

Mestrado em Engenharia Informática  
Dissertação/Estágio  
Relatório Final

# Sistema multimodal de reconhecimento de linguagem gestual para o Google Earth

Nuno Miguel Mendes Marques  
nmiguel@student.dei.uc.pt

Orientador do DEI:  
Carlos Lisboa Bento

Co-Orientador (DEC):  
Nuno Norte Pinto

Data: 3 Julho de 2013



**FCTUC** DEPARTAMENTO  
**DE ENGENHARIA INFORMÁTICA**  
FACULDADE DE CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE DE COIMBRA

## **Resumo**

Actualmente, são inúmeros os dispositivos utilizados, que dão uma ênfase de comodidade e conforto à nossa vida, características dos nossos tempos modernos. Esses dispositivos variam, desde sensores especializados até uma simples chave de um carro com fecho centralizado de portas.

No âmbito dos Sistemas Móveis e Ubíquos, existe uma preocupação evidente com a melhoria do conforto na realização de tarefas quotidianas. Com base nesse princípio, esta dissertação foca-se no controlo de aplicações informáticas através do uso de sensores, neste caso, é utilizado o *Microsoft Kinect* como dispositivo de interacção com o *Google Earth*.

O *Microsoft Kinect* é constituído por câmeras de vídeo e de profundidade, assim como, sensores de infravermelhos e de áudio. É vastamente utilizado em vários tipos de aplicações, mas com um enfoque especial no foro dos jogos de vídeo.

Para a sua utilização, como dispositivo de interacção do utilizador com o *Google Earth*, foram desenvolvidas e melhoradas as primitivas gestuais do *Microsoft Kinect*.

Como se trata de um equipamento bastante sensível, foi também utilizado um *smartphone* para auxiliar a interacção do utilizador. O *Microsoft Kinect* é responsável por reconhecer gestos feitos pela mão do utilizador e usar esses gestos como forma de interacção com o *Google Earth*. Por sua vez, o *smartphone*, é responsável pela vertente de interacção, relativa ao posicionamento e localização no modo de imersão do *Google Earth*.

Esta dissertação apresenta-se como um desafio, na medida em que, ambiciona uma interligação natural entre utilizador e sensor, de forma a proporcionar um efeito de imersão do utilizador na navegação da aplicação informática.

## **Palavras-Chave**

Google Earth, Microsoft Kinect, Primitivas gestuais, Planeamento do Território, Sistemas Móveis e Ubíquos, Smartphone, Magnetómetro

## **Agradecimentos**

Gostaria de agradecer a todas as pessoas que me apoiaram durante a elaboração deste projecto;

Ao meu orientador, Carlos Lisboa Bento, por todas as sugestões indicadas, pelo tempo dispendido em reuniões, por me apoiar em decisões, apontar novas perspectivas e zelar pelo sucesso deste projecto.

Ao meu co-orientador Nuno Norte Pinto, do Departamento de Engenharia Civil, por ter permitido a utilização do *Microsoft Kinect*, propriedade do seu Departamento, ao AmILab, sem o qual teria sido impossível realizar este projecto. Também por toda a paciência, motivação, incentivo e comentários que dele recebi.

Membro do CISUC e do Departamento de Matemática, Ana Maria de Almeida, pela sua amabilidade em me ter facultado um *smartphone* com magnetómetro para efeitos de desenvolvimento e apresentação.

Aos utilizadores de teste da aplicação, pois contribuíram para o melhoramento gradual deste projecto.

Aos meus familiares mais próximos, por me terem apoiado sempre e não me deixarem desmotivar em alturas mais complicadas.

Ao meu círculo de amigos, que tanto me apoiou para a concentração no trabalho como me ajudou a não me tornar obsessivo.

Agradeço também a Deus, que sempre me iluminou durante este percurso académico e me deu toda a capacidade de raciocínio.

# Índice

Capítulo 1 Introdução .....	1
1.1. Enquadramento .....	1
1.2. Motivação.....	1
1.3. Estrutura do relatório.....	2
Capítulo 2 Estado da Arte .....	4
2.1. Sistemas de Interacção.....	4
2.2. <i>Microsoft Kinect</i> .....	4
2.2.1. Funcionamento .....	4
2.2.2. Potencialidades.....	5
2.2.3. Limitações .....	5
2.2.4. <i>Finger recognition</i> .....	6
2.3. Outros dispositivos .....	6
2.3.1. <i>Asus Xtion Pro Live</i> .....	6
2.3.2. <i>Natural Point - TrackIR</i> .....	7
2.3.3. <i>Natural Point – SmartNav</i> .....	8
2.3.4. <i>Natural Point – OptiTrack</i> .....	8
2.3.5. <i>Eye Tracking</i> .....	9
2.3.6. <i>LEAP Motion</i> .....	9
2.4. Magnetómetros e <i>Smartphones</i> .....	10
2.5. Análise Comparativa .....	11
2.5.1. <i>Asus Xtion Pro Live</i> .....	11
2.5.2. <i>Natural Point – OptiTrack e TrackIR</i> .....	11
2.5.3. <i>LEAP Motion</i> .....	12
Capítulo 3 Abordagem Adoptada e Planeamento.....	13
3.1. Objectivos .....	13
3.2. Casos de uso .....	13
3.2.1. <i>Microsoft Kinect</i> .....	13
3.2.2. <i>Smartphone</i> .....	13
3.2.3. Aplicação principal .....	14
3.3. Requisitos .....	14
3.4. Metodologia.....	14

3.4.1. <i>Microsoft Kinect</i> .....	15
3.4.2. <i>Smartphone</i> .....	15
3.4.3. Aplicação principal .....	15
3.5. Abordagem .....	16
3.5.1. Arquitectura .....	16
3.5.2. Validação .....	17
3.5.3. Testes e planeamento .....	17
3.6. Estudos Preliminares.....	18
3.6.1. Trabalho Base .....	18
3.6.2. Resultados .....	18
3.6.3. Interpretação dos resultados obtidos .....	20
3.7. Plano de Trabalho e Implicações .....	20
3.7.1. Primeiro semestre .....	20
3.7.2. Segundo semestre .....	22
3.8. Resumo.....	22
Capítulo 4 Análise Complementar e Implementação.....	24
4.1. Resumo.....	24
4.1.1 Smartphone.....	24
4.1.2. Microsoft Kinect.....	25
4.1.3. Aplicação Principal .....	26
4.2. Implementação.....	27
4.2.1. Smartphone.....	27
4.2.2. Microsoft Kinect.....	31
4.2.3. Aplicação Principal.....	35
Capítulo 5 Testes de Usabilidade.....	42
5.1. Descrição dos Testes.....	42
5.1.1. Tarefas .....	42
5.1.2. Experiência dos Utilizadores .....	43
5.1.3. Tempo de Execução.....	43
5.1.4. Níveis de Dificuldade.....	44
5.1.5. Comentários.....	44
5.2 Conclusão dos Testes.....	45
Capítulo 6 Conclusões.....	46
6.1. Principais Obstáculos.....	46

6.2. Trabalho Futuro.....	47
Referências.....	48

## Lista de Figuras

Figura 1	<i>Microsoft Kinect</i> em cima. <i>Asus Xtion Pro Live</i> em baixo.....	7
Figura 2	<i>TrackIR 5</i> .....	8
Figura 3	<i>SmartNav</i> .....	8
Figura 4	<i>Optitrack Arena</i> .....	9
Figura 5	<i>Eye Tracking</i> .....	9
Figura 6	<i>LEAP Motion</i> .....	10
Figura 7	Caso de uso do <i>Microsoft Kinect</i> .....	13
Figura 8	Caso de uso do Smartphone.....	13
Figura 9	Caso de uso da aplicação principal.....	14
Figura 10	Arquitectura do projecto.....	16
Figura 11	Teste aos frames por segundo e à resolução do <i>Microsoft Kinect</i> .....	19
Figura 12	Teste à velocidade do USB do <i>Microsoft Kinect</i> .....	19
Figura 13	Diagrama de <i>Gantt</i> do planeamento inicial do primeiro semestre (parte 1/2)..	20
Figura 14	Diagrama de <i>Gantt</i> do planeamento inicial do primeiro semestre (parte 2/2)..	21
Figura 15	Diagrama de <i>Gantt</i> do planeamento final do primeiro semestre (parte 1/2)....	21
Figura 16	Diagrama de <i>Gantt</i> do planeamento final do primeiro semestre (parte 2/2)....	21
Figura 17	Diagrama de <i>Gantt</i> do planeamento inicial do 2º semestre (parte 1/2).....	22
Figura 18	Diagrama de <i>Gantt</i> do planeamento inicial do 2º semestre (parte 2/2).....	22
Figura 19	<i>Rating</i> do computador fixo.....	25
Figura 20	<i>Rating</i> do computador portátil.....	25
Figura 21	Menu inicial do <i>smartphone</i> .....	27
Figura 22	Exemplos de <i>handling</i> de erros no endereço IP.....	28
Figura 23	<i>Smartphone</i> em modo de localização.....	29
Figura 24	<i>Smartphone</i> em modo de posição.....	29
Figura 25	1ª parte do Diagrama de Fluxo.....	29
Figura 26	2ª parte do Diagrama de Fluxo de Dados do <i>Smartphone</i> .....	30
Figura 27	Primeira fase do reconhecimento.....	31
Figura 28	Segunda fase do reconhecimento.....	31
Figura 29	Reconhecimento final da mão e pontas dos dedos.....	32
Figura 30	Exemplo de reconhecimento das duas mãos.....	33
Figura 31	Diagrama de Fluxos de Dados do <i>Microsoft Kinect</i> .....	34

Figura 32	Calibração na aplicação.....	35
Figura 33	Conversão das coordenadas para o posicionamento.....	36
Figura 34	Utilização da API do <i>Google Earth</i> para localização.....	36
Figura 35	Aplicação no modo inicial.....	37
Figura 36	Aplicação em modo de imersão.....	38
Figura 37	Modo de imersão desligado e centrada nova posição.....	38
Figura 38	Modo de imersão ligado na nova posição.....	39
Figura 39	1ª parte do Diagrama de Fluxo de Dados da Aplicação Principal.....	40
Figura 40	2ª parte do Diagrama de Fluxo de Dados da Aplicação Principal.....	41
Figura 41	Número mínimo recomendado de utilizadores de teste	42
Figura 42	Gráfico da média de tempos de execução	43
Figura 43	Gráfico da média dos valores da dificuldade encontrada	44

## **Lista de Tabelas**

Tabela 1	Comparação entre o <i>Microsoft Kinect</i> e o <i>Asus Xtion Pro Live</i> .....	11
Tabela 2	Comparação entre dispositivos <i>Natural Point</i> .....	12
Tabela 3	Possíveis factores de risco.....	18

## **Alterações relativas aos comentários da defesa intermédia**

Os comentários atribuídos a este trabalho, foram tomados em consideração e, devido a haver concordância com os mesmos, nesta secção encontra-se a referência às alterações efectuadas.

A contextualização foi melhorada no capítulo da Introdução, no Enquadramento e Motivação.

Relativamente ao capítulo do Estado da Arte, foi removido o conteúdo relativo aos *Google Goggles*, pois com essa tecnologia não é possível atingir os objectivos deste projecto. A escolha do *Microsoft Kinect* em relação ao *Asus Xtion Pro Live*, foi clarificada e mencionada na página 11, relativamente à comparação entre os dois dispositivos.

O subcapítulo 2.4, que menciona o *smartphone* e o magnetómetro, foi aprofundado, assim como a informação sobre o funcionamento e interligação dos dois dispositivos no capítulo 4, Análise Complementar e Implementação.

A arquitectura do sistema foi totalmente revista e encontra-se no subcapítulo 3.5.1, página 16.



# Capítulo 1

## Introdução

### 1.1. Enquadramento

A ideia deste projecto teve origem no Departamento de Engenharia Civil, em que é bastante importante efectuar vários estudos de terreno e localização, nomeadamente para a cadeira de Planeamento do Território, entre outras.

Este projecto apresenta-se também como um desafio na Área dos Sistemas Ubíquos, no aspecto de se desejar cada vez mais um controlo livre das aplicações, sem ser necessário recorrer ao manuseamento de equipamento extra para essa finalidade.

Uma parte do conteúdo deste projecto tem um potencial interesse para aplicações que possam requerer uma maior acessibilidade por parte do utilizador. Poderá ter uma contribuição futura para a ajuda de pessoas que necessitem de utilizar linguagem gestual como meio de comunicação e, assim sendo, utilizariam essa mesma linguagem com aplicações informáticas. Já existem empresas dedicadas a tempo inteiro no desenvolvimento de tecnologia semelhante [19].

Outra parte deste projecto enquadra-se na necessidade da exigência por parte dos utilizadores quotidianos. Cada vez mais se procura um sistema rápido, eficaz e adaptativo. A realidade aumentada e a realidade virtual são tecnologias que despertam por completo o interesse do utilizador [10]. Este, procura o efeito de interagir directamente com o que o rodeia, desejando assim, um efeito de imersão.

### 1.2. Motivação

Ultimamente tem havido uma grande necessidade de se querer dar o controlo ao utilizador, de uma forma inovadora. Através de comandos sem fios, comandos com sensores de movimento, entre outros. Mas neste preciso momento, estamos na fase em que o melhor comando que um utilizador pode ter, são os seus próprios membros.

Tem havido um grande investimento e crescimento tecnológico na perspectiva do conforto do utilizador. A *Microsoft*, através do *Microsoft Kinect* e da nova *Xbox*, a *Engadget* com o *LEAP Motion* e outras empresas que procuram o desenvolvimento destes e de outros dispositivos idênticos.

Também a empresa *Human Spot*, em Coimbra, que incide na área da realidade aumentada, demonstrou algum interesse sobre este projecto e espera ver o resultado final.

Tendo tudo isto em conta, podemos rapidamente perceber o interesse, o impacto, que esta investigação poderá ter. A ideia de podermos estar a contribuir para mais um avanço

tecnológico, é muito apelativa e positiva para o desenvolvimento da sociedade e para um maior empreendedorismo nas empresas que as aplicarem.

Com a tecnologia que existe no nosso quotidiano, temos ao nosso dispor inúmeros meios tecnológicos para conhecer melhor territórios e zonas geográficas. Esse conhecimento passa pela observação, análise e ponderação do espaço que pode dar origem a uma viagem virtual, utilizando não só a tecnologia tradicional mas também a tecnologia móvel, como por exemplo, GPS e *smartphones*.

Torna-se assim, muito mais acessível para a geração futura, a capacidade de aprender, compreender, preservar e planear o território [11].

### **1.3. Estrutura do relatório**

Este relatório apresenta uma estrutura normal em que se pretende, neste primeiro capítulo introdutório, situar o leitor na área em que é aplicado este projecto, revelando objectivos e interesses.

No segundo capítulo é dado a conhecer ao leitor, em que ponto se encontra a tecnologia actual no que diz respeito a sistemas de interacção com o utilizador e variados sensores. É também feita uma breve comparação de especificações no final deste capítulo.

No terceiro capítulo, são definidos os objectivos deste projecto e é explicado o que se pretende fazer. É mencionada a metodologia e a abordagem para o *Microsoft Kinect* e para o *smartphone* em separado, aliviando a complexidade visual a nível de esquematização. Neste capítulo, é também revelado ao leitor, que existe um trabalho que funcionará como base neste. São também referidos os possíveis factores de risco e demonstrada a razão porque o deixaram de ser. O planeamento esperado para o primeiro semestre, assim como, o planeamento resultante do mesmo, após alguns desvios, elucida o leitor para o planeamento do segundo semestre, encontrando-se estes, no final do capítulo.

No quarto capítulo encontra-se o complemento da análise feita no capítulo anterior. É referido com maior detalhe, o modo como as componentes do *smartphone* e do *Microsoft Kinect* vão actuar e de que forma se integram na aplicação principal. Também se encontra referida a implementação deste projecto, toda a informação relevante e essencial para ser perceptível a lógica que existe na construção da aplicação se encontra descrita. Está também dividida pelas duas componentes e a sua interligação com a aplicação principal.

No quinto capítulo, são referidos os testes de usabilidade. São explicados e demonstrados os seus resultados. Este é concluído com uma síntese dos resultados obtidos.

No sexto e último capítulo encontra-se a conclusão deste trabalho, dificuldades encontradas na elaboração do mesmo e possibilidade de aplicação em trabalho futuro.

## Capítulo 2

### Estado da Arte

Neste capítulo, serão mencionadas e explicadas as tecnologias que existem a nível do *hardware* que será utilizado para a realização deste trabalho. Serão também referidos produtos semelhantes que estão a ser utilizados neste campo [17].

