nas medições e conversões do RSSI).

Seguindo àquilo que já foi experimentado nos desenvolvimentos feitos na criação das referidas API's, num campo de futebol, foram colocados dois bancos de madeira (para redução das interferências com o sinal Bluetooth) com altura aproximada de 0.7 m. Num dos bancos, colocou-se um

²Disponível em: <https://altbeacon.github.io/android-beacon-library/distance-calculations.html>

Implementação

único *iBeacon* e noutro o dispositivo móvel com SO Android.

Convém referir que estes *iBeacons* vêm configurados de fábrica para transmitir o pacote de anúncio em intervalos de 350 ms, com potência do sinal definida para um valor moderado de 12 dBm, de forma que se prolongue a vida útil da bateria e que se tenha desempenhos minimamente aceitáveis para a localização. Portanto, decidiu-se manter estas opções.

Depois, para distâncias conhecidas (0.25, 0.5, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 16, 18, 20 e 25 m) foram recolhidas as leituras de RSSI, através da aplicação Android. Estes dados são também recolhidos em intervalos de 350 ms durante um período de 2 minutos, sendo então guardadas em um ficheiro 342 amostras do RSSI para cada distância.

Portanto, como foram usados dois dispositivos móveis, foi possível fazer duas caracterizações, uma para cada par de *iBeacon*/Smartphone. Nas Figuras. 4.4 e 4.5, podem se visualizar alguns dos dados recolhidos.

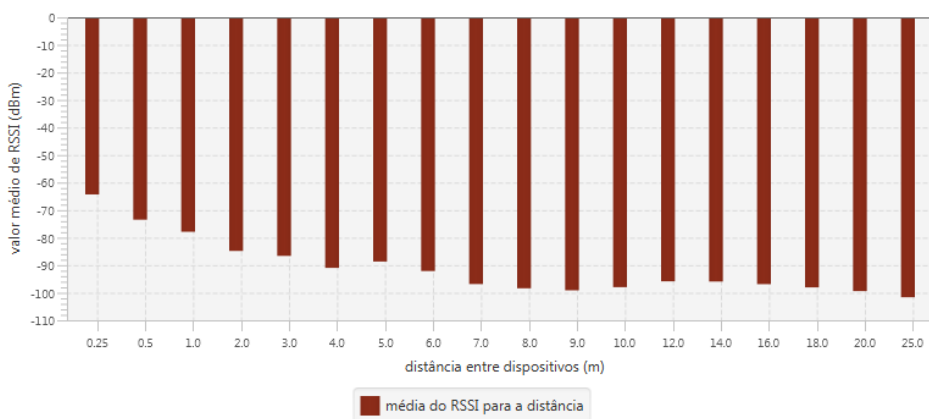


Figura 4.4: Valores médios de RSSI para caracterização com Samsung Galaxy S4.

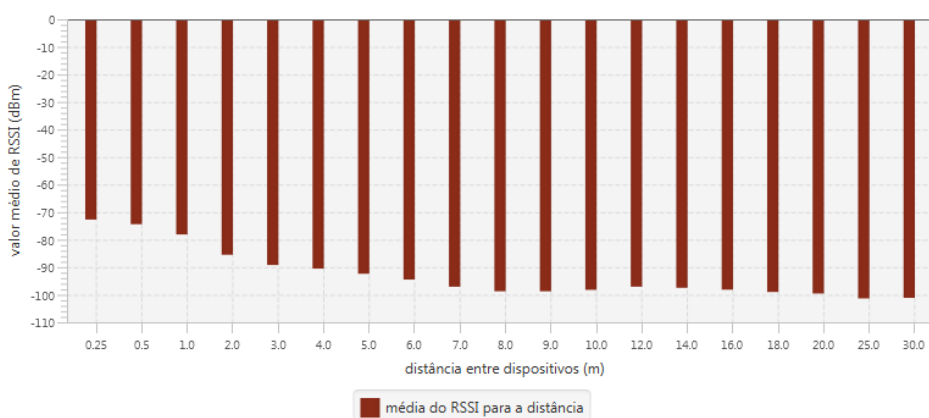


Figura 4.5: Valores médios de RSSI para caracterização com Vodafone Smart Ultra 6.

Após esta recolha, seguindo ainda o que é descrito pela AltBeacon, utiliza-se o seguinte modelo para definição de uma função $f : RSSI(dBm) \rightarrow distância(m)$:

Implementação

$$d = A * \left(\frac{RSSI}{RefPower} \right)^B + C . \quad (4.1)$$

Na Equação 4.1 d é o valor, em metros, da distância que queremos obter, $RSSI$ é o recebido do *iBeacon* pela aplicação Android, $RefPower$ é o RSSI de referência para a potência de transmissão configurada no *iBeacon* para a distância de 1 m entre os dispositivos (este valor de “RefPower” pode ser retirado do “Pacote de Anúncio” ou da calibração que foi feita com o par *iBeacon*/Smartphone, podendo levar a uma maior precisão), A , B e C são constantes.

Com estes dados, utilizando os dados recolhidos, o modelo referido na Equação 4.1 e o método não linear dos mínimos quadrados, as constantes podem ser definidas:

Tabela 4.1: Constantes para o modelo referido na Equação 4.1

Constantes	Samsung Galaxy S4	Vodafone Smart Ultra 6
A	0.3534536	0.0139812
B	14.8393466	27.4110804
C	0.7566785	1.5668953

Usando a Equação 4.1 e os dados constantes da Tabela 4.1, construiu-se um gráfico das distâncias correspondentes ao valor médio do RSSI (ver Figuras 4.6 e 4.7).

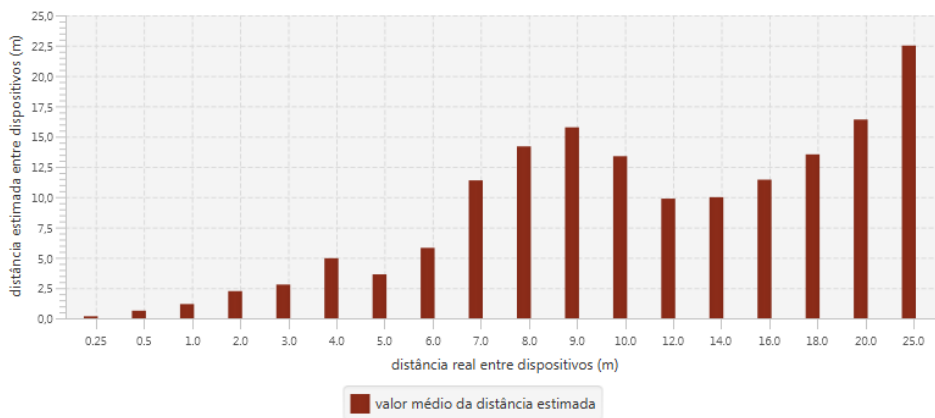


Figura 4.6: Valores estimados da distância com Samsung Galaxy S4.

4.2.2 Algoritmo de multilateração

Antes de se poder continuar com a descrição da recolha dos dados, torna-se necessário explicar o algoritmo que foi utilizado na análise de alguns dos dados recolhidos.

O termo já foi referido na Secção 2.4 e baseia-se na receção e análise dos sinais recebido por diversas fontes, neste caso os *iBeacons*. Também há quem utilize o termo trilateração, já que são necessários pelo menos 3 pontos de referência em posições diferentes, para que se consiga fazer

Implementação

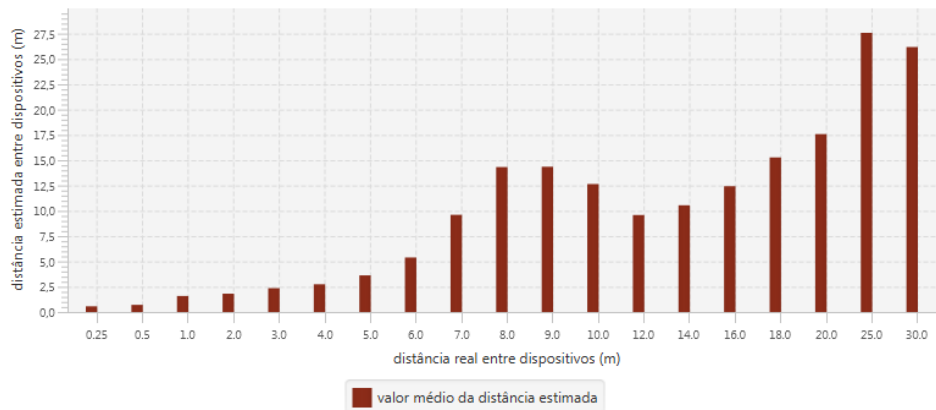


Figura 4.7: Valores estimados da distância com Vodafone Smart Ultra 6.

a junção dos círculos centrados em cada um deles para se poder definir um ponto comum que represente a posição do alvo que se deseja.