#### 2.1. Sistemas de Interação

Nos finais de século passado em que os computadores deixaram de ser instrumentos complexos para a maioria das pessoas, começou a caminhar-se a passos largos para jogos interactivos para as variadas consolas de jogos, apreciadas de um modo geral, pela população jovem.

Uma vez que esta tendência de interação obteve sucesso, foi aproveitada ao máximo pelo ensino/aprendizagem nas escolas, apostando na formação dos professores para a introdução dos quadros interactivos nas mesmas. As próprias editoras publicaram manuais interactivos das várias disciplinas, em que o aluno toma parte integrante na execução dos exercícios em plena aula.

Esta interação tem sido desenvolvida simultaneamente com a evolução dos telemóveis. Daí o aparecimento dos ecrãs que suportam a tecnologia táctil. Esta evolução levou à procura quase exponencial de equipamentos como *smartphones Android* e *iPhone*, onde a eficácia se alia à rapidez de obtenção de informação, dando ao utilizador a percepção de domínio completo do mundo que o rodeia.

Com a evolução da Inteligência Artificial, também se verificam novos sistemas interactivos em que a intervenção humana deixa de ser dominante, tendo a própria máquina um papel preponderante sobre aquela. Este caso verifica-se, por exemplo, em alguns sistemas de circuitos de metropolitano.

#### 2.2. *Microsoft Kinect*

##### 2.2.1. Funcionamento

Ao ser efectuado o estudo relativo ao *Microsoft Kinect*, foi analisado o seu modo de captura gestual e devido à sua elevada sensibilidade e baixa precisão, principalmente da câmara de profundidade, compreendeu-se a necessidade dos utilizadores frequentes do *Microsoft Kinect*, fazerem gestos amplos e por vezes, constringedores.

O *Microsoft Kinect* tem a capacidade de efectuar uma representação aproximada do esqueleto humano, transformando as articulações principais do corpo em pontos, denominando-os de *joints* ou junções. Esses pontos, de acordo com a sua disposição espacial, são unidos consoante o algoritmo desenhado pela *Microsoft* e resulta assim na construção do esqueleto, como referido.

O método de captura de gestos é elaborado de modo a que, por exemplo, a mão direita se eleve à altura do ombro direito e se mova até ao ombro esquerdo. Neste caso, o importante são as intersecções sucessivas de *joints* específicas. A *joint* que corresponde à mão direita intersecta com a do ombro direito, com a do pescoço e finalmente com a do ombro esquerdo. Mas este método nem sempre é bem sucedido, visto que, durante a captura dos

gestos, a intersecção das *joints* pode ser causa de falhas de leitura, pois um cotovelo pode ser assumido, por engano, como um ombro, em milissegundos.

Um utilizador de um modo geral, deseja um sistema rápido e fluido, e por vezes, ao efectuar um comando sem sucesso, tem tendência a acelerar o movimento. No caso do *Microsoft Kinect*, principalmente na *Xbox 360*, esse aumento de velocidade converte-se num aumento da probabilidade de erros pois perde-se a posição espacial das *joints* que tinham sido previamente detectadas e também devido às limitações do próprio *hardware*.

### **2.2.2. Potencialidades**

Uma das grandes potencialidades do *Microsoft Kinect* é o reconhecimento de imagem, sendo capaz de perceber aquilo que capta. Através da câmara de cor(RGB) e a de profundidade, juntamente com o emissor de infravermelhos, torna-se possível a detecção de objectos e até mesmo partes do corpo, facilitando assim a obtenção do objectivo pretendido.

Outra potencialidade notável é o facto de o *Microsoft Kinect* conter um *array* de 4 microfones, voltados para baixo, para poder fazer a triangulação da posição espacial do utilizador em relação à posição do dispositivo.

Também é de realçar o acelerómetro contido no dispositivo, que normalmente é utilizado para a medição da força gravítica.

### **2.2.3. Limitações**

O *Microsoft Kinect* para *Xbox 360* encontra-se limitado em vários aspectos. Desde a resolução das câmaras até à velocidade do interface USB.

De modo a não causar uma sobrecarga no sistema da *Xbox 360*, a *Microsoft* reduziu a velocidade do interface USB para cerca de 15~16Mbps. Esta decisão implica que o interface USB não consuma os recursos necessários para o devido funcionamento dos acessórios da própria consola.

Ao ser limitada a velocidade do interface USB, podemos subentender que também não será possível obter-se uma boa resolução por parte das câmaras que compõem o *Microsoft Kinect*. Desta forma, a resolução será de 320x240 pixéis, com uma frame rate de 30 frames por segundo. Se fosse possível aumentar a resolução para 640x480 pixéis, as probabilidades da frame rate não acompanhar, seriam altas [23].

Em Fevereiro de 2012, a *Microsoft* lançou o *Microsoft Kinect* para *Windows*.

Esta nova versão foi concebida com o intuito de ser utilizada maioritariamente para desenvolvimento e não funciona na *Xbox 360*, pois não existe redução de velocidade no interface USB e permite aumentar a resolução das câmaras.

Efectuaram-se testes para verificar a veracidade desta informação [8]. Confirma-se que o USB pode atingir a velocidade de um dispositivo USB 2.0 normal, e a resolução pode ser aumentada com o custo de perda de frames por segundo. A resolução base é de 640x480 pixéis com 30 frames por segundo, se for aumentada para 1280x960 pixéis, os frames por segundo são inferiores a 12, sendo este número meramente indicativo no *software* do *Microsoft Kinect*, pois o sistema operativo *Windows* ficou parado nesse teste.

O sensor de profundidade, como referido anteriormente, tem uma precisão baixa pelo que precisa de estar sempre a uma distância óptima para o seu correcto funcionamento. A câmara de profundidade opera em dois modos, o modo normal e o modo *near*. Este modo é usado para quando o utilizador precisa de estar um pouco mais perto do *Microsoft Kinect* e muda a definição de distância ideal para capturar movimentos.

Foi também testada a distância mínima que esse sensor consegue medir e o resultado obtido foi de 366 milímetros, em modo *near*, apresentando dificuldades.

#### **2.2.4. Finger recognition**

Algumas empresas e outros entusiastas dedicam-se a explorar o *Microsoft Kinect* e a criar aplicações que se ajustam aos seus interesses e/ou necessidades.

Encontram-se várias aplicações disponíveis para download e outras são pagas.

Uma preocupação que se teve em conta, foi o facto de alguns vídeos apresentados poderem ser algo fictícios no que diz respeito à execução das aplicações. Por vezes, parece que a aplicação efectua inúmeras operações sem problemas, mas ao explorá-las com maior cuidado, verifica-se a utilização de *hardware* extra e/ou modificado [9].

A empresa *CogniMem*, utiliza hardware adicional para o funcionamento de aplicações para o *Microsoft Kinect*. Através de chips de redes neuronais, a velocidade de processamento e a precisão do *Microsoft Kinect* são melhorados [3]. Um dos projectos desta empresa, vai de encontro a um dos objectivos de este estágio, que é a detecção do esqueleto da mão e interpretar os gestos feitos com os dedos [4].

Como já referido, a câmara de profundidade tem problemas a nível de precisão, o que torna difícil a distinção entre os vários dedos da mão, principalmente quando os dedos estão próximos uns dos outros, os resultados são maioritariamente errados.

### **2.3. Outros dispositivos**

No mercado actual existem outros dispositivos que fazem concorrência ao *Microsoft Kinect*. Alguns conseguem fazer concorrência directa, sendo capazes de apresentar melhores resultados pois têm uma precisão maior e por vezes utilizam tecnologia diferente.

#### **2.3.1. Asus Xtion Pro Live**

O *Asus Xtion Pro Live* é um concorrente directo do *Microsoft Kinect*, por utilizar a mesma tecnologia base. Consegue atingir uma resolução maior sem uma perda de frames por segundo significativa, como acontece no *Microsoft Kinect*, e tem uma precisão superior na sua câmara de profundidade.

O *Asus Xtion Pro Live* não dispõe de um sistema de *audio* que permita o cálculo da posição do utilizador pelo som, e utiliza um mapeamento das *joints* diferente do *Microsoft Kinect*.

Apesar de existir um *SDK* da *Asus*, muitos dos projectos elaborados utilizam um *SDK open source* chamado *OpenNI*. O *OpenNI* tem vindo a ser uma alternativa tanto para o *Asus Xtion Pro Live* como para o *Microsoft Kinect*. Em alguns casos, apresenta melhores resultados em comparação com os *SDKs* proprietários.

Energeticamente falando, o *Asus Xtion Pro Live* utiliza apenas a ligação USB de 5V, enquanto o *Microsoft Kinect* necessita de estar ligado à corrente para obter os 12V e mais 5V do USB para o seu devido funcionamento [20].



Figura 1: *Microsoft Kinect* em cima. *Asus Xtion Pro Live* em baixo

### **2.3.2. Natural Point - TrackIR**

O *TrackIR* é um produto que permite ao utilizador, com o simples virar ou acenar da cabeça, possa ver no monitor o que está na área correspondente à rotação horizontal ou vertical da cabeça.

Este produto é composto por um sensor de infravermelhos que se coloca no ponto de rotação correspondente aos zero graus, ou seja, de frente para o utilizador. O utilizador, nas primeiras versões deste produto, apenas tinha um indicador que podia ser colado na testa ou num chapéu. Esse indicador foi evoluindo e nestas últimas versões já se trata de um dispositivo mais complexo para melhoramento da precisão e sensibilidade da captura do *TrackIR*.

Como utilizador da primeira versão deste produto, foi possível verificar em primeira mão a sua evolução tanto a nível de *hardware* como de *software*.

Inicialmente, ao virar-se a cabeça, a imagem no monitor virava de acordo com os mesmos graus de rotação que eram feitos, o que por vezes, fazia com que o indicador desaparecesse do alcance do sensor infravermelho, ou os nossos olhos não pudessem olhar para o monitor, no caso de uma rotação de cerca de 40 graus para um dos lados. Esse problema foi resolvido com o reajustamento angular que o *software* pode fazer. Pode ser feita uma rotação emulada, em que 10 graus reais, correspondem a 40 graus no monitor. Isto permite que o utilizador consiga fazer rotações completas, sem ter de virar a cabeça a um nível exagerado [14].

Este dispositivo em comparação com o *Microsoft Kinect*, permite que não seja necessário o reconhecimento de imagem para efectuar operações semelhantes para o efeito de imersão pretendido, como por exemplo, observar o ambiente em redor e até mesmo possibilitar o movimento do utilizador imerso.



Figura 2: *TrackIR 5*

### **2.3.3. *Natural Point – SmartNav***

O *SmartNav* é um produto bastante semelhante ao *TrackIR*, com a particularidade de se poder controlar o rato com movimentos da cabeça. O *software* do produto é bastante configurável, pois um dos factores que faz confusão, é compreender como funciona o clique do rato.

O clique do rato pode consistir em pressionar uma tecla, ou pode acontecer ao fim de alguns segundos, estando o cursor do rato parado no local em que se pretende clicar [14].

O *SmartNav* tem uma precisão superior no controlo do rato em relação ao *Microsoft Kinect*, visto que, é a funcionalidade principal do dispositivo. Desta forma, pode efectuar alguns comandos com maior precisão comparando com a câmara de profundidade do *Microsoft Kinect* ao tentar detectar o ponto que controla o rato.



Figura 3: *SmartNav*

### **2.3.4. *Natural Point – OptiTrack***

O *OptiTrack* e o *OptiTrack Arena* são duas soluções que poderiam facilmente completar os objectivos deste trabalho. O *OptiTrack Arena* serve para capturas *full motion 3D*, utilizando várias câmaras de elevada precisão e um estúdio com condições favoráveis.

Estas duas soluções são de nível profissional e direccionadas a empresas que se dedicam nesta área, pois o custo de cada câmara é elevado [14].



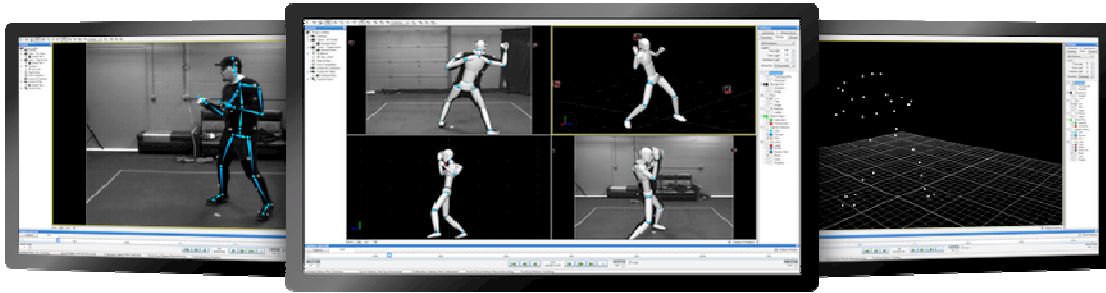


Figura 4: *Optitrack Arena*

### 2.3.5. Eye Tracking

O *Eye Tracking* utiliza uma grande componente de hardware para conseguir efectuar todos os cálculos necessários, a nível de software, para determinar com precisão o local que o utilizador está a olhar [21].

Através deste dispositivo seria também possível controlar o ambiente de imersão apenas em modo de visualização.

O inconveniente deste dispositivo é o facto de ser necessário ter o equipamento em contacto directo com o utilizador.

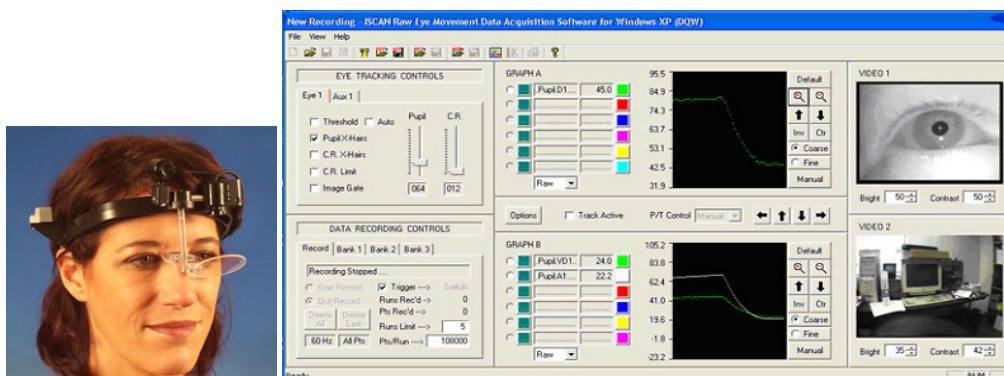


Figura 5: *Eye Tracking*

### 2.3.6. LEAP Motion

O *LEAP Motion*, da *Engadget*, aparenta ser um dispositivo que será revolucionário nesta componente de reconhecimento gestual, mais precisamente para os dedos.

Este produto ainda não foi lançado no mercado, mas espera-se o seu lançamento para 2013.

Trata-se de um dispositivo de tamanho reduzido, que pode ser colocado, por exemplo, entre o teclado e o monitor, e que faz uma leitura vertical à medida que a mão se vai aproximando dessa mesma zona que o sensor define. Ao entrar-se nessa zona, é feita uma construção em tempo real dos dedos e da mão do utilizador.

O *LEAP Motion*, apresenta uma capacidade de resposta quase imediata ao olho humano. Em várias demonstrações da *Engadget*, é feito um teste com o *Google Earth* e por vezes, a velocidade de processamento do *LEAP Motion* acaba por ser tão elevada, que o *buffering* do browser onde está a ser feita a demonstração, denota ligeiros atrasos [12].



Figura 6: LEAP Motion

## 2.4. Magnetómetros e *Smartphones*

Um magnetómetro é instrumento de medição utilizado para medir a força de um campo magnético e também, por vezes, para determinar a direcção desses campos.

Nos *smartphones* relativamente recentes, já se começa a verificar a presença de magnetómetros. Desta forma, o *smartphone* é capaz de mostrar ao utilizador a sua orientação geográfica que depende do tipo de magnetómetro utilizado, pois a sua referência pode ser o norte magnético ou não. Pode também saber a rotação que o *smartphone* tem perante o seu próprio eixo e também a sua inclinação [13].

O *iPhone 3GS* e *4* são os *smartphones* que se dizem ser os primeiros a ter um magnetómetro. Mas os *smartphones Android*, com cerca de 2 anos, também têm esta especificação e existem aplicações de bússolas digitais.

Algumas aplicações, não utilizam somente o magnetómetro devido a terem sido apresentadas críticas sobre o erro que existe nesses instrumentos. Essas aplicações utilizam o magnetómetro, o acelerómetro e o giroscópio, para poderem ter o devido controlo dos 3 eixos que envolvem o *smartphone*, com maior precisão [7].

Os *smartphones* apresentam os valores do magnetómetro em *microTesla* ( $\mu T$ ) nos 3 eixos; a força do campo geomagnético perante o eixo dos X, Y e Z. Imaginando que o *smartphone* está sobre uma mesa, assume-se um mapeamento em 2D, em que os valores de X e Y mudam consoante o mover ao longo do eixo X e Y respectivamente. Ao fazer mudar a altura em que o *smartphone* se encontra, o mapeamento já se torna 3D e o valor do eixo do Z irá mudar.

Também se podem capturar os valores do magnetómetro através do modo bússola que calcula o norte magnético. O eixo dos X será o *azimuth*, eixo dos Y será o *pitch* e o eixo dos Z será o *roll*. Estes valores são representados em ângulos perante o seu respectivo eixo [1].

## 2.5. Análise Comparativa

### 2.5.1. *Asus Xtion Pro Live*

	<i>Microsoft Kinect</i>	<i>ASUS Xtion Pro Live</i>
<b>Campo de Visão</b>		
Campo de visão horizontal	57 graus	58 graus
Campo de visão vertical	43 graus	45 graus
Alcance de profundidade	1.2 metros - 3.5 metros	0.8 metros - 3.5 metros
<b>Fontes de dados</b>		
Profundidade	320x240 16-bit @ 30 frames/seg	640x480 @ 30 frames/seg 320x240 @ 60 frames/seg
Cor	640x480 32-bit @ 30 frames/seg	1280x1024 @ 30 frames/seg
<b>Alimentação</b>	12 Volts DC + 5Volts Ligação USB	5Volts Ligação USB

Tabela 1: Comparação entre o *Microsoft Kinect* e o *Asus Xtion Pro Live*

Como referido, o *Asus Xtion Pro Live* utiliza a mesma tecnologia que o *Microsoft Kinect*.

Podemos verificar através da comparação de especificações, na figura 7, que existe uma superioridade no *Asus Xtion Pro Live*, a nível de *hardware* [20].

Apesar do *Asus Xtion Pro Live* apresentar melhores especificações de *hardware*, foi utilizado o *Microsoft Kinect* devido a ser superior a nível de *software*. Apresenta uma maior fiabilidade e compatibilidade a nível dos controladores, uma documentação detalhada da MSDN e por ser o equipamento disponível na Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

### 2.5.2. *Natural Point – OptiTrack e TrackIR*

Como o *OptiTrack* é uma solução profissional, verificam-se algumas especificações um pouco fora de série.

Dispositivo	Imagem	I/O Power
<p style="text-align: center;"><b>OptiTrack</b></p>	<ul style="list-style-type: none"> <li>• Pixel Size : 4.8 <math>\mu\text{m}</math> <math>\times</math> 4.8 <math>\mu\text{m}</math></li> <li>• Imager Size : 6.144 mm <math>\times</math> 4.9152 mm</li> <li>• Imager Resolution : 1280 <math>\times</math> 1024 (1.3 Megapixels)</li> <li>• Frame Rate: 30-120 FPS (adjustable)</li> <li>• Accuracy: Sub-millimeter</li> <li>• Latency: 8.3 ms</li> <li>• Shutter Type: Global</li> <li>• Shutter Speed:                             <ul style="list-style-type: none"> <li>○ Default: 500 <math>\mu\text{s}</math></li> <li>○ Minimum: 20 <math>\mu\text{s}</math></li> <li>○ Maximum: 7.5 ms (at 120 FPS)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Data: USB 2.0</li> <li>• Camera Sync: USB 2.0 (via OptiSync)</li> <li>• Power: USB 2.0 @ 1 A</li> </ul>
<p style="text-align: center;"><b>TrackIR</b></p>	<ul style="list-style-type: none"> <li>• Raw Sensor Resolution: 640 x 480</li> <li>• Reporting Resolution: 96,000 x 72,000</li> <li>• Field of View: 51.7°</li> <li>• Resolution/Horizontal Degree: 1850 subpixels/degree</li> <li>• Sample Rate: 120 fps</li> <li>• Response Time: 9 ms</li> </ul>	<ul style="list-style-type: none"> <li>• Connection: USB</li> </ul>

Tabela 2: Comparação entre dispositivos *Natural Point*

### 2.5.3. LEAP Motion

O *LEAP Motion* como ainda não foi lançado no mercado, a empresa *Engadget* não revelou que tipo de tecnologia está a ser utilizada, mas já se sabe que não é semelhante à tecnologia do *Microsoft Kinect* [2].

## Capítulo 3 Abordagem Adoptada e Planeamento

### 3.1. Objectivos

Com este trabalho pretende-se efectuar uma integração completa do *Microsoft Kinect* com o *Google Earth*, caso contrário, ao tentar-se controlar o rato do computador com o *Microsoft Kinect*, verifica-se que o cursor não mantém uma posição estável e rapidamente pode mover-se para um local não desejado.

Pretende-se assim reduzir esse erro que existe, procurando uma solução viável, que passa por redefinir e melhorar as primitivas gestuais do *Microsoft Kinect*.

O *Microsoft Kinect* deverá ser utilizado para captura de gestos e o *smartphone* deverá ser utilizado para o cálculo da posição do utilizador no local.

### 3.2. Casos de uso

#### 3.2.1. Microsoft Kinect

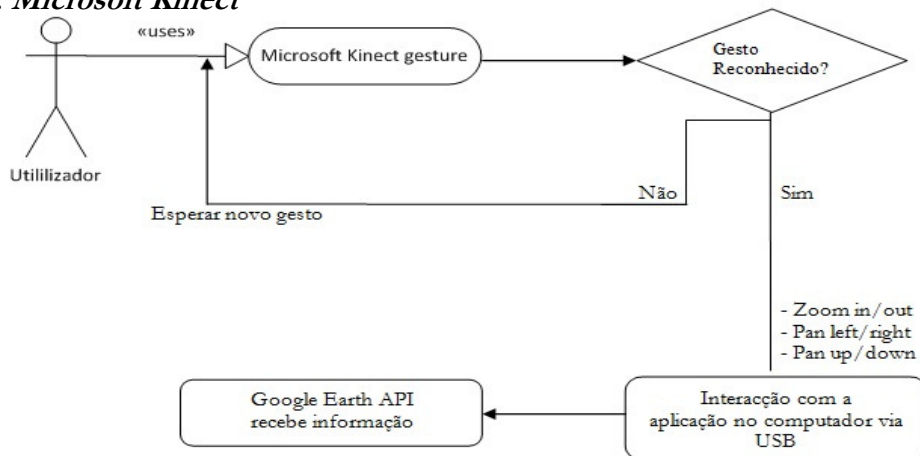


Figura 7: Caso de uso do *Microsoft Kinect*

#### 3.2.2. Smartphone

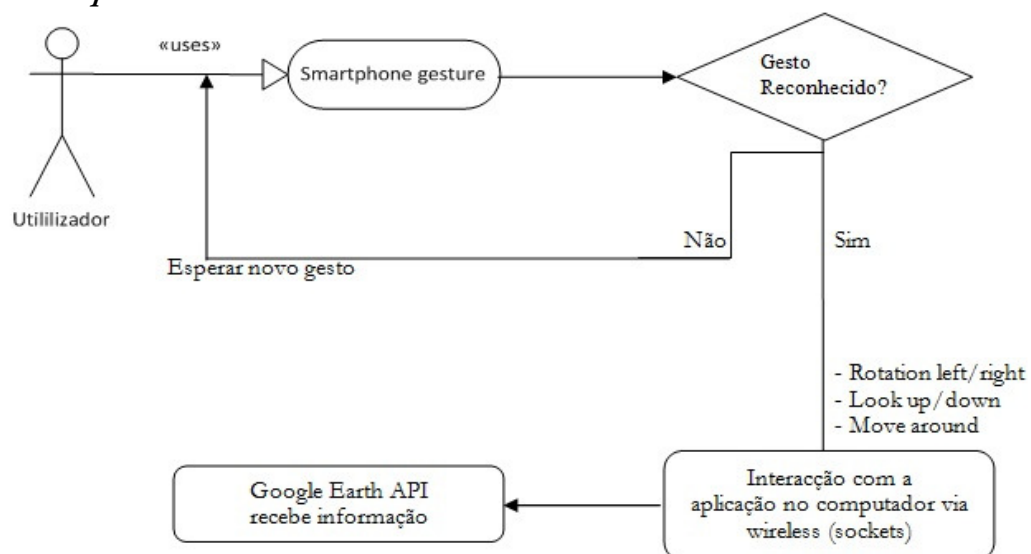


Figura 8: Caso de uso do *Smartphone*

### 3.2.3. Aplicação principal

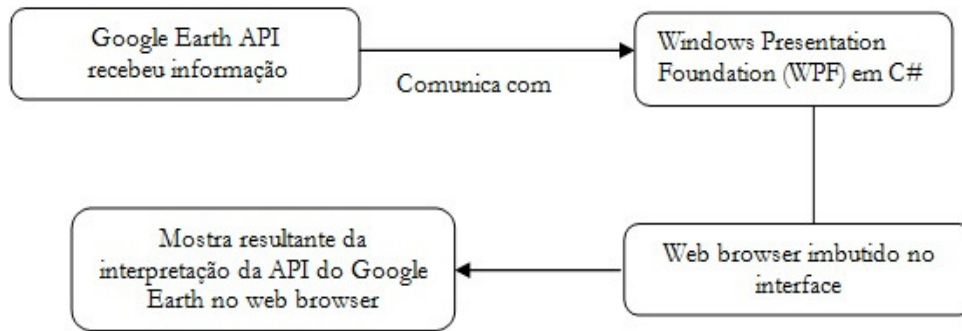


Figura 9: Caso de uso da aplicação principal

### 3.3. Requisitos

O que é requerido para este projecto é a obtenção de dados provenientes do magnetómetro do *smartphone*, que serão enviados para uma aplicação central que após a sua recepção, os trata e aplica no *Google Earth*.

Relativamente ao *Microsoft Kinect*, pretende-se efectuar o reconhecimento de imagem de modo a que seja possível capturar a mão, os dedos e as suas extremidades. Esta componente estará ligada directamente à aplicação principal que terá todo o repositório gestual para poder associar à respectiva acção.

A aplicação principal, tem de estar pronta a receber os dados do *smartphone* e em simultâneo identificar qualquer gesto efectuado por parte do *Microsoft Kinect*.

A *Framework* que vai ser utilizada para programar em C# será o *Microsoft Visual Studio 2012*. Para a programação do *Smartphone Android (Java)* será utilizado o *Eclipse*.

O *Kinect SDK 1.6* tem de estar instalado no computador. Não foi feita a actualização de versão, devido a algumas funções terem sofrido alterações.

O *Microsoft Kinect* tem de ser a versão para *Windows* e o *smartphone* a utilizar, tem de ter um magnetómetro. Pode ser um *iPhone 3GS* ou superior, ou um *Android* relativamente recente.

### 3.4. Metodologia

Este trabalho passa por uma metodologia de investigação e também de experimentação, sendo assim dividido em duas componentes; uma das componentes será responsável pela detecção dos gestos do utilizador. Esta é a componente do *Microsoft Kinect*, que após ser feita a interligação com a API do *Google Earth* irá interagir directamente com esta, e efectuar operações de *Zoom*, *Pan* e toda a interacção com o interface da aplicação através da detecção das pontas dos dedos.

A outra componente, do *smartphone*, será responsável pela obtenção dos dados recolhidos pelo magnetómetro que serão enviados para a aplicação principal via *wireless*. Esses dados serão utilizados para o posicionamento do utilizador em modo de imersão.

A aplicação principal será responsável pelo controlo da API do *Google Earth* para a obtenção dos efeitos desejados e será também responsável pela interacção que existe com o interface da mesma.

### **3.4.1. Microsoft Kinect**

Tendo em consideração as componentes do *Microsoft Kinect*, vai ser tirado maior partido da câmara de profundidade juntamente com o sensor de infravermelhos.

Na programação do *Microsoft Kinect*, em C# e utilizando o *Kinect SDK 1.6*, pretende-se aumentar a precisão do sistema, melhorando, nomeadamente o de captura das primitivas gestuais, reformulando alguma delas, de modo a que sejam mais perceptíveis para o sensor.

Não serão efectuadas alterações no SDK do *Microsoft Kinect*, mas será utilizado o *near mode* para eliminar possíveis interferências de objectos a mais de 80cm do sensor. Após estar definido o intervalo de profundidade a que será feita a detecção das mãos, todas as primitivas gestuais previamente existentes serão eliminadas e será feito o reconhecimento de imagem para a captura da mão e depois poder efectuar o cálculo do mapeamento virtual dos dedos e suas extremidades.

Após ter sido adquirido o mapeamento das pontas dos dedos, serão criadas novas primitivas de acordo com os valores que o *Microsoft Kinect* irá devolver, para que, possa existir uma maior precisão e sensibilidade para a interacção com o *Google Earth*.

### **3.4.2. Smartphone**

A nível do *smartphone* será criada uma aplicação que efectue leituras do magnetómetro. Essas leituras serão enviadas via *wireless* para a aplicação principal.

As leituras serão feitas em tempo real para permitir também um controlo em tempo real do *Google Earth*.

Os movimentos que afectam o magnetómetro do *smartphone* para este projecto, serão movimentos de rotação dos 3 eixos cartesianos. Através desses movimentos de rotação feitos com o *smartphone*, o utilizador pode decidir para onde olhar e também para onde se mover, dentro do modo de imersão.

O *smartphone* terá um papel exclusivo para o modo de imersão.

### **3.4.3. Aplicação principal**

A aplicação principal terá duas *threads*, uma delas estará em background e será responsável por receber e tratar toda a informação proveniente do *smartphone*. A *thread* ficará em background para que ao ser terminado o programa a mesma também seja terminada.

A *thread* principal será responsável pela componente do *Microsoft Kinect*, que tratará de todos os gestos efectuados e fará a interligação com o *Google Earth*.

A aplicação principal ficará assim apta para que os resultados do *smartphone* e do *Microsoft Kinect* sejam apresentados em tempo real no *Google Earth*.

### 3.5. Abordagem

Tendo em atenção um projecto que existe como base para este trabalho, além do estudo do estado da arte e determinar quais os métodos mais apropriados para a elaboração desta investigação, foi feita uma exploração detalhada do seu funcionamento. Foram encontrados vários problemas, dos quais muitos deles se pretendem eliminar.

Um dos problemas principais é a falta de precisão que se faz sentir e que pode ser notada ao escolhermos a opção de ver o que o *Microsoft Kinect* está a captar. Verifica-se que uma pessoa mesmo estando parada, dificilmente consegue ser interpretada como tal, no *Microsoft Kinect*. Este constrangimento torna difícil determinar qual o gesto efectuado pelo utilizador e obriga-o a repetir o movimento com menor velocidade e com maior amplitude.

Para resolver esta situação, foi tomada a decisão de dividir a percepção total do ambiente, isto é, a percepção dos gestos e a da movimentação em dois dispositivos, *Microsoft Kinect* e *smartphone* respectivamente, como já referido na metodologia.