No caso em concreto desta dissertação, embora não sejam usados mais do que 3 pontos de referência, o algoritmo prevê a sua utilização, ou até de apenas 2 (como pode ser visto num dos testes que se apresenta a seguir), mas leva sempre em consideração o facto da possibilidade de existência de pontos que estejam na mesma posição, não permitindo que sejam analisados ao mesmo tempo, obrigando sempre a que se faça a utilização de referências que estejam distantes entre si.

Importa também salientar, que uma leitura com os dados de RSSI de um ponto de referência a um dado instante é utilizada mais do que uma vez, já que o algoritmo na análise de uma leitura utiliza sempre os dados de 2 leituras anteriores de mais 2 pontos, que estejam em posições diferentes e dentro de um limite temporal definido. Não há qualquer salto ou inutilização de uma leitura por já ter sido utilizada.

Este algoritmo utiliza uma matriz 2D para referenciar os dados que vão sendo lidos, atualizando a sua probabilidade de ocupação a cada análise. No fim, recorre a conversão desta matriz em uma imagem para que possa ser fácil a visualização do resultado final, como se poderá ver nos resultados dos testes apresentados a seguir.

4.2.3 Teste estático

Um dos testes foi realizado de modo estático. Procedeu-se a montagem de um cenário para o teste, onde os *iBeacons* ficaram fixos em pontos conhecidos e medidos, assim como o dispositivo móvel.

4.2.3.1 Teste de multilateração

No teste de multilateração, foi criado um esquema base de fixação dos *iBeacons*, para permitir o cálculo da localização com uma alta frequência. Como o próprio título indica, recorreu-se ao

Implementação

algoritmo simples de multilateração descrito na Secção 4.2.2.

Assim, recorrendo mais uma vez ao campo de futebol, foram colocados três bancos de madeira com aproximadamente 0.7 m de altura, separados por distâncias aproximadas de 6 m entre cada um deles, formando entre si um triângulo equilátero. Chamemos os bancos de “banco 1”, “banco 2” e “banco 3”.



Figura 4.8: Local de teste de multilateração.

Em cada um dos bancos foram posicionados os *iBeacons* da seguinte forma (para se poder identificar os *iBeacons*, apenas utilizaremos um ID que está presente numa etiqueta por baixo de cada *iBeacon*):

1. **banco 1** - o7nO, gQ0Q, hbdi;
2. **banco 2** - B0Vw, yvlm, gmta, unPx;
3. **banco 3** - 9MBi, gjV2, XK4T.

Foram então realizados 4 testes com esta configuração, onde o que varia é a posição do dispositivo móvel de acordo com a enumeração abaixo relativa a cada um dos testes:

1. entre os 3 bancos, próximo do centro do triângulo formado, mais especificamente, a 3.68 m do banco 1, 3.95 m do banco 2, 3.74 m do banco 3, direcionado para o banco 1;
2. fora do triângulo formado, entre o banco 1 e o banco 2, a 3.46 m do banco 1, 3.16 m do banco 2, direcionado para a linha entre banco 1 e banco 3;
3. fora do triângulo formado, entre o banco 2 e o banco 3, a 1.39 m do banco 2, 5.25 m do banco 3, direcionado para o centro do triângulo;
4. em linha com o banco 1 e o banco 3, a 0.5 m do banco 1 e a 5.5 m do banco 3, direcionado para o banco 2.

Nestes testes, os dados de RSSI foram recolhidos durante dois minutos. Os dados recolhidos são de todos os *iBeacons* visíveis, o que nos traz uma grande quantidade de dados, mas também

Implementação

traz a necessidade de fazer uma análise cuidadosa dos mesmos. O facto de termos apenas 3 bancos com 10 *iBeacons*, faz com que se tenha cerca de 3 vezes mais dados do que teríamos com apenas 1 *iBeacon* em cada banco. Isto faz com que a localização possa ser feita com uma frequência mais alta, já que temos uma leitura a cada 50 a 100 ms (valores médios aproximados obtidos nos testes realizados).

Os dados aqui recolhidos foram analisados em bruto, não se recorrendo a nenhuns métodos para correção de erros. O que se verifica é que como esperado, não se consegue a definição de uma localização exata ou pelo menos de uma pequena área a que se possa cingir a posição do dispositivo móvel. Como se pode ver na Figura 4.9, os números visíveis em cada um dos resultados representam cada um dos bancos mencionados no cenário do teste (número 1 para o banco 1, número 2 para o banco 2, número 3 para o banco 3), cada imagem representa um quadrado com 10 m de lado, o que representa uma resolução³ de 0.1 m/pixel. Com a observação, o que se consegue é definir uma região onde o dispositivo possa estar.

4.2.4 Testes dinâmicos

Nestes testes, ao contrário dos anteriores, os dispositivos móveis recolhem os dados enquanto estão em movimento. Consoante o teste, pode haver ou não o registo da localização do dispositivo móvel com base em outros dados recolhidos, como de GPS ou IMU. Não havendo acesso a dados deste género, há uma descrição do modo como o teste foi realizado.

4.2.4.1 Teste de passagem em região

A montagem deste cenário de teste foi feita numa sala de um jardim de infância, com forma essencialmente retangular, com aproximadamente 6 x 11 m. Foram colocados dois suportes de esferovite com cerca de 1.3 m de altura distanciados de 5 m, centrados com a sala (na Figura 4.10 podemos ver o local de testes). Chamemos os suportes de “suporte 1” e “suporte 2”. Em cada um deles foram colocados os 10 *iBeacons* da seguinte forma (como anteriormente, utilizamos o ID para identificação dos *iBeacons*):

1. **suporte 1** - gQ0Q, B0Vw, yvlm, gmta, 9MBi;
2. **suporte 2** - o7nO, hbdi, unPx, gjV2, XK4T.

Também aqui, foram realizados 4 testes com a configuração descrita, onde com o dispositivo móvel, se realizaram diferentes passagens pelas zonas dos suportes como se descreve a seguir:

1. passagem entre o suporte 1 e o suporte 2;
2. passagem entre o suporte 1 e o suporte 2 exatamente na linha perpendicular à medida dos 2.5 m da linha que liga os dois suportes;
3. entrada na sala e passagem à esquerda do suporte 1;

³De referir que nas imagens impressas a resolução é alterada pela sua redução.

Implementação

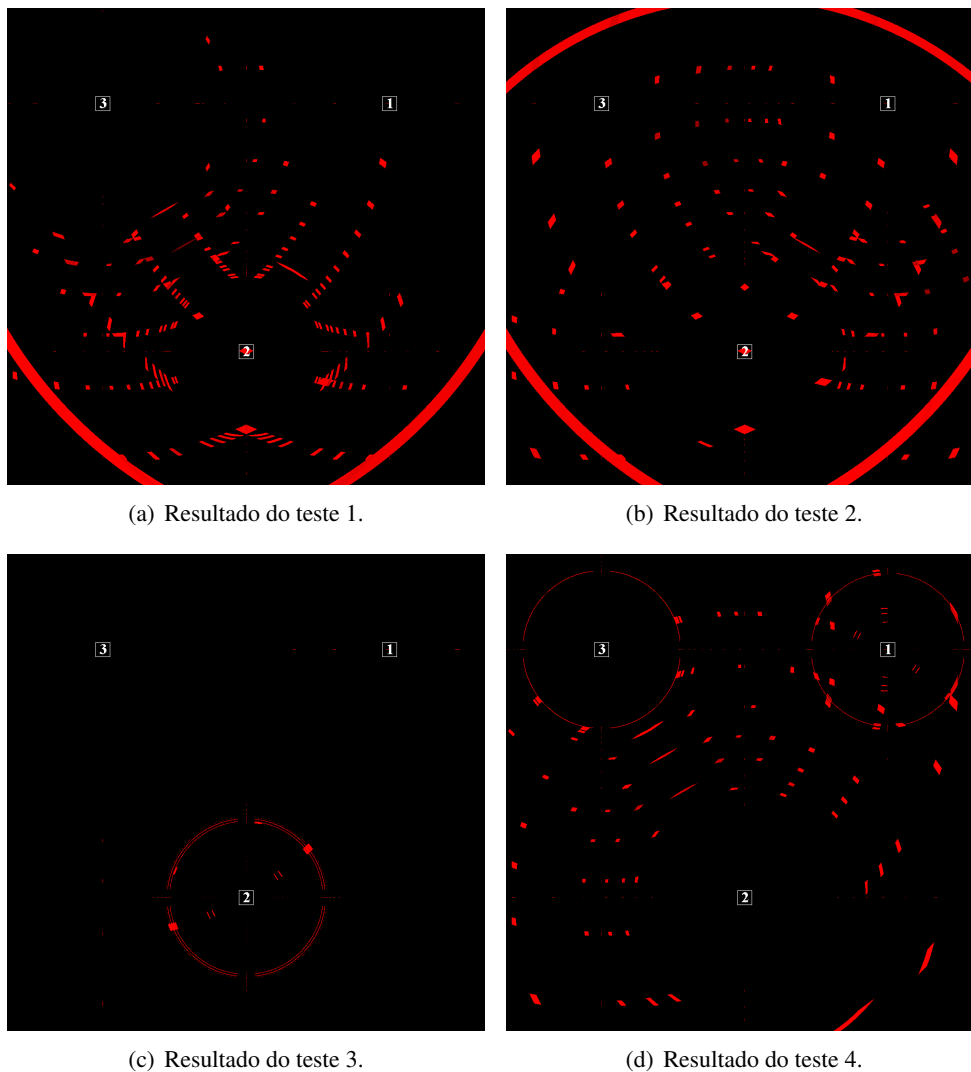


Figura 4.9: Resultados dos testes de multilateração.