#### 3.5.1. Arquitectura

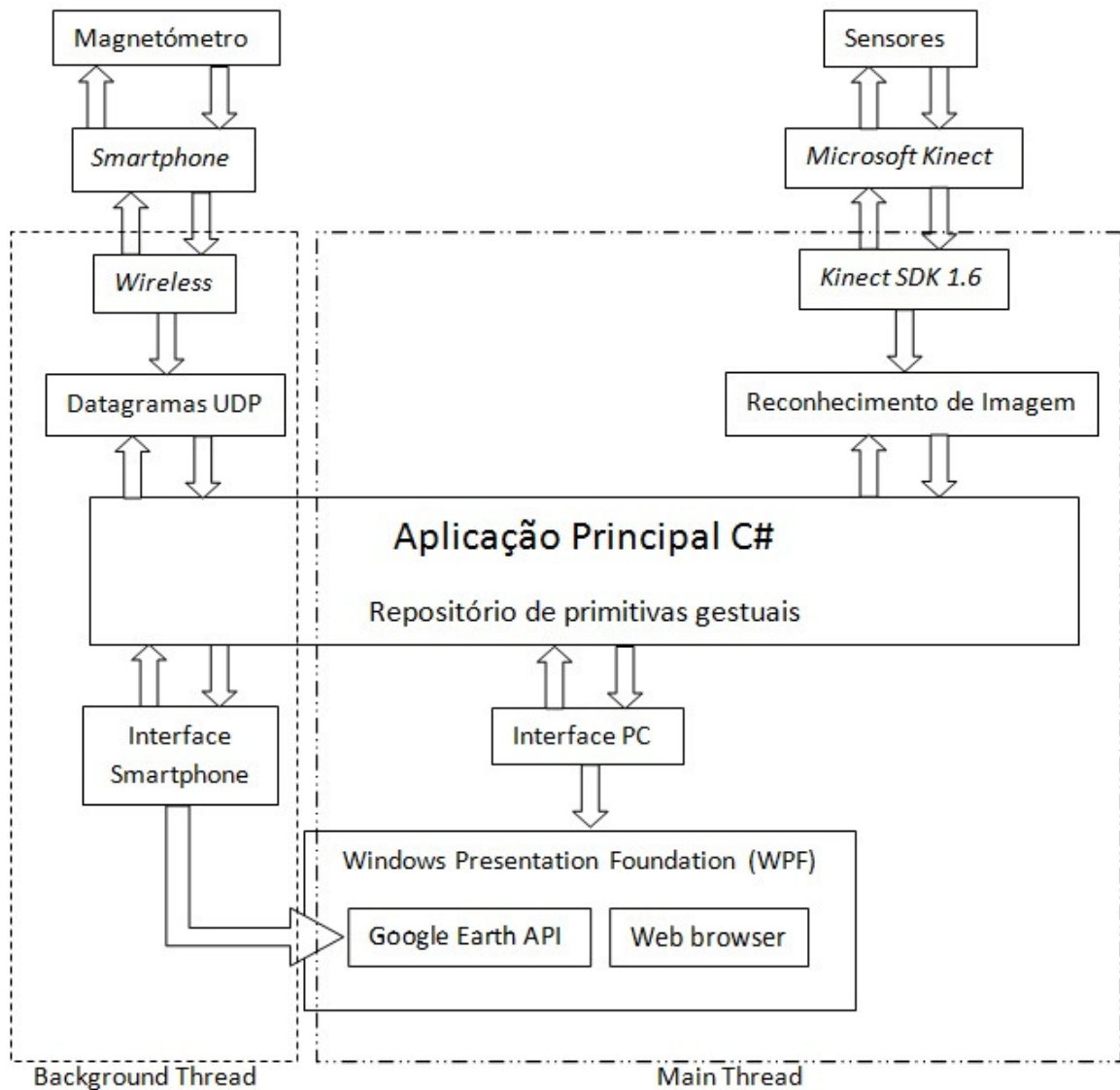


Figura 10: Arquitectura do projecto



Nesta fase do projecto, a sua arquitectura pode-se traduzir na utilização do *Microsoft Kinect* e do *smartphone*, por parte do utilizador. Após a sua interacção, os dados são processados num computador que, por sua vez, apresenta os resultados conforme a interpretação feita pela aplicação que se encontra no computador.

### **3.5.2. Validação**

Esta investigação será validada segundo os parâmetros de funcionalidade dos vários dispositivos.

Ao longo da construção das várias componentes, serão feitos vários testes à precisão do equipamento, com o intuito de diminuir o erro existente em condições normais.

A capacidade de integrar em simultâneo o *Microsoft Kinect* e o *smartphone* com a aplicação será um factor importante para determinar a viabilidade deste projecto.

Pretende-se que o *Microsoft Kinect* reconheça com alguma rapidez a mão do utilizador assim como os dedos, e consiga também fazer a sua reconstrução virtual para poder interagir com o *Google Earth*.

O *smartphone* será avaliado pela capacidade de resposta por parte do magnetómetro como controlador da posição e orientação do utilizador.

### **3.5.3. Testes e planeamento**

Nesta fase de investigação, já foram efectuados alguns testes ao *Microsoft Kinect* de modo a pôr de parte alguns factores de risco encontrados. No capítulo do estado da arte, já foram mencionados esses testes embora, de uma forma abreviada.

Os testes à velocidade do interface USB, à distância mínima da câmara de profundidade e aos frames por segundo tiveram prioridade pois era pretendido saber até que ponto poderia o *hardware* estar limitado. Após estes testes terem sido elaborados com sucesso, foi testada a aplicação base que permitiu saber quais os pontos a melhorar no desenvolvimento deste projecto.

Após estes testes terem sido efectuados, foi ponderado o papel que cada componente terá na aplicação principal.

Pretende-se que a aplicação principal funcione em dois modos. O modo *default* e o modo de imersão. O modo *default* inicia com a vista do planeta como habitualmente se costuma ver no *Google Earth*. Quando esse modo é invocado novamente, ao sair do modo de imersão, ele vai apresentar a vista da zona, que previamente estava em imersão, com uma altura de 4000m em relação ao solo. O modo *default* é para ser controlado somente pelo *Microsoft Kinect*, com a função de mover o mapa consoante o posicionamento dos dedos no *Google Earth*.

O modo de imersão, efectua a aproximação ao solo mantendo uma altura média de 1.8m. Este modo simula o *street view* do *Google Earth*. Não é semelhante ao *street view*, pois pretende-se navegar livremente em todo o mapa. Este modo é para ser controlado pelo *smartphone*, com excepção da função de *zoom* que será o *Microsoft Kinect* a controlar.

### 3.6. Estudos Preliminares

#### 3.6.1. Trabalho Base

Para a elaboração deste projecto existe um trabalho já desenvolvido no âmbito da cadeira de Sistemas Ubíquos de 2011/2012, que irá servir como base de desenvolvimento desta investigação.

Nesse trabalho apenas está feita a ligação entre o *Google Earth* e o *Microsoft Kinect*. São também detectadas, com muita dificuldade, algumas primitivas gestuais previamente definidas.

#### 3.6.2. Resultados

Na fase inicial da investigação apenas existiam os resultados dos testes feitos ao nível do *Microsoft Kinect* para verificar a sua capacidade e o funcionamento do trabalho base.

Não existem testes ao magnetómetro do *smartphone*, devido a nessa fase, não ter em minha posse, um aparelho com essa tecnologia.

Inicialmente, foram considerados três factores que poderiam determinar o impedimento da obtenção do objectivo pretendido, que foram descritos no capítulo do Estado da Arte, relativamente às limitações do *Microsoft Kinect*.

Ordem de relevância	Limitação	Resultado do teste
Baixa	Frames por segundo $\leq 30$	Frames por segundo $\leq 30$
Média	Resolução do Kinect até 320x240	Resolução até 1280x960
Elevada	Interface USB do Kinect apenas funcionar até uma velocidade de 15~16 Mbps	Velocidade USB semelhante a um dispositivo USB 2.0 normal

Tabela 3: Possíveis factores de risco

Foi repensado o factor de 30 frames por segundo, por ter uma relevância muito baixa devido ao objectivo da captura de gestos não requerer velocidades rápidas.

## Testes aos factores de risco

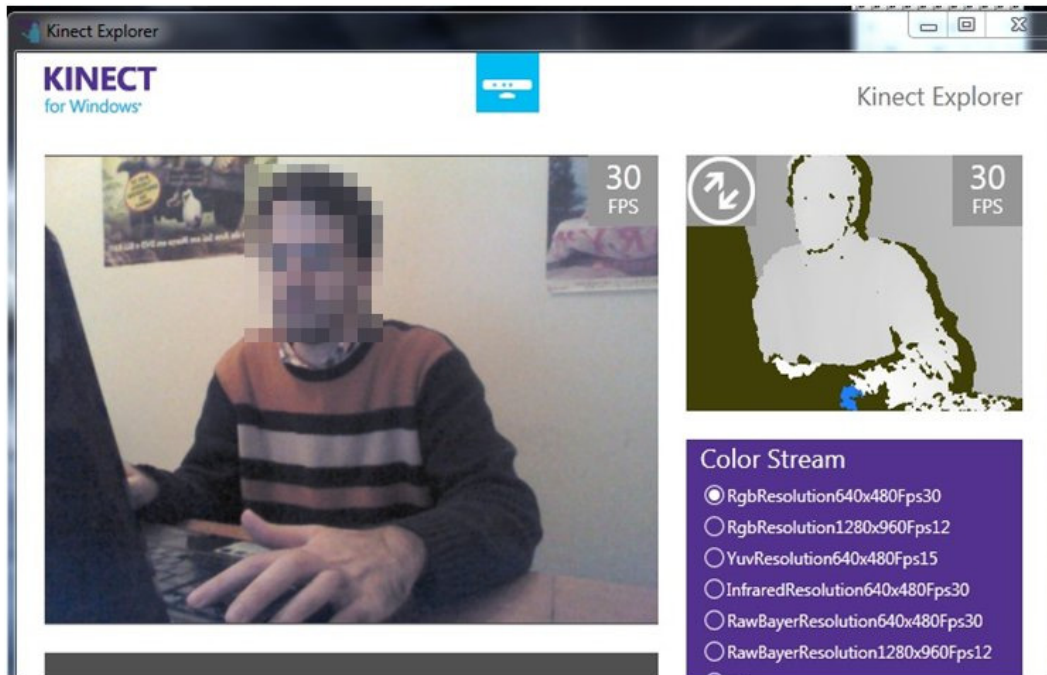
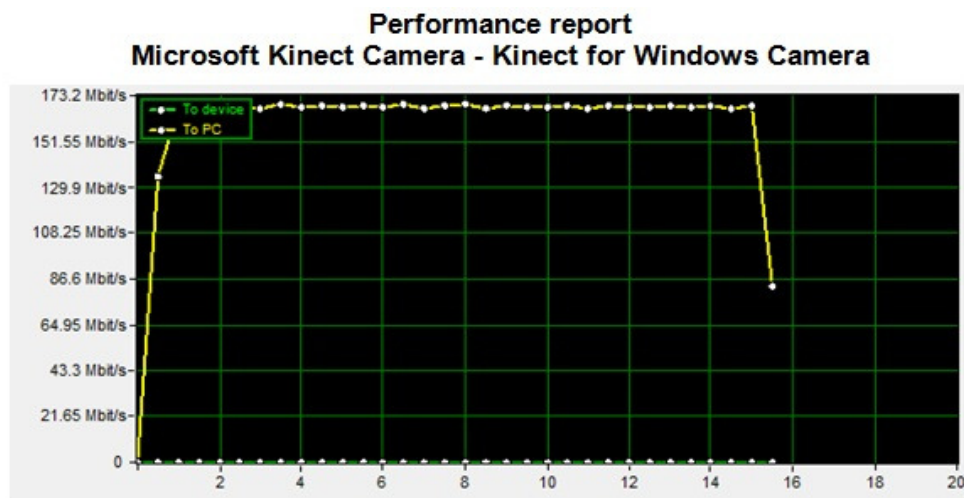


Figura 11: Teste aos frames por segundo e à resolução do *Microsoft Kinect*



### Report information

Scale	Auto (168.98 Mbit/s)
Units	Mbit/s

Figura 12: Teste à velocidade do USB do *Microsoft Kinect*

O teste da figura 11 foi elaborado com o software do *Microsoft Kinect - 'Developer Toolkit Browser'*. O teste da figura 12 foi elaborado com o programa '*Advanced USB Port Monitor*'.

### 3.6.3. Interpretação dos resultados obtidos

Os resultados obtidos na figura 11, mostra os frames por segundo obtidos, na câmara de profundidade e na câmara RGB.

Como o *Microsoft Kinect* tem uma sensibilidade muito elevada, os gestos não podem ser repentinos, o que faz com que 30 frames por segundo à resolução de 640x480 seja um valor aceitável para a realização deste projecto.

Se a velocidade do interface USB estivesse limitada com os parâmetros da Xbox, possivelmente a realização deste projecto não poderia ser concebida, pois seria necessária uma boa largura de banda para a comunicação entre o *Microsoft Kinect* e o computador. Com apenas 16Mbps, a câmara de profundidade não seria capaz de captar mais nada para além do esqueleto virtual mencionado no Estado da Arte. Tendo obtido velocidades semelhantes a um dispositivo USB 2.0 normal, já é possível completar o objectivo deste projecto.

## 3.7. Plano de Trabalho e Implicações

### 3.7.1. Primeiro semestre

Planeamento inicial:

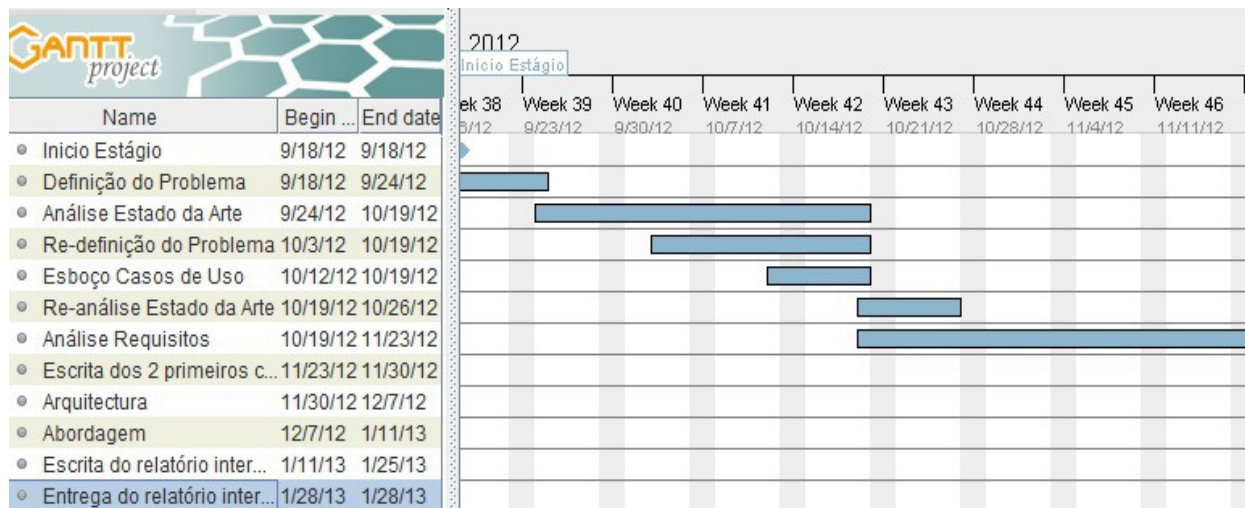


Figura 13: Diagrama de *Gantt* do planeamento inicial do primeiro semestre (parte 1/2)

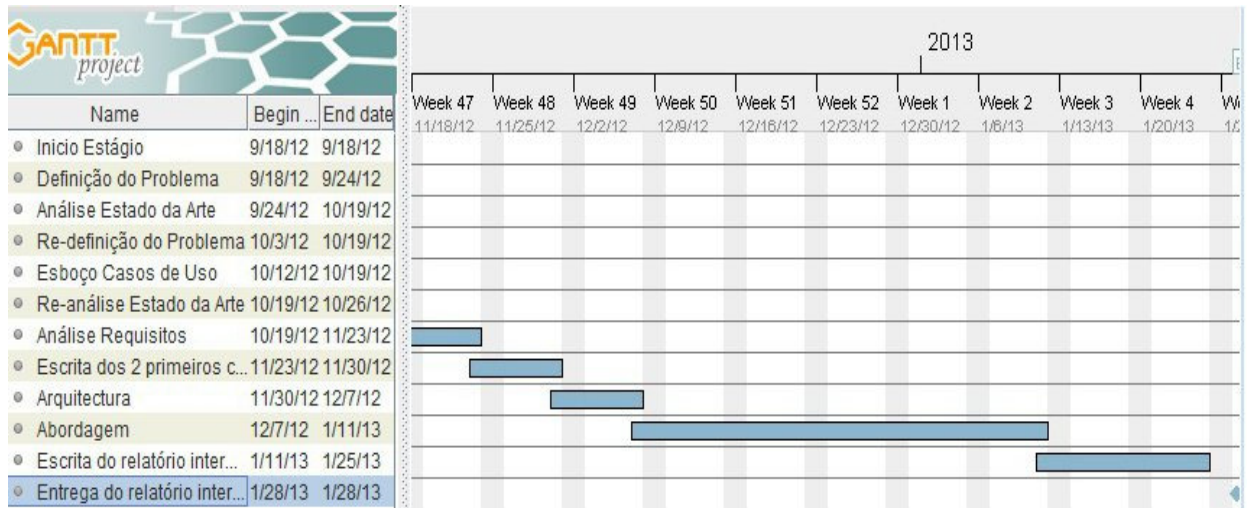


Figura 14: Diagrama de Gantt do planeamento inicial do primeiro semestre (parte 2/2)

### Planeamento final

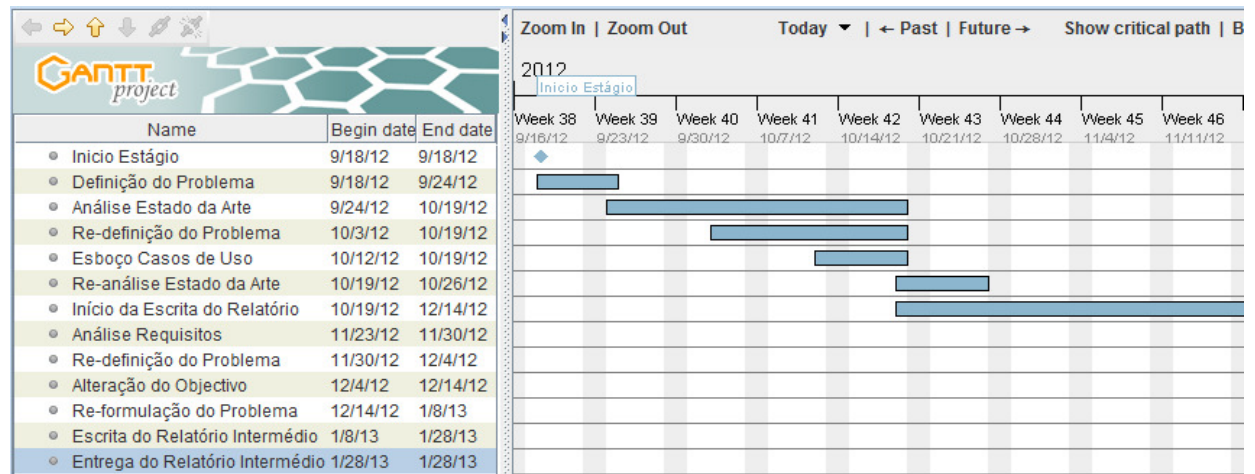


Figura 15: Diagrama de Gantt do planeamento final do primeiro semestre (parte 1/2)

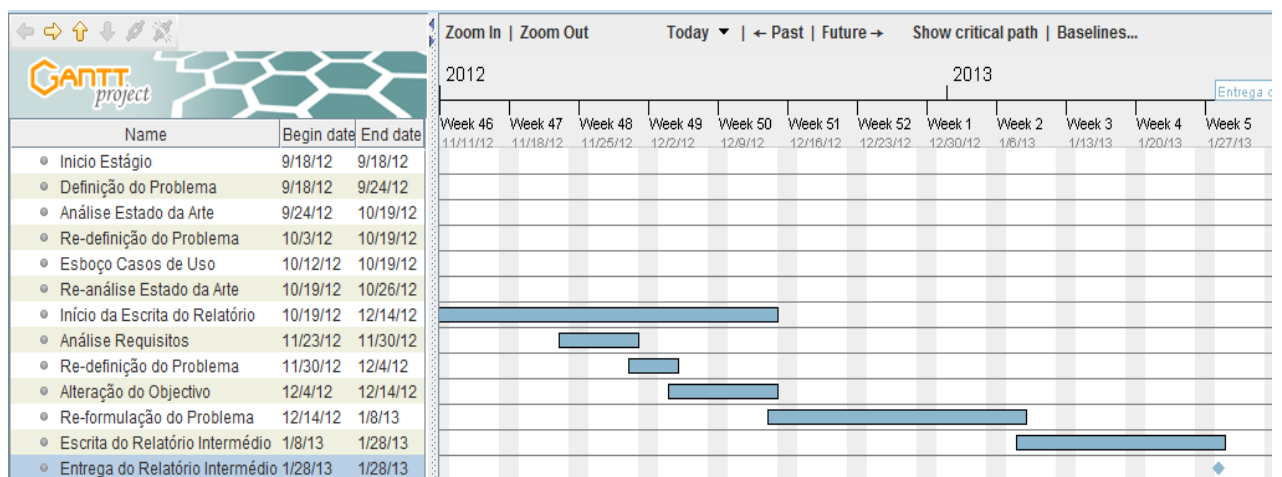


Figura 16: Diagrama de Gantt do planeamento final do primeiro semestre (parte 2/2)



### 3.7.2. Segundo semestre

#### Planeamento inicial

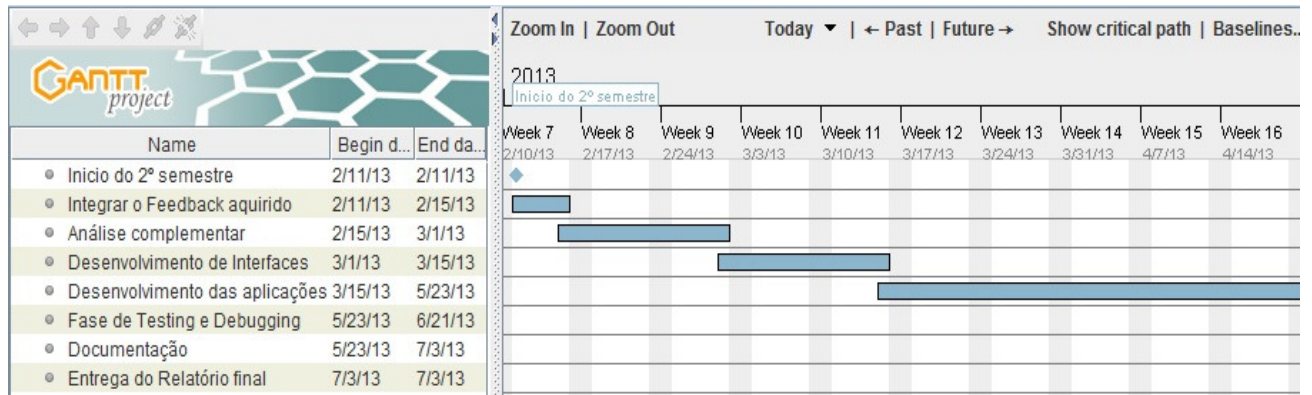


Figura 17: Diagrama de Gantt do planeamento inicial do 2º semestre (parte 1/2)

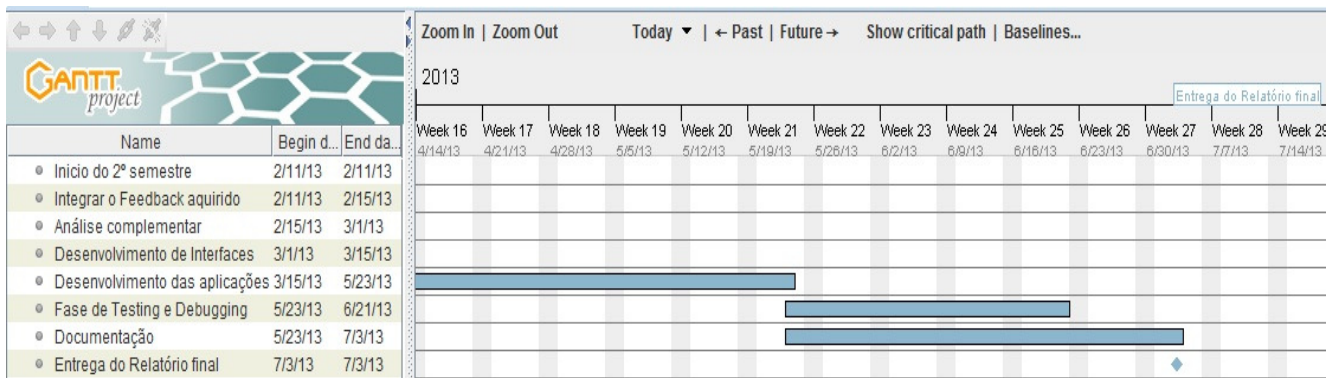


Figura 18: Diagrama de Gantt do planeamento inicial do 2º semestre (parte 2/2)

### 3.8. Resumo

Neste capítulo foi referido o objectivo deste projecto. Por um lado, devido à baixa precisão do *Microsoft Kinect*, pretende-se aliviar o método de captura de pontos que não serão considerados relevantes. Ao ser feita essa parte, é nosso objectivo aumentar a precisão do *Microsoft Kinect*, deixando para o *smartphone* as operações que o *Microsoft Kinect* deixou de conseguir fazer.

O utilizador interage com o *Microsoft Kinect* apenas com os gestos das mãos e dos dedos, para poder efectuar operações de *zoom* e *pan*. O *smartphone* será responsável por saber se o utilizador pretende rodar e inclinar a perspectiva da câmara já dentro da aplicação principal.

Foi também mencionada a existência de um trabalho que será utilizado como referência para o desenvolvimento deste projecto.

Neste capítulo foram também referidos os potenciais factores de risco e os testes que os levaram a ser descartados.

Foram apresentados os diagramas de *Gantt* do planeamento do primeiro e segundo semestre. É também possível verificar-se a existência de vários desvios no planeamento do primeiro semestre.

## Capítulo 4

# Análise Complementar e Implementação

### 4.1. Resumo

Neste capítulo são descritas as dúvidas, problemas e aproximações feitas ao projecto que foram surgindo até se chegarem a conclusões e resultados fiáveis para a tomada de decisão final.

#### 4.1.1 Smartphone

Na utilização inicial com o *smartphone*, nomeadamente o magnetómetro, verificaram-se algumas dificuldades. Numa primeira aproximação, pretendia-se captar os valores em *microTesla* e depois serem efectuados os cálculos necessários para determinar a posição do utilizador com um sistema de calibração a ser implementado. Mas verificou-se que o magnetómetro não estava a mostrar os valores correctamente e foi feita uma comparação de valores com outro *smartphone*. Verificou-se então, que os dados não coincidiam e como a comparação foi feita com um *iPhone*, que mostra directamente tais valores e com fiabilidade, foi descartada a utilização dos valores em *microTesla*.

Na segunda aproximação ao problema, foram captados os valores em modo bússola. Em que eram obtidos os valores de *azimuth*, *pitch* e *roll* em graus. Quando o valor de *azimuth* é 0, significa que estamos apontados para o Norte magnético, 90 corresponde a Este, 180 a Sul e

270 a Oeste. Os valores de *pitch* variam entre -180 e 180 graus, que corresponde à inclinação do *smartphone* para a frente e para trás. O valor 0 corresponde à posição horizontal, pousado numa mesa. O *roll* varia entre -90 e 90 graus. Quando o *smartphone* está inclinado para a esquerda apresenta valores positivos, inclinado para a direita apresenta valores negativos.

O *azimuth* corresponde à rotação, em graus, no eixo dos X, o *pitch* no eixo dos Y e o *roll* no eixo dos Z.

Ao ser decidida a utilização do *smartphone*, foi também necessário pensar no modo em como os dados iriam ser transmitidos para a aplicação principal.

Inicialmente, foi ponderada a transferência via *bluetooth*, visto que, o *smartphone* rapidamente se emparelha com o portátil e a distância que iria existir entre os dois dispositivos não iria passar dos 10 metros. Essa implementação tornou-se impossível devido ao facto, do portátil a ser utilizado ser um *Toshiba* e o sistema de *bluetooth* ser proprietário da marca e não disponibilizando esta, uma API para desenvolvimento.

Após a tentativa com o *bluetooth*, passou-se para a hipótese do protocolo MQTT. Este protocolo, tem como finalidade minimizar a utilização de largura de banda da rede ao mesmo tempo que, assegura uma fiabilidade até um certo nível, na troca de mensagens entre dispositivos. É ideal para dispositivos móveis onde a largura de banda e o consumo energético estão em causa.

A tentativa de implementação deste protocolo, que aparentava ser a melhor solução, também falhou. Durante o processo de *debug*, verificou-se que do lado cliente deste



protocolo, eram utilizados *sockets*. Este ponto levou a uma reanálise da componente do *smartphone*, visto que, só é necessário enviar uma *string* com dados cada vez que o magnetómetro detecte uma alteração de valores.

Desta forma, a solução foi a utilização de uma ligação *wireless* através de *sockets*.

#### 4.1.2. Microsoft Kinect

Desde o início da experimentação com o *Microsoft Kinect*, a câmara de profundidade tornou-se o foco principal, pois é necessária para ser possível cumprir os objectivos deste projecto.

Ao serem experimentados vários projectos que testam a câmara de profundidade, rapidamente se chegou à conclusão que a sua precisão é baixa e a sensibilidade é muito elevada. O fraco equilíbrio entre sensibilidade e precisão levou a questionar os controladores do *Microsoft Kinect* e o próprio *Kinect SDK* sendo necessário efectuar testes como já referido.

Tanto com o *Kinect SDK* com os controladores de origem da *Microsoft*, como utilizando o *OpenNI* e os seus controladores respectivos, a câmara de profundidade não apresentou melhorias relevantes para solucionar o problema [15].

Um aspecto interessante que se fez denotar numa altura posterior do projecto, foi o desempenho da câmara de profundidade em relação a computadores diferentes.

##### Rate and improve your computer's performance

The Windows Experience Index assesses key system components on a scale of 1.0 to 7.9.

Component	What is rated	Subscore	Base score
<b>Processor:</b>	Calculations per second	7,4	 Determined by lowest subscore
<b>Memory (RAM):</b>	Memory operations per second	7,8	
<b>Graphics:</b>	Desktop performance for Windows Aero	7,3	
<b>Gaming graphics:</b>	3D business and gaming graphics performance	7,3	
<b>Primary hard disk:</b>	Disk data transfer rate	7,3	

Component	What is rated	Subscore	Base score
<b>Processor:</b>	Calculations per second	6,9	 Determined by lowest subscore
<b>Memory (RAM):</b>	Memory operations per second	7,4	
<b>Graphics:</b>	Desktop performance for Windows Aero	3,9	
<b>Gaming graphics:</b>	3D business and gaming graphics performance	5,7	
<b>Primary hard disk:</b>	Disk data transfer rate	5,6	

Figura 19: *Rating* do computador fixo

Figura 20: *Rating* do computador portátil

Devido a existir uma diferença considerável na *performance* dos dois computadores, a câmara de profundidade teve comportamentos diferentes. Enquanto que, no computador portátil o *tracking* dos dedos por vezes se perdia, no computador fixo era algo que não acontecia com a mesma frequência.

Este aspecto fez com que a restante implementação deste projecto fosse efectuada no computador de desempenho mais elevado, vindo a ser testada várias vezes no computador portátil para assegurar os resultados pretendidos.

### 4.1.3. Aplicação Principal

A aplicação principal, que é responsável por juntar a interacção do *smartphone* com o *Microsoft Kinect* e apresentar os resultados no *Google Earth*, teve como base um projecto anterior, como já foi referido. Assim sendo, inicialmente o projecto tinha a capacidade de receber alguns gestos predefinidos, utilizando o sistema normal de mapeamento do esqueleto. Tinha também a implementação da API do *Google Earth* no *web browser* embutido na aplicação C# (WPF).

Numa primeira aproximação, todas as primitivas gestuais existentes, que eram relativas às *joints* existentes no *Kinect SDK*, foram removidas, pois o que é pretendido é o mapeamento da mão e dos dedos. Após essa remoção, a aplicação somente mostrava o *Google Earth*, e tinha o modo de entrar e sair em imersão, mostrar e esconder a visualização 3D dos edifícios e a ligação à câmara do *Microsoft Kinect*.

A abordagem inicial foi direccionada para o *smartphone*. A recepção dos dados enviados foi bastante simples, mas é necessária a conversão dos valores recebidos para coordenadas que possam ser interpretadas pela API do *Google Earth*. Este processo demorou um pouco mais do que o previsto, devido a não ter havido uma compreensão por completo logo no início da implementação destes métodos.

Após a componente do *smartphone* estar dada por terminada, passou-se para o *Microsoft Kinect*. Esta componente teve de ser reimplementada apesar de já existir, pois foi necessário juntar novas referências e retirar outras que não estavam a ser utilizadas. Foram também importadas para este projecto, algumas soluções *open source* [18]. Essas soluções não são suficientes para a resolução deste projecto, mas fazem parte de vários estudos em que tiram o máximo proveito da câmara de profundidade, utilizando o *Kinect SDK*, e conseguem delimitar com sucesso a zona de acção do *Microsoft Kinect* para a detecção das mãos e dedos do utilizador.

Para a última fase do projecto, foi necessário pensar em como usar as pontas dos dedos. Primeiramente pensou-se em utilizar numeração, visto que era possível fazer a contagem de quantos dedos estavam a ser mostrados. Com cinco ou zero dedos, não era efectuada operação nenhuma, com um dedo seria feita a correspondência ao *click*. Ao estar só um dedo, seria apenas necessário mover a mão e o *Google Earth* iria funcionar como se tivesse a ser controlado pelo rato. Mas o problema apareceu quando surgiram as questões de duplo *click* e *zoom*. Foi necessária outra aproximação do problema o que levou à necessidade da criação de um interface intuitivo para a escolha das várias funcionalidades possíveis.

Apesar de o projecto ser capaz de captar as pontas de todos os dedos, para atingirmos os objectivos requeridos, optou-se por se fazer foco no dedo indicador da primeira mão a ser detectada, como controlador principal da componente *Microsoft Kinect*, funcionando por isso também para utilizadores esquerdinos.