4. entrada na sala e passagem à direita do suporte 2.

As análises feitas aos dados deste teste também foram analisadas com recurso ao algoritmo simples de multilateração descrito na Secção 4.2.2. Na Figura 4.11, podem ver-se as imagens geradas com as áreas onde é feito o teste. Os números que são visíveis nas imagens representam cada um dos suportes referidos anteriormente (o número 1 para o suporte 1 e o número 2 para o suporte 2). Cada imagem representa um quadrado com 10 m de lado, o que nos dá uma resolução⁴ de 0.1 m/pixel. Embora não se torne muito claro ao primeiro olhar, já que a figura é uma soma dos possíveis locais de presença do dispositivo móvel ao longo do teste, se verificarmos com cuidado e de acordo com os dados de cada teste, podemos ver que será possível determinar uma região onde o dispositivo passou, ainda que com a utilização de apenas dois pontos de referência.

⁴De referir que nas imagens impressas a resolução é alterada pela sua redução.

Implementação



(a) Vista do suporte 1.

(b) Vista do suporte 2.

Figura 4.10: Local dos testes de passagem em região.

4.2.4.2 Teste com robô para agricultura AGROB V14

Para realização deste teste foi montado um cenário num jardim próximo à Faculdade de Engenharia da Universidade do Porto. Onde foram espalhados os *iBeacons* por bancos desse jardim.

Com vista à recolha dos dados, continuou-se a utilizar o dispositivo móvel Samsung Galaxy S4, mas agora, acoplado à plataforma do robô para agricultura, AGROB V14, cujas especificações podem ser vistas em [NDSSC⁺15]. Neste teste, a aplicação envia todos os dados das leituras dos *iBeacons* para este robô, através de uma ligação à rede sem fios do mesmo utilizando o UDP. Com isto, estes dados são guardados num ficheiro ROSbag⁵ para posterior análise.

Na prática, o teste consistiu na passagem do robô pelos bancos do jardim, recolhendo o RSSI dos *iBeacons* visíveis entre outros dados dos seus sensores, como a localização dada pelo GPS e de velocidade dada pela IMU.

De seguida, os dados são analisados de acordo com o algoritmo de “Procedimento para Mapeamento de Beacons” (BMP) descrito em [DNdSSM15].

Resumidamente, é criada uma matriz para representar a probabilidade da presença de um *iBeacon* em determinado local. Assim, sempre que um *iBeacon* é detectado pela primeira vez, uma matriz é gerada e o seu centro representa a posição do robô naquele instante. Depois, sempre que houver uma nova deteção do mesmo *iBeacon* essa matriz é atualizada refletindo a sobreposição dos dados anteriores com as novas probabilidades da localização do *iBeacon* na matriz.

Claro é, que existem algumas restrições que são utilizadas para o bom funcionamento do BMP. A primeira delas tem relação com o tamanho e o que representa a matriz criada inicialmente. Assim, surge a seguinte equação [DNdSSM15]:

$$Scale * NumberCells = max_{BeaconDistance} \cdot \quad (4.2)$$

⁵Tipo de ficheiro utilizado pelo Robot Operating System (ROS). Uma definição mais profunda pode ser vista em: <http://wiki.ros.org/rosbag>

Implementação

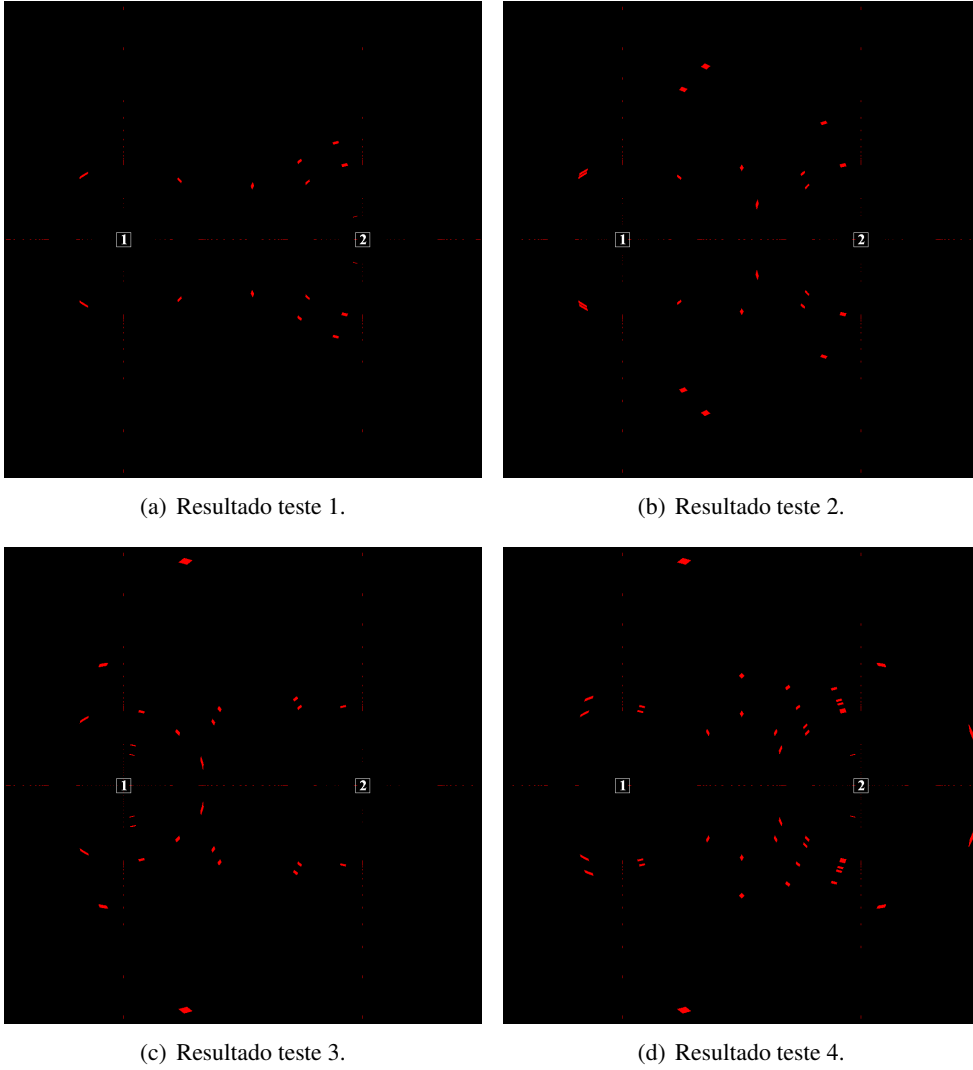


Figura 4.11: Resultados dos testes de passagem em região.

Na Equação 4.2 $max_{BeaconDistance}$ é a distância máxima a qual um *iBeacon* é visível, $Scale$ o valor de escala para os eixos que representam latitude e longitude, e $NumberCells$ a resolução vertical e horizontal da matriz.

A seguir, temos a formulação da probabilidade de existência de um *iBeacon* numa determinada célula da matriz:

$$P_t(x, y) = \frac{1}{\sigma_{RB}\sqrt{2\pi}} e^{-\frac{(d_{RB} - z_t^{beacon})^2}{2\sigma_{RB}^2}} \quad (4.3)$$

Na Equação 4.3, d_{RB} é a distância entre o robô e o centro da matriz, σ_{RB} é a soma das incertezas associadas as posições estimadas do robô e da distância do *iBeacon*.

Portanto, para a junção de várias medidas em diferentes momentos na matriz temos a seguinte equação:

Implementação



Figura 4.12: Local de teste com robô agricultor AGROB V14.

$$P(x,y) = \frac{P_t(x,y) * P_{t-1}(x,y) * P_{t-2}(x,y) * \dots}{\sum_{i=0}^T \sum_{i=0}^X \sum_{j=0}^Y P_t(i,j)} . \quad (4.4)$$

Baseado nestes valores das probabilidades, a posição estimada para um *iBeacon* é dada pela seguinte fórmula:

$$B_{x,y} = \sum_{i=0}^X \sum_{j=0}^Y C_{i,j} * P(i,j) . \quad (4.5)$$

Na Equação 4.5 $B_{x,y}$ é o vetor bidimensional que representa a posição do *iBeacon* e $C_{i,j}$ é o vetor bidimensional que representa a posição central da célula.