Na etapa de finalização, verificou-se que a câmara de profundidade é afectada por inúmeros factores que nem sempre são controlados; como a distância entre o *Microsoft Kinect* e a parede, a cor da nossa roupa e a cor da parede. Para resolver esse problema, foi adicionado um sistema de calibração antes de se entrar na aplicação em si.

## 4.2. Implementação

Nesta secção é referida a implementação das várias componentes que constituem este projecto. São também mencionadas duas partes de código que foram consideradas relevantes.

A componente do *Microsoft Kinect* está em separado do tópico da aplicação principal, apenas para facilitação de compreensão. Essa componente foi desenvolvida em conjunto com a aplicação principal.

### 4.2.1. Smartphone

O desenvolvimento da aplicação para o *smartphone* foi elaborado em *Java*, utilizando o *Eclipse* como *Framework*. Trata-se de um *smartphone Android* versão 2.3.3. Ao iniciar a aplicação, é apresentado um menu que activa o magnetómetro e podem ser visualizados os valores que o mesmo regista. É pedido um endereço IP, este endereço refere-se ao endereço do computador que tem a aplicação em execução.

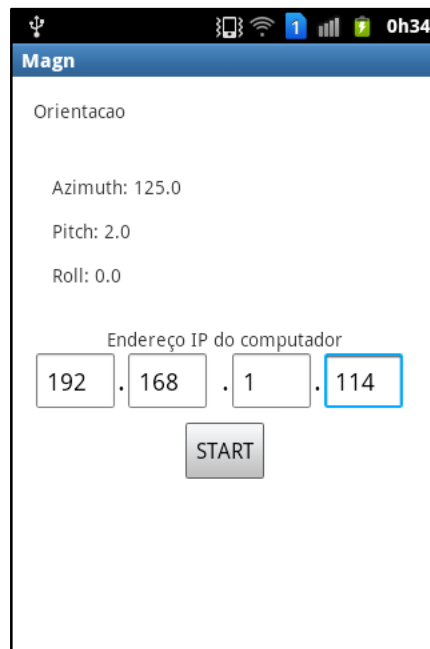


Figura 21 : Menu inicial do *smartphone*

Após ter sido inserido um endereço IP, é verificada a sua validade. Teoricamente falando, um endereço IP poderia começar em 0, mas como na prática existem alguns sistemas de redes que não suportam essa configuração, não é permitido inserir um endereço IP que cujo valor do primeiro octeto seja 0. Também não é permitido ter octetos superiores ao valor 255, ou todos os octetos com valor 255. E por último, não é permitido deixar campos

vazios. Foi ponderado o caso de o último octeto ser 255 ou 0, mas com uma correcta aplicação de *subnetting*, com máscara de 255.255.0.0, por exemplo, já é possível existir um endereço IP válido que termine em 255 ou 0.

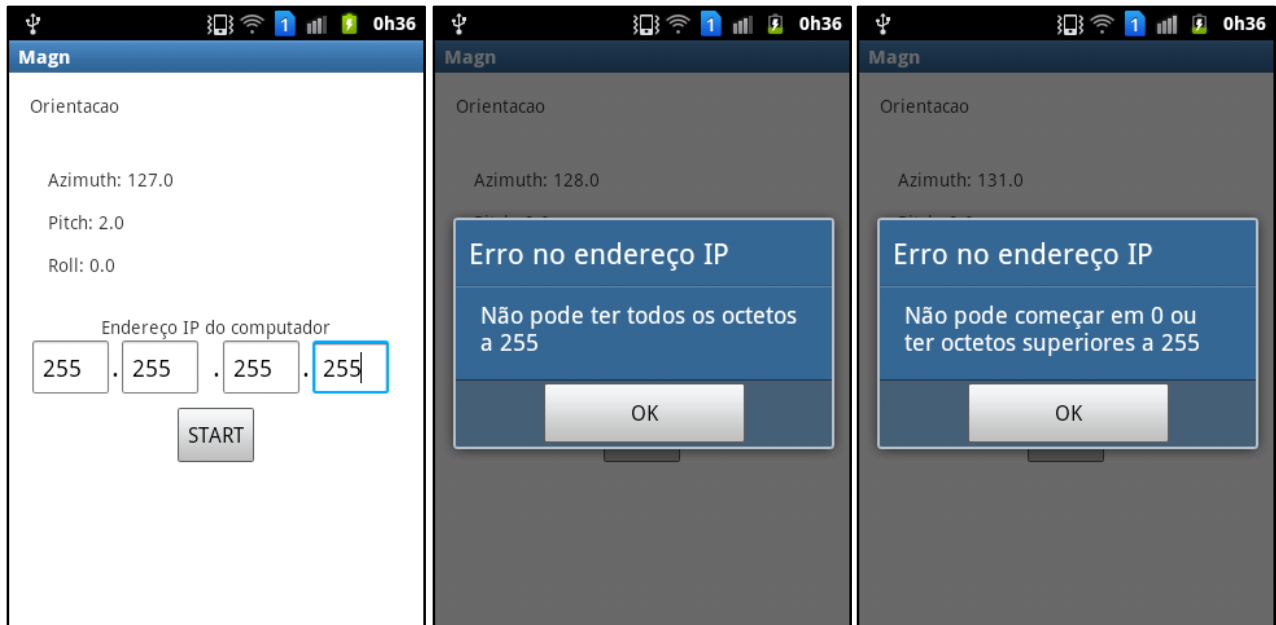


Figura 22: Exemplos de *handling* de erros no endereço IP

Após o endereço de IP ter sido validado, os valores que o magnetómetro está constantemente a mostrar no ecrã do *smartphone*, vão ser enviados via *wireless* em pacotes UDP. No evento de *'onSensorChanged'* são lidos os valores actuais do magnetómetro e são então, enviados para o endereço IP inserido no menu anterior.

Para não gerar muito tráfego na rede foram escolhidos pacotes UDP, devido a não ser necessário esperar pelo *acknowledge* para poder enviar o próximo pacote. Mesmo que exista a perda sucessiva de um ou dois pacotes, não vai afectar directamente o funcionamento da aplicação, apenas causa um ligeiro atraso enquanto não recebe um pacote.

Dentro de cada pacote, irá a informação relativa ao *azimuth*, *pitch* e *roll* em formato *string*. Essa *string* tem a configuração de acordo com o seguinte exemplo: *'L\_180.0\_2.0\_1.0'*. A primeira letra varia entre L e P, que serve para determinar se as coordenadas se tratam para localização ou posicionamento, respectivamente. Os símbolos de *underscore* *'\_'*, são os delimitadores para os valores de *azimuth*, *pitch* e *roll*.

A aplicação do *smartphone* funciona com base em 3 *Activities*, a principal activa o magnetómetro e valida o endereço IP do computador que tem a aplicação principal em execução. As outras duas são responsáveis por registar e enviar os dados do magnetómetro cada vez que se verifique uma alteração do estado do sensor. Uma é responsável pela localização e a outra pelo posicionamento do utilizador em modo de imersão no *Google Earth*.



Figura 23: *Smartphone* em modo de localização



Figura 24: *Smartphone* em modo de posição

Os botões de *START* e *STOP*, servem para iniciar e terminar o envio dos pacotes UDP. Para evitar possíveis problemas que foram ocorrendo no computador portátil a nível da API do *Google Earth*, só é possível fazer *STOP* ao programa quando estão a ser enviadas coordenadas de localização.

### Diagrama de fluxo

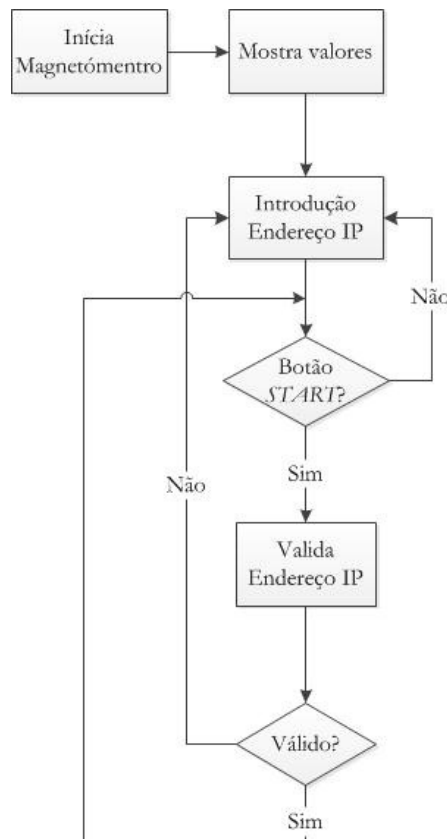


Figura 25: 1ª parte do Diagrama de Fluxo

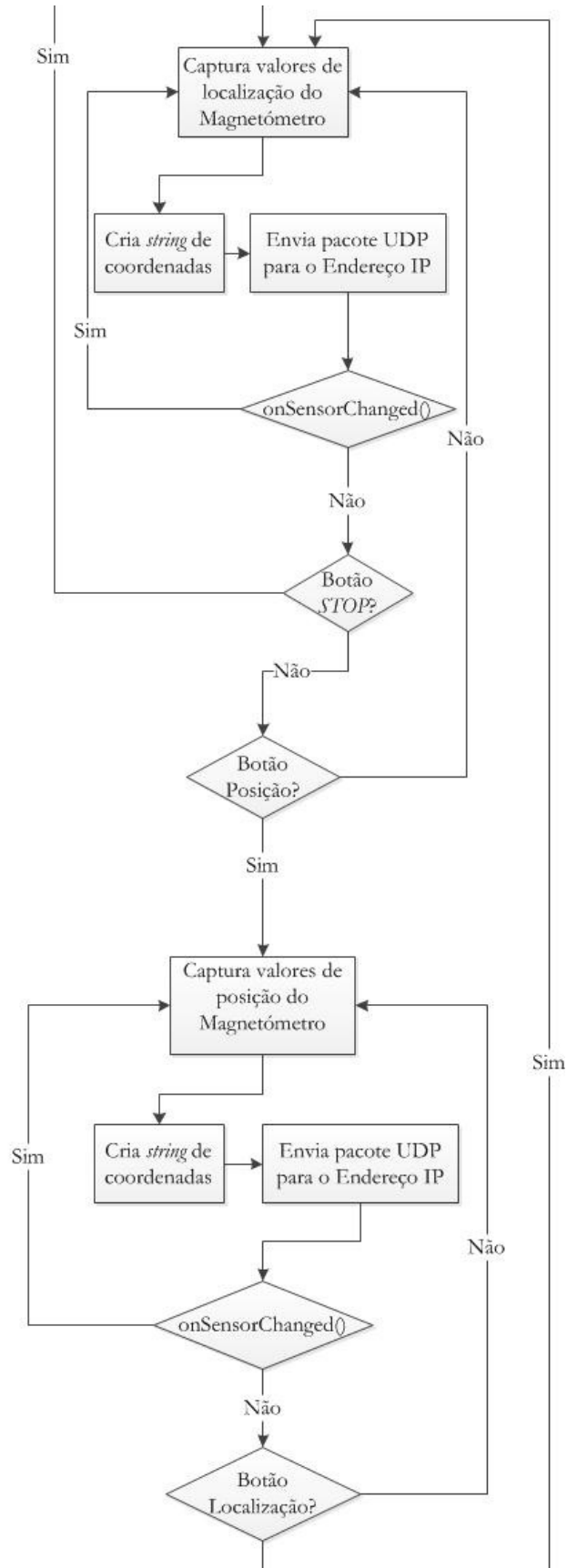


Figura 26: 2ª parte do Diagrama de Fluxo de Dados do *Smartphone*

#### 4.2.2. Microsoft Kinect

Esta componente é responsável pela captura e identificação da mão e das pontas dos dedos. Como o *Microsoft Kinect*, juntamente com o seu SDK, já tem a capacidade de efectuar reconhecimento de imagem, através de várias *libraries* e *samples* importadas para este projecto foi possível proceder ao reconhecimento da mão [18].

O *Microsoft Kinect* está em modo *near* e apenas detecta objectos entre os 50 e 80cm. Ao detectar um objecto, inicia por interpretar o mesmo como se fosse uma mão e desenha um ponto azul no centro deste. Ao verificar que esse objecto se mantém, é calculado o seu centro e é desenhada uma elipse verde que marca aquela zona como sendo a palma da mão. Em seguida é efectuado a cálculo da dimensão da palma da mão, de acordo com as suas coordenadas para as operações de translação e *zoom*.

São delineados os contornos da área que corresponde à mão em geral, que consiste em desenhar um polígono convexo quando o sistema reconhecer mais de 3 pontos convexos perto da palma da mão.

Após estar delineado, com cor branca, o polígono convexo à volta da área a que corresponde a mão em si, o programa vai registar aquela área como a primeira mão detectada e começa a fazer novos cálculos para mostrar a posição actual da mão e dos dedos. Juntamente com o sistema de reconhecimento de imagem, são detectados e desenhados os contornos da mão, a amarelo.

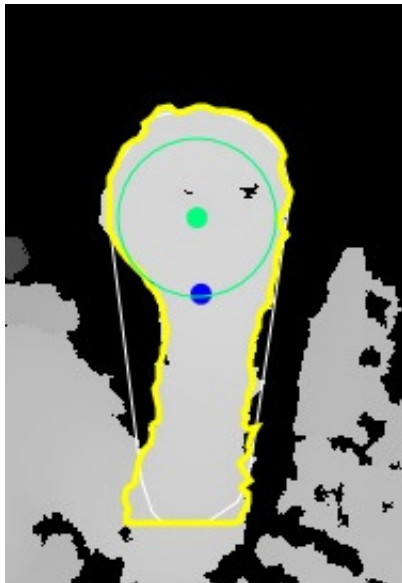


Figura 27: Primeira fase do reconhecimento

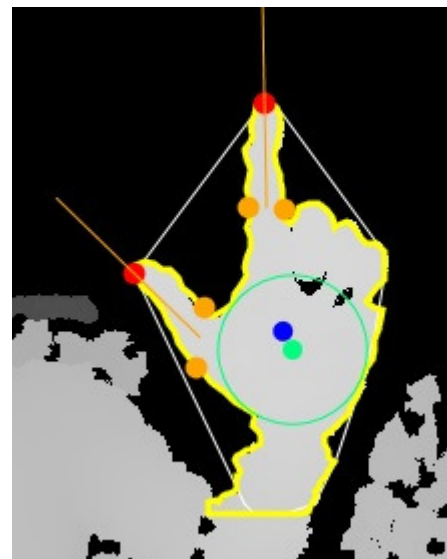


Figura 28: Segunda fase do reconhecimento

Ao serem desenhados todos os contornos, são procurados pontos que estejam a uma distância fixa do delimitador da palma da mão, ao verificar que essa distância existe, começa a interpretar essas zonas como dedos e efectua mais cálculos para marcar dois novos pontos a laranja, por dedo. Esses pontos servem de controlo para o reconhecimento das pontas dos dedos, que são desenhadas a vermelho. Ao verificar que existe uma extremidade no contorno e que a distância entre os pontos laranja é equidistante, o programa assume como a ponta de um dedo.

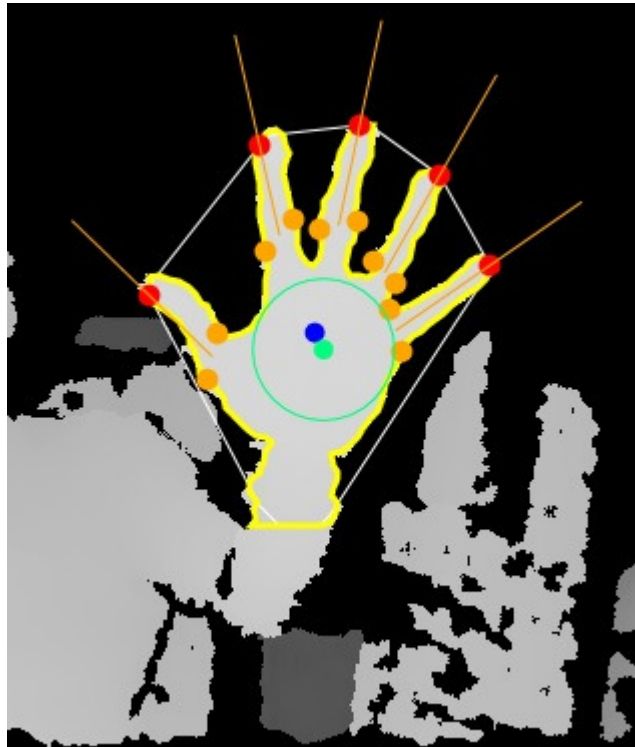


Figura 29: Reconhecimento final da mão e pontas dos dedos

Ao serem analisadas as figuras anteriores, principalmente na figura 27 e 28, pode-se verificar o erro de precisão na câmera de profundidade. Quando a mão está fechada, vêem-se ‘buracos’ na mão. Na figura 29, pode-se verificar que a ponta do dedo indicador não coincide exactamente com o real, mas que se verifica que é o ponto detectado como equidistante dos pontos laranja da base do dedo.

Toda a interacção do *Microsoft Kinect*, neste projecto, foca-se no que foi referido como base e utiliza somente a câmera de profundidade. A câmera RGB não entra na parte de reconhecimento de imagem.

Apenas no menu de calibração está visível este modo de visualização, para que o utilizador possa ajustar a sua posição na cadeira, ver as possíveis interferências de outros objectos e eliminá-los antes de começar a utilizar a aplicação em si.

Após terminada a fase de calibração, apenas a posição das pontas dos dedos é mostrada. São apresentadas circunferências brancas para substituir os pontos vermelhos.

A mão e os dedos detectados são marcados com um identificador. Por ordem de reconhecimento por parte do *Microsoft Kinect*, são numerados. A primeira mão detectada terá o número 1, o primeiro dedo detectado será também o número 1. Dentro da circunferência branca que simula a ponta do dedo, fica escrito o valor 11. Que representa mão 1 e dedo 1.



Ao aparecer outro dedo, será registado como 12; mão 1 e dedo 2. Se o utilizador elevasse a outra mão e fosse apresentado um dedo da mesma, seria registado 21; mão 2 e dedo 1, como ilustrado na figura 30.

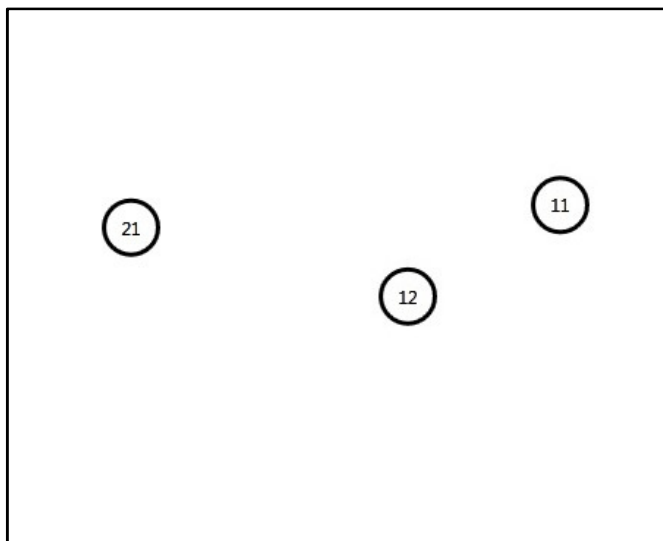


Figura 30: Exemplo de reconhecimento das duas mãos

## Diagrama de fluxo

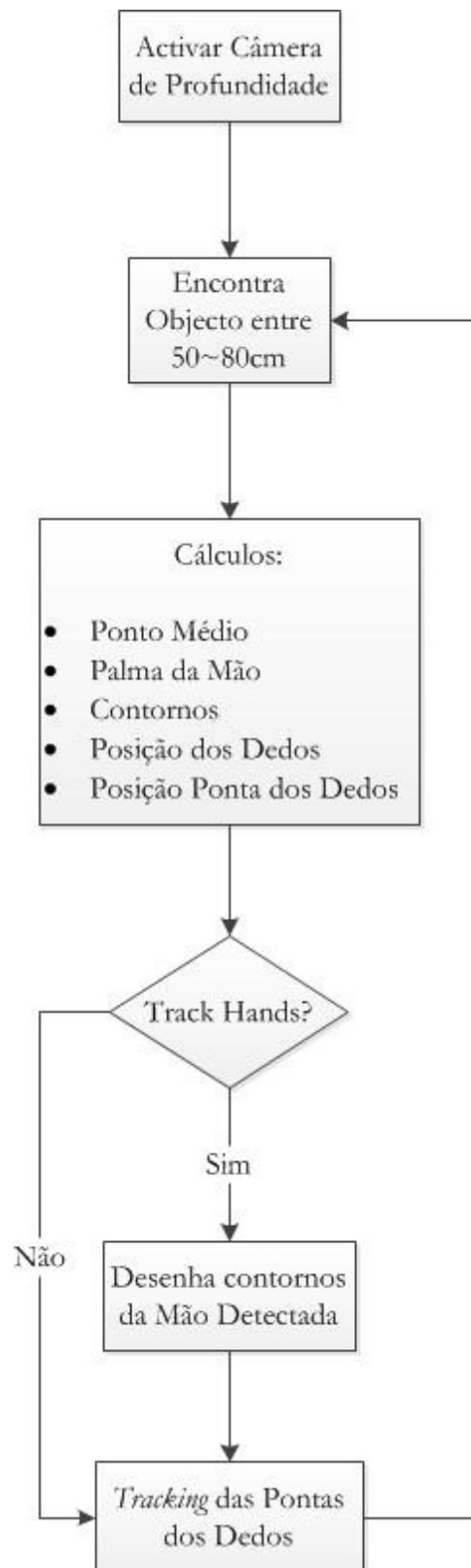


Figura 31: Diagrama de Fluxo de Dados do *Microsoft Kinect*

### 4.2.3. Aplicação Principal

A aplicação desenvolvida em C#, começa com o sistema de calibração do utilizador com o *Microsoft Kinect*. Ao ser activada a opção de *Track Hands*, pode-se visualizar o que o *Microsoft Kinect* está a reconhecer como mãos, ver onde ocorrem interferências e reduzir o ruído que a câmara de profundidade vai captando. Como alguns factores estão para além do nosso controlo, como a luminosidade do ambiente e o contraste entre a cor do vestuário e a cor da parede, foi criada a funcionalidade de se alterar o valor da inclinação do *Microsoft Kinect*. Existe um intervalo, em graus, em que o dispositivo pode ser inclinado para cima e para baixo.

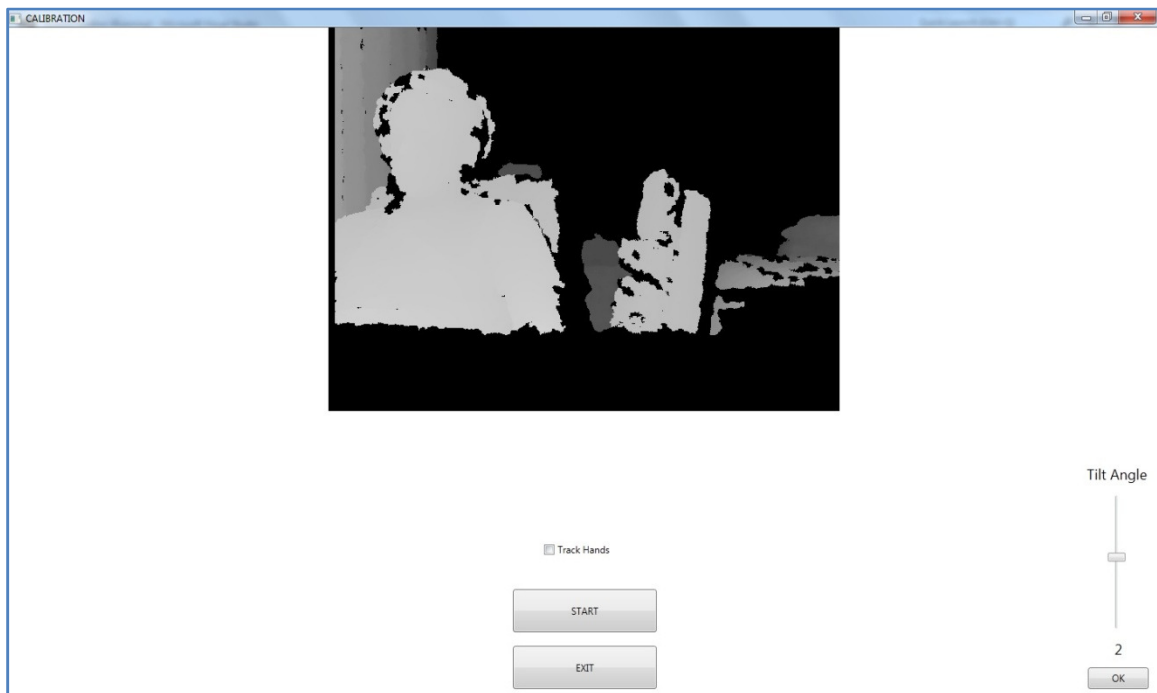


Figura 32: Calibração na aplicação

Na maioria das vezes, para aumentar a profundidade de um local, inclina-se o sensor para cima para se poder aproveitar o espaço que existe até ao tecto. Após terminada a calibração, entra-se na aplicação em si.

Ao lado esquerdo encontra-se o *Google Earth* num *web browser* embutido na aplicação e do lado direito está a *interface* que será de utilização exclusiva da componente do *Microsoft Kinect*.

Tendo em consideração a implementação do *smartphone* e do *Microsoft Kinect*, rapidamente se verifica que é necessário efectuar duas operações em simultâneo. É necessário receber os pacotes UDP do *smartphone* e captar os gestos da parte do *Microsoft Kinect*. Como a câmara de profundidade apresenta melhor desempenho quando está ligada a um computador de *performance* elevada, foi decidido manter todas as operações na *thread* principal da aplicação. Numa *thread* em *background*, ficou a recepção dos dados do magnetómetro.

Esta aproximação é bastante relevante, pois não seria possível estar a receber os dados do magnetómetro, interagir com a API do *Google Earth* e estar a captar gestos em simultâneo, sem a utilização de *threading* neste projecto.

A aplicação está constantemente à escuta de qualquer ligação *wireless* feita no porto especificado na aplicação do *smartphone*. O tamanho do pacote a receber, está restrito ao tamanho da *string* que contem os dados, explicado anteriormente. Ao receber a informação, esta é preparada para ser aplicada na API do *Google Earth*. Visto que a informação recebida chega em graus perante os eixos X, Y e Z, quando se trata da movimentação do utilizador em modo de imersão, é necessário converter essas coordenadas cartesianas em coordenadas polares [5].

```
var camera = ge.getView().copyAsCamera(ge.ALTITUDE_RELATIVE_TO_SEA_FLOOR);
if (pitch < -10)
{
    POS[0] -= 0.00002 * Math.Cos(ANGLE * Math.PI / 180); //andar pra tras
    POS[1] -= 0.00002 * Math.Sin(ANGLE * Math.PI / 180);
    camera.set(POS[0], POS[1], cameraAltitude, ge.ALTITUDE_RELATIVE_TO_SEA_FLOOR, HEADING_ANGLE, TILT_ANGLE, 0);
}
else if (pitch > 10)
{
    POS[0] += 0.00002 * Math.Cos(ANGLE * Math.PI / 180); //andar pra frente
    POS[1] += 0.00002 * Math.Sin(ANGLE * Math.PI / 180);
    camera.set(POS[0], POS[1], cameraAltitude, ge.ALTITUDE_RELATIVE_TO_SEA_FLOOR, HEADING_ANGLE, TILT_ANGLE, 0);
}
else
{
    camera.set(POS[0], POS[1], cameraAltitude, ge.ALTITUDE_RELATIVE_TO_SEA_FLOOR, HEADING_ANGLE, TILT_ANGLE, 0);
}

if (rooll < -10)
{
    POS[1] += 0.00002 * Math.Cos(ANGLE * Math.PI / 180); //direita
    camera.set(POS[0], POS[1], cameraAltitude, ge.ALTITUDE_RELATIVE_TO_SEA_FLOOR, HEADING_ANGLE, TILT_ANGLE, 0);
}
else if (rooll > 10)
{
    POS[1] -= 0.00002 * Math.Cos(ANGLE * Math.PI / 180); //esquerda
    camera.set(POS[0], POS[1], cameraAltitude, ge.ALTITUDE_RELATIVE_TO_SEA_FLOOR, HEADING_ANGLE, TILT_ANGLE, 0);
}
else
{
    camera.set(POS[0], POS[1], cameraAltitude, ge.ALTITUDE_RELATIVE_TO_SEA_FLOOR, HEADING_ANGLE, TILT_ANGLE, 0);
}

ge.getView().setAbstractView(camera);
```

Figura 33: Conversão das coordenadas para o posicionamento

No caso de localização ou observação, não é necessário converter as coordenadas, pois não interfere com o sistema de posicionamento da API do *Google Earth*.

```
var camera = ge.getView().copyAsCamera(ge.ALTITUDE_RELATIVE_TO_SEA_FLOOR);
TILT_ANGLE = tilt;
HEADING_ANGLE = heading;
camera.set(POS[0], POS[1], cameraAltitude, ge.ALTITUDE_RELATIVE_TO_SEA_FLOOR, HEADING_ANGLE, TILT_ANGLE, 0);
ge.getView().setAbstractView(camera);
```

Figura 34: Utilização da API do *Google Earth* para localização

Como se pode verificar na figura 33, a alteração do *array* POS, implica a conversão referida. Na figura 34, como essas variáveis se mantém inalteradas, não é necessária a conversão.

A aplicação divide-se em dois estados. Modo de imersão ligado e desligado.

A informação proveniente do *smartphone* apenas é utilizada quando o modo de imersão está activo. Quando esse modo está desligado, o *smartphone* pode estar também desligado. Ao voltar a entrar no modo de imersão, é só voltar a ligar o *smartphone* que a ligação se restabelece.

Para o controlo da interface, modo de imersão desligado e para efectuar a operação de *zoom* em modo de imersão, é utilizado o *Microsoft Kinect*.

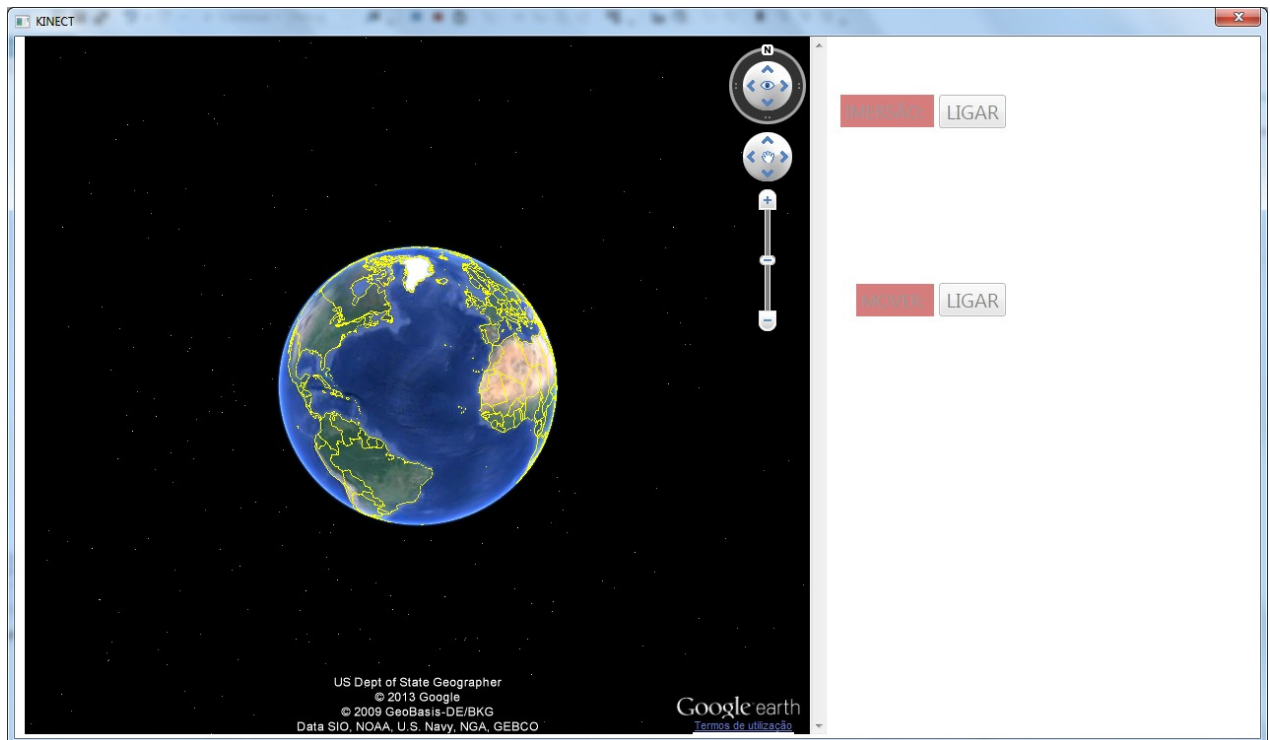


Figura 35: Aplicação no modo inicial

Como se pode observar na figura 35, o *interface* que se encontra do lado direito é para ser controlado apenas pelo *Microsoft Kinect*. Para ajudar na usabilidade, existe um código de cores para cada funcionalidade, para aumentar a percepção do utilizador.