Usando o ficheiro ROSbag criado com os dados recolhidos pelo robô no local dos testes, dois testes foram realizados. No primeiro, a localização estimada do robô foi feita com a fusão entre dados do GPS e de velocidade, utilizando um simples filtro de partículas. Na Figura 4.13(c) é mostrada a trajetória estimada do robô com a linha a vermelho e as posições dos *iBeacons* com os alfinetes coloridos. É possível verificar uma melhor estimação da trajetória do robô em comparação com a do GPS mostrada na linha verde.

No segundo teste, a trajetória estimada do robô foi obtida da fusão do GPS, da velocidade e considerando os *iBeacons* já mapeados e suas distâncias observadas, com um simples filtro de partículas. Na Figura 4.13(d) é mostrada a trajetória estimada do robô com a linha verde e as posições dos *iBeacons* com os alfinetes coloridos. Em comparação com o primeiro teste, é possível verificar que a utilização dos dados já recolhidos dos *iBeacons* melhoram as estimações da trajetória do robô e conseqüentemente as dos próprios *iBeacons*.

A Figura 4.14 mostra a matriz de localização do *iBeacon* com Minor 64416 construída com dados do segundo teste em diferentes instantes.

Implementação

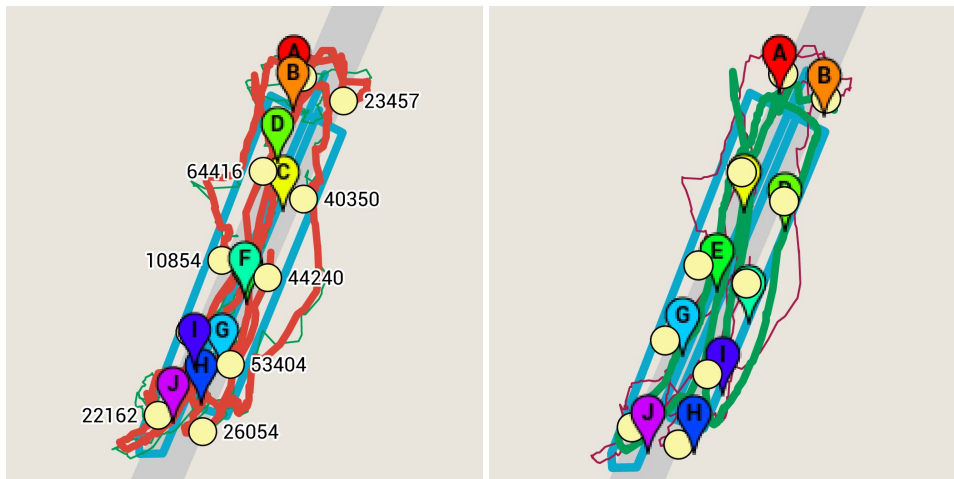
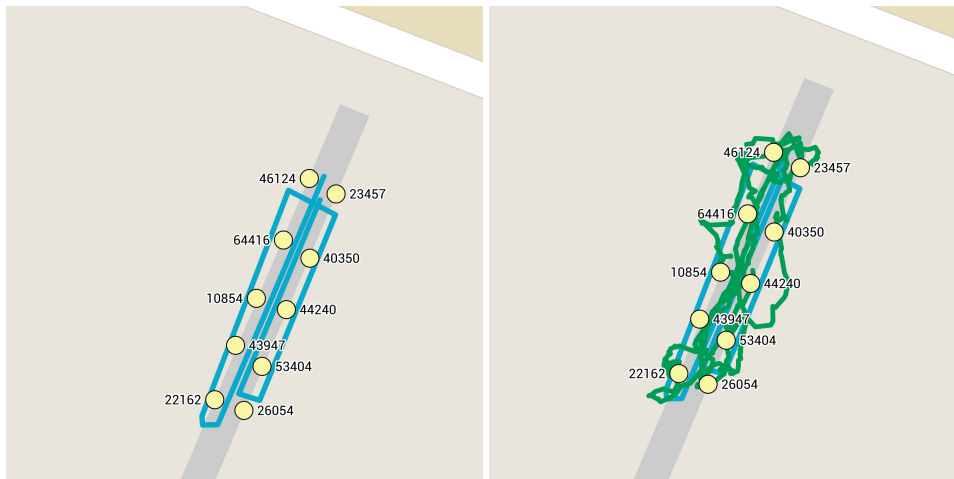


Figura 4.13: Resultados obtidos.



Figura 4.14: Matriz de localização com uma resolução⁶ de 0.25 m/pixel para o *iBeacon* com Minor 64416 aos instantes 10, 230 e 460 s.

Embora em alguns momentos seja necessária a conjugação de outros dados para o mapeamento, pois os erros nas estimativas da posição dada apenas pelo GPS fazem divergir o traçado

⁶De referir que nas imagens impressas a resolução é alterada pela sua redução.

estimado do robô pelo local de testes do real, levando a que esses erros se reflitam na localização dos *iBeacons*, há uma estimação relativa das posições dos *iBeacons* entre si com um erro médio de 0.43 m. Mas o erro médio na estimação individual dos *iBeacons* é próxima de 1.5 m devido aos erros associados a baixa precisão da posição estimada do robô.

4.3 Resumo e Conclusões

Com tudo que foi descrito, a realização dos testes comprova que os *iBeacons* podem ser utilizados para a localização, com menor ou maior precisão. Um ponto desfavorável à sua utilização é a necessidade de uma calibração dos pares *iBeacon/Smartphone* para que haja maior precisão. No caso da utilização do algoritmo BMP, há uma grande dependência da precisão na estimação da posição do robô.

Implementação

Capítulo 5

Conclusões e Trabalho Futuro

5.1 Satisfação dos Objetivos

O trabalho realizado durante esta dissertação está de acordo com os objetivos propostos. Embora as necessidades impostas pelos requisitos do projeto tenham levado a algumas alterações do plano inicial, como que dispositivos móveis seriam utilizados, já que a utilização do Bluetooth v4.0 com o Android, requer que este último esteja pelo menos na versão 4.3, que é onde começa a ser dado suporte a referida tecnologia. Uma outra alteração tem relação com o método utilizado para validação das tecnologias utilizadas, tendo se passado a realizar testes com o AGROB V14 como é referido na Secção 4.2.4.2, deixando de lado as ideias iniciais, por falta de disponibilidade dos dispositivos que se previu usar, como era o caso do sistema de localização Hagisonic Stargaze ¹

Portanto, os objetivos foram alcançados, comprovando que os *iBeacons* podem ser utilizados na localização, com menor ou maior precisão, dependendo apenas da necessidade e dos recursos disponíveis para cada caso. No caso da utilização juntamente com o robô de agricultura, as precisões máximas conseguidas utilizando em simultâneo os dados de 3 *iBeacons* estão em cerca de 1.5 m, como demonstrado na Secção 4.2.4.2. Aqui, fica demonstrada a possibilidade do seu uso em diversos cenários onde seja necessário aliar as suas vantajosas características de baixo custo, consumo e do seu tamanho, independentemente da área específica que necessite de solução para o problema da localização.

5.2 Trabalho Futuro

Há a necessidade de comparação das características entre *iBeacons*, embora seja de presumir que só haja diferença caso se tratem de marcas diferentes, ou dentro da mesma marca, de diferen-

¹É um sistema utilizado para a localização interior de robôs móveis inteligentes. Tem um sensor que analisa uma imagem convertida de raios infravermelhos que são refletidos a partir de um marco passivo colocado no teto do local onde se pretende localizar um alvo. A posição e direção do robô são dados com resolução muito precisa e alta velocidade. Raramente é afetado pelo ambiente, como por raios infravermelhos, luz fluorescente, luz do sol, entre outros. Características mais completas podem ser vistas em: <http://www.hagisonic.com/>

Conclusões e Trabalho Futuro

tes lotes de produção. Outra comparação deve ser a do funcionamento com diferentes níveis de bateria, para verificação da necessidade de adaptação da Equação 4.1. A correção de erros deve ser aplicada aos algoritmos que não tiveram essa consideração.

Existe também a possibilidade de integração dos *iBeacons* com outros tipos de algoritmos e a necessidade de mais testes com vista à confirmação dos dados das caracterizações.

Referências

- [ABFB10] Marco Altini, Davide Brunelli, Elisabetta Farella e Luca Benini. Bluetooth indoor localization with multiple neural networks. In *IEEE 5th Int. Symp. Wirel. Pervasive Comput. 2010*, pages 295–300. IEEE, 2010. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5483748>.
- [Apa08] A fusion method based on bluetooth and WLAN technologies for indoor location. pages 487–491. IEEE, August 2008. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4648042>.
- [BHX13] Sara Sinclair Brody, Rich Hyndman e Matthew Xie. Best Practices for Bluetooth Development — Google I/O 2013, 2013. URL: <https://developers.google.com/events/io/sessions/326240948>.
- [Blu10] Bluetooth SIG. Specification of the Bluetooth System, 2010. URL: https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=229737.
- [BP00] P. Bahl e V.N. Padmanabhan. RADAR: an in-building RF-based user location and tracking system. In *Proc. IEEE INFOCOM 2000. Conf. Comput. Commun. Ninet. Annu. Jt. Conf. IEEE Comput. Commun. Soc. (Cat. No.00CH37064)*, volume 2, pages 775–784. IEEE, 2000. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=832252>.
- [DNdSSM15] M. Duarte, F. Neves dos Santos, A. Sousa e R. Morais. Agricultural wireless sensor mapping for robot localization. In *Robot 2015: Second Iberian Robotics Conference*, pages 359–370, November 2015.
- [FWZJ14] Sheng Feng, Cheng-dong Wu, Yun-zhou Zhang e Zi-xi Jia. Grid-Based Improved Maximum Likelihood Estimation for Dynamic Localization of Mobile Robots. *Int. J. Distrib. Sens. Networks*, 2014:1–15, 2014. URL: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84896471621&partnerID=tZOtx3y1>.
- [HBF⁺04] D. Hahnel, W. Burgard, D. Fox, K. Fishkin e M. Philipose. Mapping and localization with RFID technology. In *IEEE Int. Conf. Robot. Autom. 2004. Proceedings. ICRA '04. 2004*, volume 1, pages 1015–1020 Vol.1. IEEE, 2004. URL: <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=1307283>.
- [KHLH03] A. Kotanen, M. Hannikainen, H. Leppakoski e T.D. Hamalainen. Experiments on local positioning with Bluetooth. In *Proc. ITCC 2003*.

REFERÊNCIAS

- Int. Conf. Inf. Technol. Coding Comput.*, pages 297–303. IEEE Comput. Soc, 2003. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1197544>.
- [KHNF05] T. Kitasuka, K. Hisazumi, T. Nakanishi e A. Fukuda. WiPS: Location and Motion Sensing Technique of IEEE 802.11 Devices. In *Third Int. Conf. Inf. Technol. Appl.*, volume 2, pages 346–349. IEEE, 2005. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1488983>.
- [Kon14] Kontakt. Kontakt.io Beacon Datasheet v2.0, 2014. URL: <http://docs.kontakt.io/beacon/kontakt-beacon-v2.pdf>.
- [LL06] Hyun-jeong Lee e Min Lee. Localization of Mobile Robot Based on Radio Frequency Identification Devices. In *2006 SICE-ICASE Int. Jt. Conf.*, pages 5934–5939. IEEE, 2006. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4108641>.
- [MnOJD12] Mario Muñoz Organero, Pedro J. Muñoz-Merino e Carlos Delgado Kloo. Using bluetooth to implement a pervasive indoor positioning system with minimal requirements at the application level. *Mob. Inf. Syst.*, 8(1):73 – 82, 2012. URL: <http://iospress.metapress.com/content/2n2321462401t3g7/>.
- [NDSSC⁺15] F. Neves Dos Santos, H. Sobreira, D. Campos, R. Morais, A.P. Moreira e O. Contente. Towards a reliable monitoring robot for mountain vineyards. In *Autonomous Robot Systems and Competitions (ICARSC), 2015 IEEE International Conference*, pages 37–43, April 2015.
- [NLLP04] Lionel M. Ni, Yunhao Liu, Yiu Cho Lau e Abhishek P. Patil. LANDMARC: Indoor Location Sensing Using Active RFID. *Wirel. Networks*, 10(6):701–710, November 2004. URL: <http://link.springer.com/10.1023/B:WINE.0000044029.06344.dd>.
- [OA14] Luis Oliveira e Luis Almeida. RF-based Relative Position Estimation in Mobile Ad-Hoc Networks with Confidence Regions. In *Rob. 2014*, 2014.
- [ODAA13] Luis Oliveira, Carmelo Di Franco, Luis Almeida e Traian E. Abrudan. Fusing Time-of-Flight and Received Signal Strength for Adaptive Radio-Frequency Ranging. In *2013 IEEE Int. Conf. Adv. Robot.*, 2013.
- [Oks14] Irfan Oksar. A Bluetooth signal strength based indoor localization method, 2014. URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6837678>.
- [OLAA14] Luis Oliveira, Hongbin Li, Luis Almeida e Traian E. Abrudan. RSSI-based relative localisation for mobile robots. *Ad Hoc Networks*, 13:321–335, February 2014. URL: <http://www.sciencedirect.com/science/article/pii/S1570870513001741>.
- [RHYHHW⁺08] Wu Rong-Hou, Lee Yang-Han, Tseng Hsien-Wei, Jan Yih-Guang e Chuang Ming-Hsueh. Study of characteristics of RSSI signal. In *2008 IEEE Int. Conf. Ind. Technol.*, pages 1–3. IEEE, April 2008. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4608603>.

REFERÊNCIAS

- [TgStMHj10] Frede Aakmann Tøgersen, Flemming Skjøth, Lene Munksgaard e Søren Højsgaard. Wireless indoor tracking network based on Kalman filters with an application to monitoring dairy cows. *Comput. Electron. Agric.*, 72(2):119–126, July 2010. URL: <http://www.sciencedirect.com/science/article/pii/S0168169910000748>.
- [Yip05] Jason Yipin Ye. *Atlantis: Location Based Services with Bluetooth*. PhD thesis, Brown University, 2005. URL: <https://cs.brown.edu/research/pubs/theses/ugrad/2005/jye.pdf>.
- [Zhu12] Improving crowd-sourced Wi-Fi localization systems using Bluetooth beacons. pages 290–298. IEEE, June 2012. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6275790>.
- [ZZZZ08] Giovanni Zanca, Francesco Zorzi, Andrea Zanella e Michele Zorzi. Experimental comparison of RSSI-based localization algorithms for indoor wireless sensor networks. In *Proc. Work. Real-world Wirel. Sens. networks - REALWSN '08*, page 1, New York, New York, USA, April 2008. ACM Press. URL: <http://dl.acm.org/citation.cfm?id=1435473.1435475>.

REFERÊNCIAS

Anexo A

Artigo - Agricultural Wireless Sensor Mapping for Robot Localization

Artigo desenvolvido em simultâneo com esta dissertação e apresentado na conferência Robot 2015 ¹ ².

¹<https://web.fe.up.pt/~robot2015/>.

²A publicação final está disponível na Springer via http://dx.doi.org/10.1007/978-3-319-27146-0_28.

Agricultural Wireless Sensor Mapping for Robot Localization

Marcos Duarte¹, Filipe Neves dos Santos¹, Armando Sousa¹, Raul Morais²

¹ INESC TEC - INESC Technology and Science (formerly INESC Porto) and Faculty of Engineering, University of Porto, Porto, Portugal,

`ei10007@fe.up.pt, fbsantos@inesctec.pt, asousa@fe.up.pt`

² INESC TEC - INESC Technology and Science (formerly INESC Porto) and Universidade de Trás-os-Montes e Alto Douro, UTAD, Vila Real, Portugal
`rmorais@utad.pt`

Abstract. Crop monitoring and harvesting by ground robots in steep slope vineyards is an intrinsically complex challenge, due to two main reasons: harsh conditions of the terrain and reduced time availability and unstable localization accuracy of the Global Positioning System (GPS). In this paper the use of agricultural wireless sensors as artificial landmarks for robot localization is explored. The Received Signal Strength Indication (RSSI), of Bluetooth (BT) based sensors/technology, has been characterized for distance estimation. Based on this characterization, a mapping procedure based on Histogram Mapping concept was evaluated. The results allow us to conclude that agricultural wireless sensors can be used to support the robot localization procedures in critical moments (GPS blockage) and to create redundant localization information.

1 Introduction

Crop monitoring and harvesting by robots remains a complex challenge, particularly due its low efficiency, accuracy, and robustness on sensing, perception and interpretation of the agricultural environment [1]. The strategic European research agenda for robotics [2] states that robots can improve agriculture efficiency and competitiveness. But there are still very few available commercial robots for agricultural applications [1].

In Europe space we can identify two ongoing research and development projects to deploy monitoring robots on flat vineyards: the VineRobot [3], and Vinbot [4]. However, there are other kinds of vineyards that are not built on flat terrains but on steep slope hills which is a complex environment for the machinery and for the robotic algorithms (such as localization, mapping and path planning). These called steep slope vineyards exist in Portugal in the Douro region - an UNESCO heritage place - Fig. 1, and in other regions of five European countries. As these crops are not built on flat terrain but in steep hills, the development of robotic localization and mapping module that is accurate, reliable and all-time-available is a challenge because [5]:



Fig. 1. A typical steep slope Terraced Vineyard in the Douro region of Portugal.

- the GPS availability and accuracy are largely reduced due to the signal blockage or multi-reflection;
- the dead-reckoning systems (for example odometry and inertial measurement systems) accuracy is drastically reduced due to the harsh conditions of the terrain.

In [5], a localization system based on natural vineyard features, such as the trunks and poles of the vine, is proposed. This system can be seen as an alternative/complement to GPS-based localization systems. However, localization systems based on natural feature detectors are not all the time reliable, because the detector accuracy depends largely on the environment conditions (light, fog and dust) and for that reason the use of artificial beacons is advisable to increase the accuracy, reliability and availability of these localization systems.

The latest techniques for precision agriculture are applying static sensors to the fields for continuously monitoring important variables, such as temperature, humidity and conductivity. The use of wires to connect these sensors to the base station is not always easy or an option and for that reason wireless sensors are a common approach. These sensors inserted in Wireless Sensor Network (WSN) can be used as artificial beacons and they have an enormous potential to increase the accuracy of the robot location estimation.

Various international standards have been established for WSN applications in the past decades. Among them, the standards for wireless LAN IEEE802.11b (“WiFi”), IEEE 802.15.1 (BT) and IEEE 802.15.4 (ZigBee) are the most commonly used [6]. In this work, BT version 4.0 also known as Bluetooth Smart or Bluetooth Low Energy (BLE) based technologies were selected as it provides some features like [7]: “Ultra-low peak, average and idle mode power consumption; ability to run for years on standard coin-cell batteries; lower implementation costs; multi-vendor interoperability; enhanced range”. Particularly, the beacons manufactured by Kontakt³ were used to explore the wireless sensor signals in

³ Available in <http://www.kontakt.io> .

order to improve and increase redundancy of robot localization estimation. In the tests, the robot used was AGROB V14 system defined in [5].

In this paper, in section 2 a global overview about localizing and mapping considering RSSI-based distance estimations is given. In section 3 the RSSI-based distance estimation using *iBeacon* artifacts is evaluated. In section 4 our beacons mapping procedure (BMP) is described. In section 5 the realized tests are detailed and the obtained results are presented. Paper conclusions are presented in section 6.

2 Localization and mapping based on RSSI

Localization can be realized considering several techniques, among them lateration and angulation are the most common techniques. However, a distance and/or an angle measurement between the robot and beacons is required for the robot location to be observable. In Radio Frequency based beacons it is possible to estimate the distance and angle between a beacon and a receiver, by considering three solutions [8]:

- Received Signal Strength (RSS) to distance (path loss models)
- RSS to Angle of Arrival (AoA) (directional antenna models)
- Time-of-Flight to distance (ToF) (Speed of light)

RSS is defined as the voltage measured by a receivers RSSI circuit. Most of the time, RSS is equivalently reported as measured power (the squared magnitude of the signal strength). RSS can be used for acoustic, Radio Frequency (RF), or other signals. Agricultural wireless sensors naturally communicate with neighboring sensors/base-stations, so the RSS of RF signals can be measured by each receiver during normal data communication without presenting additional bandwidth or energy requirements. RSSI is available in a large percentage of commercial receivers, these measurements are relatively inexpensive and simple to implement in hardware.

As the GPS signals are not available in indoor scenarios, the RSS based localization has become an important and popular topic of localization research [9]. However, RSS measurements are notoriously unpredictable. When we consider this RSS measurements for a robust localization system, their sources of error must be precisely modeled. In optimal conditions (open-spaces and non-noisy) the signal power decays proportional to d^{-2} , where d is the distance between the transmitter and receiver. In real conditions, the signal reflection is the most common source of errors. Reflection generates multiple signals with different amplitudes and phases at the receiver antenna, and these signals add constructively or destructively as a function of the frequency, causing frequency-selective fading, in [8] several methods are presented to minimize these errors and increase the distance estimation accuracy.

After the best estimation of the distance between the receiver and beacon, the robot/receiver location can be estimated or corrected. However, for 2D localization purposes, the number of measurements must be enough (at least 3) and

the beacons position well known. Nevertheless most of times these agricultural wireless sensors are installed without being georeferenciated, which is a requirement for this distance to be useful for robot location estimation. Here emerges a question: *how can a robotic mapping procedure for artificial beacons (agricultural wireless sensors) be developed?*

The process of mapping beacons can be solved considering the robot location known or unknown. When it is unknown we must formalize a mapping procedure considering Simultaneous Localization and Mapping (SLAM) based techniques. In [10], a FastSLAM based algorithm is used for mapping seventy Radio-Frequency IDentification (RFID) tags, which are spread through an indoor environment. After the RFID tags mapping stage they have evaluated the robot location considering the RFID distance measurements and ignoring the odometry information, with this work the authors have concluded that the robot was able to localize itself with the same magnitude of accuracy and reduce the computational demands for the global localization of a moving mobile robot.

Despite the fact of SLAM based techniques being the most appropriated for situations when the environment and robot location are unknown, in outdoor robots and particularly in agricultural robots, we can consider that the robot location is always know. In this particular work, of applying robots to steep slope vineyard, we will consider that the robot position is always known during the beacons mapping procedure. This can be considered true, because the mapping procedure will occur only when GPS signals are available or the robot location is observable by other means with an acceptable accuracy.

Knowing the robot position, to be possible to estimate the beacon position in a 2D space, at least 3 distance measurements from different observations points are required. This beacon location estimation is obtained from the circle interception points/spaces. However:

- **To the distance measurements is always associated an uncertainty** which makes the circle boundaries less defined and instead of a single interception point one or several likelihood interception regions are obtained; and,
- **Only one receiver will be available**, which means that at each instant only one distance measurement will be obtained per beacon.

The number of beacon distance measurements can be solved by considering distinct moments and robot locations. With minimum distance measurements satisfied, the beacon location can be estimated. However as several uncertainties are associated to the distance measurements and robot position, the used circle/ellipsoid equation interception solvers are intractable. Instead of finding a single interception point our problem now is to find the most likely interception region. In this way three filters used in robot localization problem can be considered for the beacons mapping procedure [11]:

- Kalman filter - continuous tracking representation;
- Histogram filters - Grid based approaches; and,
- Particle filters;

The Histogram based filters makes a discretization of the 2/3D space into a set of cells and assign to each cell a value with the probability of the robot/object/beacon being localized in that cell. In contrast particle filters spread randomly a large number of particles through the space and to each particle is associated a probability value of that particle being closer to the robot/object/beacon, in each iteration these probability values are updated and those particle with the lowest probability are deleted and new ones are generated in the neighborhood of the most likelihood particles. Due to the non-linearity of the beacons mapping problem, Histogram filters and Particle filters are the most adequate for this problem.

3 iBeacons RSSI based distance evaluation

The BT based sensors called *iBeacons*⁴ are simple emitters that advertises a packet with a certain frequency. They are developed using BT version 4.0 or “Bluetooth Low Energy” (BLE) that specifies the use of Generic Attribute Profile (GATT) built on top of the Attribute Protocol (ATT)[12]. This defines common operations and a framework for use of stored and transported data. GATT has two types of roles:

- server - stores the data transported over ATT and accepts requests, commands and GATT client confirmations;
- client - sends requests, commands and confirmations trough ATT to GATT servers.

Thus, GATT is divided into Attributes, which in turn contain Services with certain Characteristics. These have a Value and a Description. Among others [13], some key features provided by these *iBeacons* are:

- “TX Power Level” - sets the transmission signal strength based on which GATT clients will get RSSI;
- “Advertising Interval” - sets the period advertising packets are emitted.

To characterize the RSSI’s and distances of the *iBeacons* some outdoor tests have been made based on the description of Android Beacon Library[14]. On an empty football field, one put two wooden benches with approximately 0.7m high and separate from each other for needed and known distances of 0.25, 0.5, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 16, 18, 20 and 25m. Then put the *iBeacon* (GATT servers) in a bench and in the other an Android mobile device with an application (GATT clients) that measures the RSSI from the *iBeacon*.

In this particular case, the beacons used were the ones from Kontakt and the mobile device a Samsung GT-I9505. The beacons were set to transmit the advertising packet⁵ each 350 ms and its TxPower to a moderate value⁶ of -12 dBm.

⁴ Devices that complies to certain adopted specifications created by Apple Inc.. See <https://developer.apple.com/ibeacon/> .

⁵ Advertising interval can be set to a value between 20 ms and 10,240 ms.[13]

⁶ Transmission power can be set to -30 , -20 , -16 , -12 , -8 , -4 , 0 or 4 dBm.[13]

On the mobile side, the application developed records the RSSI readings also in intervals of 350 ms during a period of 2 minutes. That takes to 342 samples of RSSI that are saved to a text file to future analysis. In Fig. 2 are examples of histograms based on data recorded by the application ⁷.

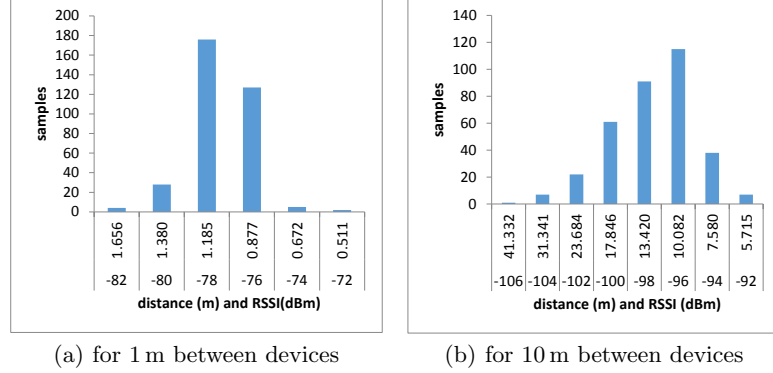


Fig. 2. Example histograms of recorded RSSI samples.

Accordingly to the model described in the Android Beacon Library [14]

$$d = A * \left(\frac{RSSI}{RefPower} \right)^B + C . \quad (1)$$

In 1, d is the distance in meters that we want to obtain, $RSSI$ is the one received from the *iBeacon* by the mobile application, $RefPower$ is the reference received RSSI for the configured TxPower at the distance of 1 m between devices⁸, A , B and C are constants.

With that data, each distance has a corresponding RSSI obtained with a mean value of the 342 samples measured in our field tests. Then, a fitting curve is constructed and a function for converting RSSI to distance is derived using 1 and the nonlinear least squares method to the pairs of distance and mean RSSI.

The function derived:

$$d = 0.3534536 * \left(\frac{RSSI}{RefPower} \right)^{14.8393466} + 0.7566785 . \quad (2)$$

Using 2, a graph of the distances corresponding to the mean RSSI for each distance can be constructed (Fig. 3). Reading the generated graph, one can see that the error on converting from RSSI to distance increases with the distance.

⁷ To improve readability the values of measured RSSI were merged using a interval of 2 dBm instead of 1 dBm.

⁸ RefPower can be extracted from the advertising packet or can be measured for calibration of an *iBeacon* with the specific mobile device. The calibration can give more accuracy to the function of distance estimation.

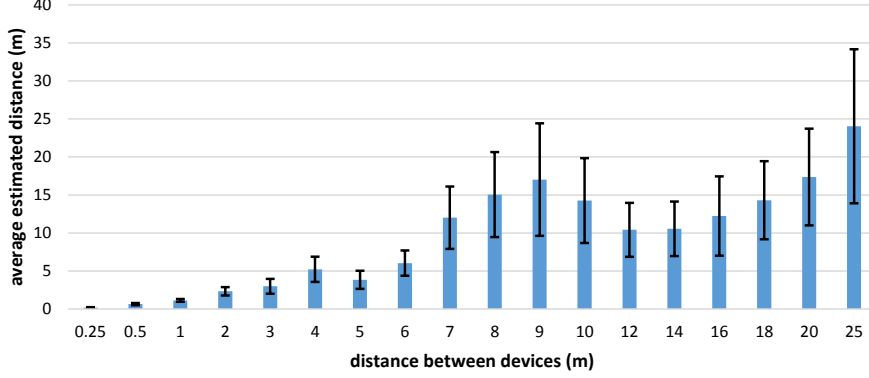


Fig. 3. Estimated distances based on the RSSI.

4 Beacons Mapping Procedure

Due to their simplicity and accuracy, a grid based filter was selected for our BMP. An occupancy grid map is a classical metric map representation [11] and is based on the idea that occupancy grids represent the map as a field of random variables, arranged in an evenly spaced grid. They assign to each (x, y) (or even in 3D (x, y, z)) coordinate a occupancy value which specifies whether or not a location is occupied with an object.

Our BMP algorithm at the first beacon distance observation, generates a likelihood grid map (Fig. 4) with its center located at the robot position with the next constrain:

$$Scale * NumberCells = max_{BeaconDistance} . \quad (3)$$

where $max_{BeaconDistance}$ is maximum observable beacon distance, $Scale$ the scale value for longitude and latitude axis, and $NumberCells$ the horizontal and vertical resolution.

The gold standard of any occupancy grid mapping algorithm is to calculate the posterior over maps given the data:

$$p(m|z_{1:t}x_{1:t}) . \quad (4)$$

where m is the map, $z_{1:t}$ the set of all measurements up to time t , and $x_{1:t}$ is the set of robot poses from time 1 to t .

Based on these concepts and considering that the errors associated to beacons measurements and robot localization follows a normal distribution it is time to formalize a cell probability value as a function of distance observations and robot position estimation. So from a beacon distance observation z_t^{beacon} at time t - in the BMP algorithm - the probability of existing a beacon at the cell $c(x, y)$ is given by:

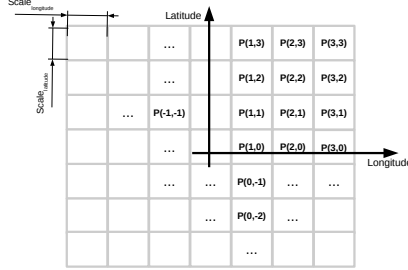


Fig. 4. Likelihood grid map for beacon existing probability region description.

$$P_t(x, y) = \frac{1}{\sigma_{RB}\sqrt{2\pi}} e^{-\frac{(d_{RB} - z_t^{beacon})^2}{2\sigma_{RB}^2}} . \quad (5)$$

where, d_{RB} is the distance between the robot location and the cell gridmap center, and σ_{RB} is the sum of uncertainties associated to the robot position estimation and beacon distance estimation.

For the fusion of several distance measurements at different time instants, the probability value at each grid map cell $P(x, y)$ is given by:

$$P(x, y) = \frac{P_t(x, y) * P_{t-1}(x, y) * P_{t-2}(x, y) * \dots}{\sum_{t=0}^T \sum_{i=0}^X \sum_{j=0}^Y P_t(i, j)} . \quad (6)$$

Based on these probabilities values, the beacon position estimation is simply given by:

$$B_{x,y} = \sum_{i=0}^X \sum_{j=0}^Y C_{i,j} * P(i, j) . \quad (7)$$

where $B_{x,y}$ is bi-dimensional vector that represents the beacon position and $C_{i,j}$ is bi-dimensional vector that represents the cell position center.

5 Tests and results

For BMP algorithm test we have selected the AGROB V14 platform. AGROB V14 is small and cost effective outdoor robot for application on steep slope vineyard monitoring tasks. The robot is built on top of a commercial radio-controlled model (RC-model) Traxxas E-Maxx, a 1/10 scale 4WD Electric Monster Truck, Fig. 5. To this RC-mode are attached two tiny computers (one UDOO quadcore version, and one RaspberryPi version B), with Advanced RISC Machine (ARM) processors. This robot has two processing units running Linux (Ubuntu 12.04) with the robotic operating system (ROS-groovy) on top. More details about AGROB V14 can be found in [5].

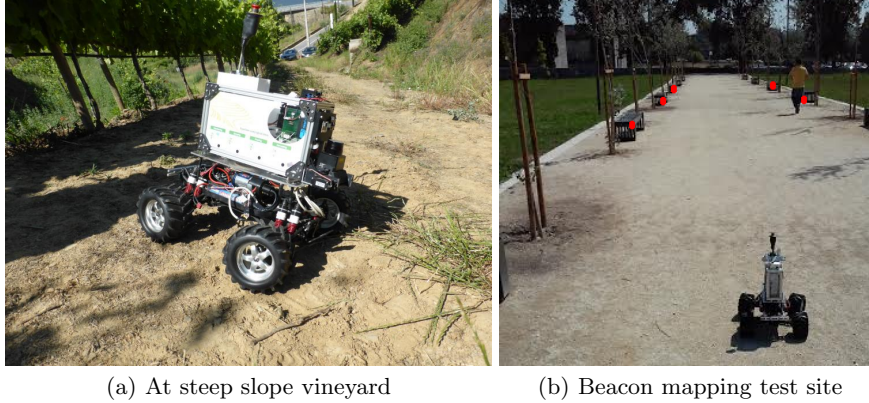


Fig. 5. AGROB V14 robot at the vineyard and test site. Red dots are the places where the beacons were deployed.

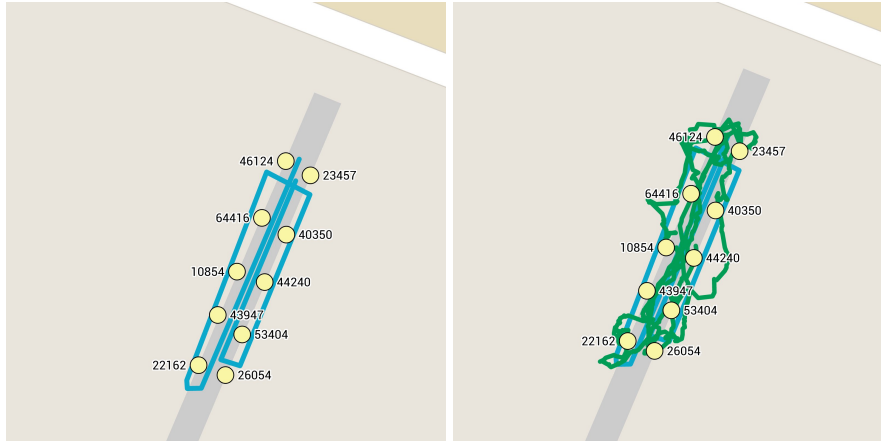
To make the robot able to acquire distance measurements to these beacons, an Android application was developed and deployed into a smartphone (Samsung GT-I9505), which was attached to the robot. The Android application running in the mobile device starts reading the RSSI of every visible beacon and sends the real-time information every 1 s via User Datagram Protocol (UDP) to the AGROB V14 robot (via the wireless interface).

In Fig. 5 is shown in the left the AGROB V14 operating in a steep slope vineyard, and in the right the robot at the beacon mapping test site. As shown in Fig. 5 and in 6, ten *iBeacons* were deployed over ten park benches, which are spaced by 12 meters longitudinally and 6 meters transversely. To illustrate their positions, the minor ID of each *iBeacon* has been used. Robot estimated trajectories and beacons position estimation were drawn using Google Maps API.

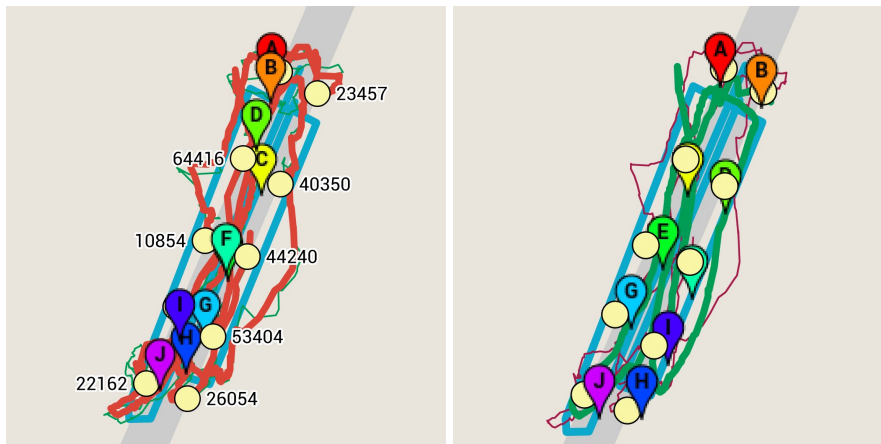
The data acquired by the robot at the test site has been recorded and stored in a ROSbag file ⁹. Using this file two tests were realized. On the first one, the robot location estimation was obtained from the fusion of GPS and odometry velocity, by means of a simple particle filter. In Fig. 6.(a) is shown the resulted estimated robot trajectory (Red bold line) and beacons position (colorful pinpoints). From this figure is possible to verify a better estimation of the robot trajectory (in contrast to GPS estimation (Green thin line)).

On the second, the robot location estimation was obtained from the fusion of GPS, odometry velocity and considering the mapped beacons and beacons distance observations, by means of a simple particle filter. In Fig. 6.(b) is shown the estimated robot trajectory (Green bold line) and beacons position (colorful pinpoints). In contrast to the first test, from this figure is possible to verify a better estimation of the robot trajectory and better beacons position estima-

⁹ The file can be accessed at <http://hyselam.com/agrob.html>



(a) Yellow dots are the *iBeacons* and blue (b) Green line is the robot trajectory ex- line the planned trajectory to be realized tracted from the GPS receiver estimations. by the robot.



(c) Results of the first test

(d) Results of the second test

Fig. 6. The obtained test results.

tion. When the robot position estimation was realized considering the beacons distance observations (and their beacons position estimation) has improved significantly the robot trajectory estimation and consequently the beacons position estimation, as can be seen in 6.

Fig. 7 shows the obtained likelihood grid map for the *iBeacon* with minor ID 64416 at three different instants. These likelihood grid maps were obtained during the second test and it is possible to verify the evolution of the beacon

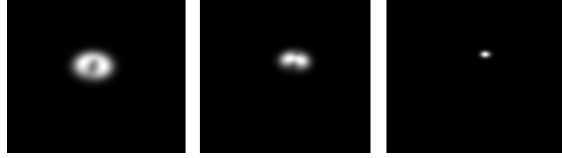


Fig. 7. Grid map with resolution of 0.25 meters/pixel for *iBeacon* with minor ID 64416 at 10, 230, 460 seconds instant.

position estimation uncertainty. The possible beacon position has evolved from a circular region, to two possible regions and ends in likelihood single spot.

On the second test, the beacon position estimations has in average an error of 1.5meters. The main origin of this error is associated to the low accuracy of robot position estimation, which is drifted from the original trajectory as can be seen in Fig. 6.(b). However, if we analyze the relative error positioning of the beacons (distance between beacons) the average error is 0.43meters, near to the likelihood grid-map resolution, which is 0.25meters/pixel. This means that beacon mapping accuracy is highly dependent on the robot position estimation accuracy, as expected.

6 Conclusion

In this work, was shown that the use of agricultural wireless sensor signals can help to improve and increase redundancy of robot location estimation, even when beacons mapping procedure is occurring, as show in section 5. Of course there is the need of doing more and repeated tests to have enough data to help the support of these results.

Besides that the BMP algorithm is easily implementable and integrable to other localization and mapping procedures. As future work we will integrate agricultural wireless sensors and BMP into the hybrid-map based localization system proposed in [5], in order to evaluate the system localization robustness in situations of GPS absence.

Acknowledgment

This work is financed by the ERDF European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project FCOMP-01-0124-FEDER-037281

References

1. Bac, C. W., van Henten, E. J., Hemming, J. and Edan, Y., "Harvesting Robots for High-value Crops: State-of-the-art Review and Challenges Ahead.", *Journal of Field Robotics* 31.6 (2014): 888-911.
2. euRobotics (2013). Strategic research agenda for robotics in Europe. Draft 0v42 [Online]. Available: http://ec.europa.eu/research/industrial_technologies/pdf/robotics-ppp-roadmap_en.pdf
3. "VineRobot - FP7 project." [Online]. Available: <http://www.vinerobot.eu/>
4. "Vinbot - FP7 project." [Online]. Available: <http://vinbot.eu/?lang=pt>
5. Neves Dos Santos, F., Sobreira, H., Campos, D., Morais, R., Moreira, A.P., Contente, O., "Towards a Reliable Monitoring Robot for Mountain Vineyards" Autonomous Robot Systems and Competitions (ICARSC), 2015 IEEE International Conference, 2015, April: 37-43
6. Wang, Ning. "13 Worksite Management for Precision Agricultural." *Agricultural Automation: Fundamentals and Practices* (2013): 343.
7. Bluetooth. "The Low Energy Technology Behind Bluetooth Smart." [Online]. Available: <http://www.bluetooth.com/Pages/low-energy-tech-info.aspx>.
8. Patwari, N., Ash, J. N., Kyperountas, S., Hero, A. O., Moses, R. L., Correal, N. S., "Locating the nodes: cooperative localization in wireless sensor networks." *Signal Processing Magazine, IEEE* 22.4 (2005): 54-69.
9. Honkavirta, V., Perala, T., Ali-Loytty, S., Piche, R., "A comparative survey of WLAN location fingerprinting methods." *Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop on. IEEE, 2009.*
10. Hahnel, D., Burgard, W., Fox, D., Fishkin, K., Philipose, M., "Mapping and localization with RFID technology". In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on* (Vol. 1, pp. 1015-1020). IEEE.
11. Thrun, Sebastian, Wolfram Burgard, and Dieter Fox. "Probabilistic robotics". MIT press, 2005.
12. Bluetooth SIG. "Specification of the Bluetooth System." (2010): 201-203. [Online]. Available: https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=229737 .
13. Kontakt. "Kontakt.io Beacon Datasheet v2.0." (2014). [Online]. Available: <http://docs.kontakt.io/beacon/kontakt-beacon-v2.pdf> .
14. "Android Beacon Library - Distance Estimates." [Online]. Available: <http://altbeacon.github.io/android-beacon-library/distance-calculations.html> .