Na fase inicial é possível entrar em modo de imersão ou mover o globo, controlando-o com o dedo indicador da mão que aparecer em primeiro lugar. Ao serem activadas as funcionalidades, um pequeno texto descritivo aparece para ajudar o utilizador a perceber os comandos de controlo gestual.

Ao entrar em modo de imersão, é possível activar a vista 3D dos edifícios e efectuar *zoom in/out* através do controlo gestual. Neste modo, o *smartphone* pode controlar o ambiente a nível de localização e posicionamento.



Figura 36: Aplicação em modo de imersão

Como demonstra a figura 36, o modo de imersão é muito semelhante ao *street view*, com a diferença de existir liberdade de movimento.

Ao sair do modo de imersão, é feito um *zoom out* automático até 4km de altitude em relação ao solo. Ao ser activada a opção de mover, podemos mover o mapa no eixo dos X e Y, não afectando a altitude.

Ao ser centrada a uma nova posição no mapa, essas coordenadas são automaticamente processadas para a API do *Google Earth*, fazendo com que a próxima imersão seja feita naquele preciso local. Como podemos verificar na figura 37 e 38.

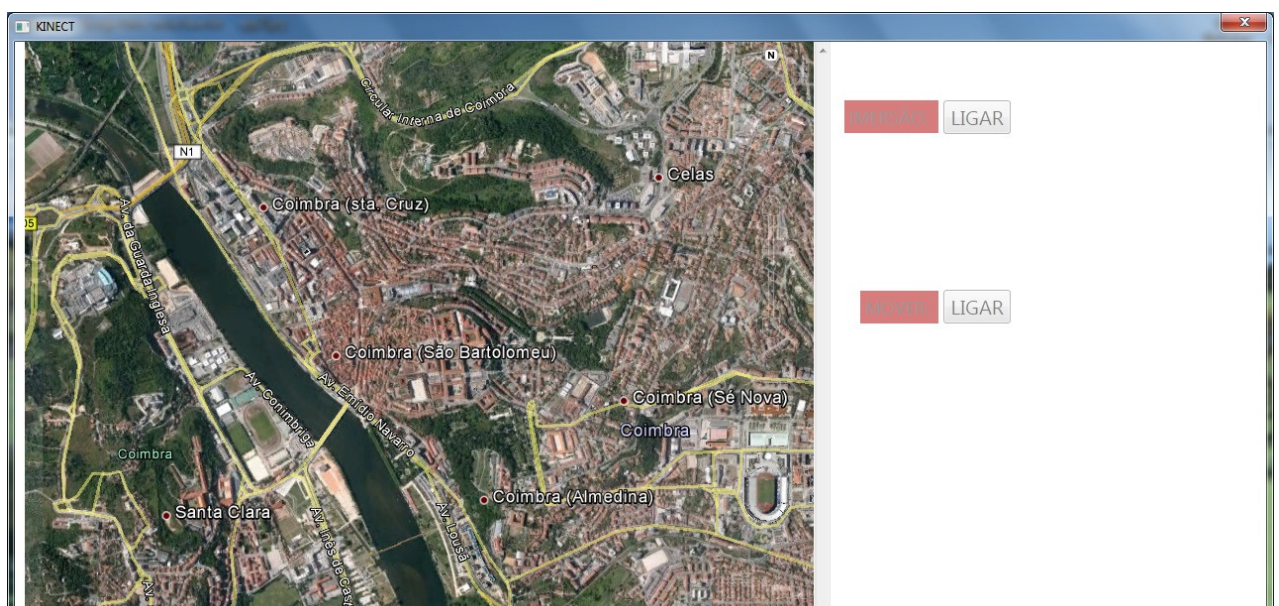


Figura 37: Modo de imersão desligado e centrada nova posição



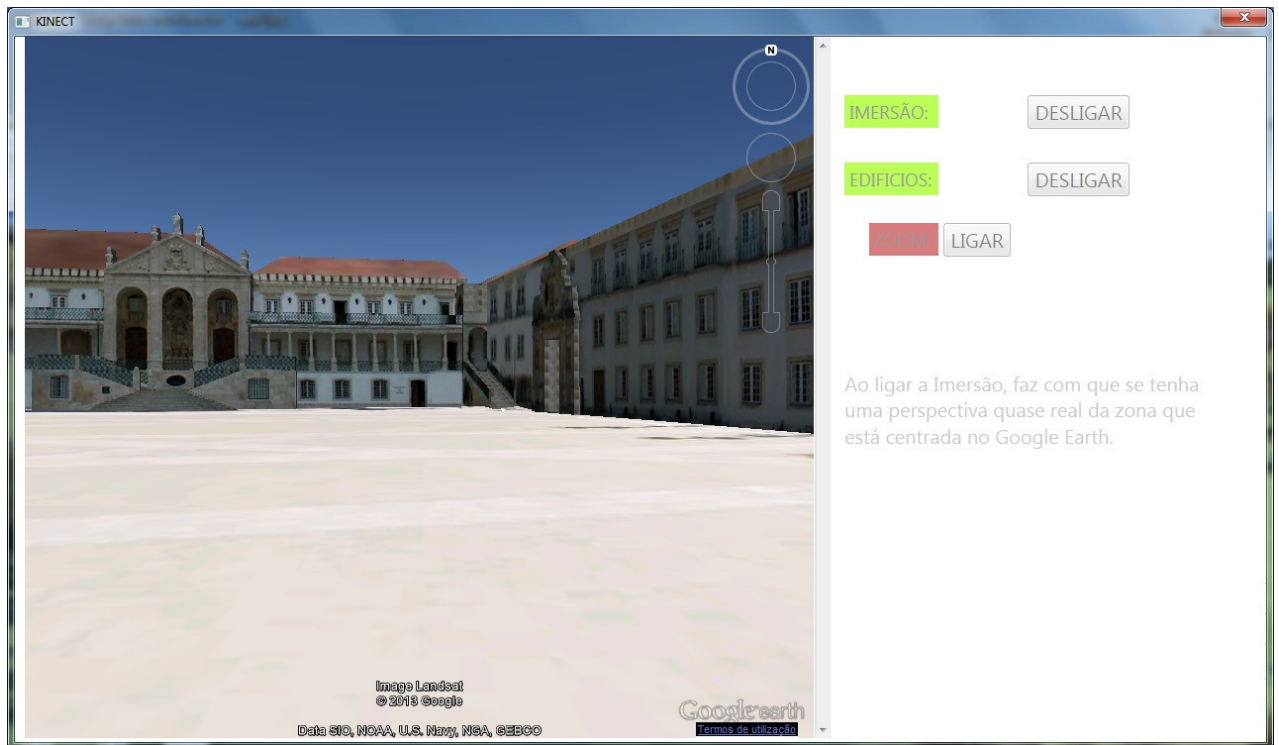


Figura 38: Modo de imersão ligado na nova posição

O interface da aplicação encontra-se ao lado do *web browser*, e as opções disponíveis estão marcadas pelo meio de botões.

O sistema implementado faz com que se escondam automaticamente, as opções que não teriam qualquer efeito no modo em que se encontra a aplicação.

Podemos comparar o modo de imersão na figura 38 e o modo fora de imersão, *default*, na figura 37, em que se confirma que existem opções diferentes disponíveis.

Os botões do interface fazem o *click* com a presença do dedo indicador, na sua respectiva área. Como existem oito botões no total, estão mapeadas oito áreas de interacção.

## Diagrama de fluxo

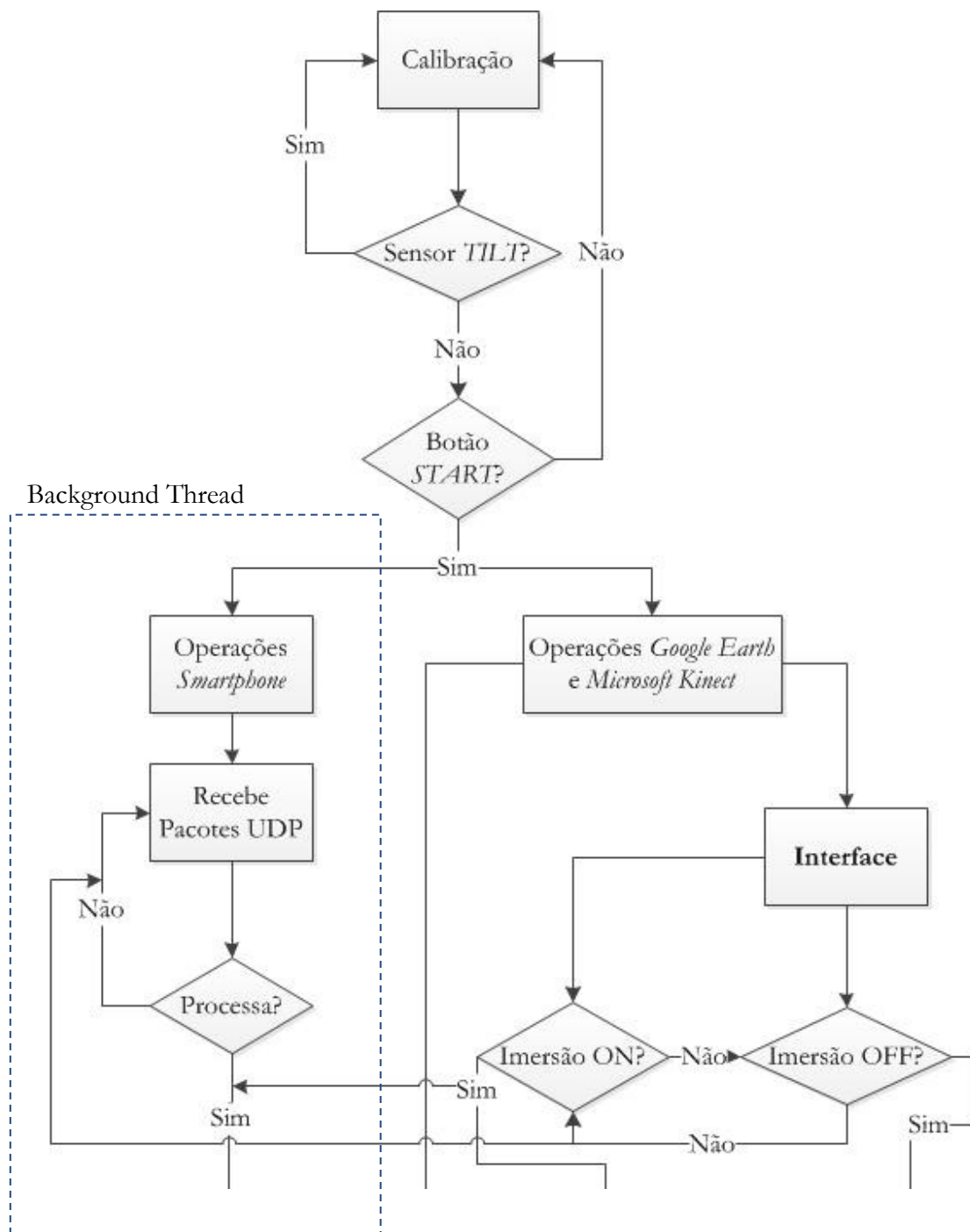


Figura 39: 1ª parte do Diagrama de Fluxo de Dados da Aplicação Principal



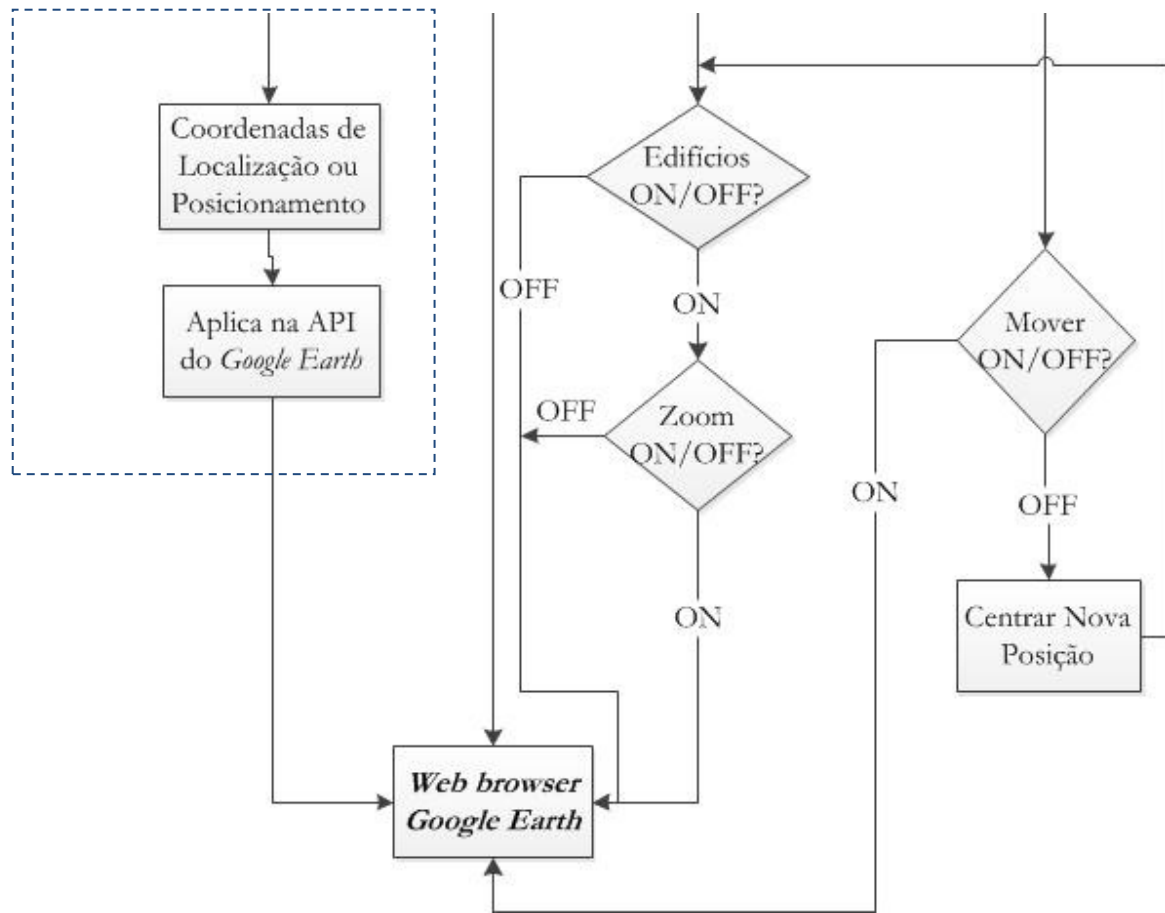


Figura 40: 2ª parte do Diagrama de Fluxo de Dados da Aplicação Principal

## Capítulo 5

### Testes de Usabilidade

Devido a este projecto ter uma componente de interface bastante importante para o funcionamento do mesmo, foi necessário recorrer a testes de usabilidade com vários utilizadores de diferentes áreas para reduzir os problemas de usabilidade que normalmente uma aplicação tem.

Esta aplicação baseia-se nas dez heurísticas de *Nielsen* [16].

Para uma aplicação ter um número reduzido de problemas de usabilidade, é recomendada a utilização de pelo menos cinco utilizadores. Devem elaborar um conjunto de tarefas para calcular os tempos de execução e nível de dificuldade atribuído a cada tarefa.

São também registados e tomados em conta, todos os comentários e sugestões que os utilizadores possam mencionar [6].

A utilização de pelo menos cinco utilizadores baseia-se no gráfico da figura seguinte [22].

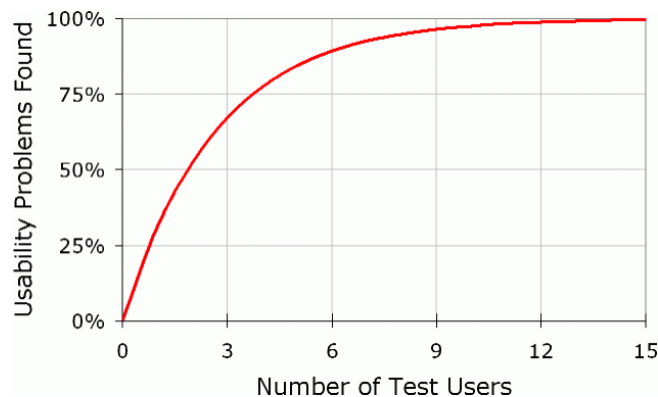


Figura 41: Número mínimo recomendado de utilizadores de teste.

#### 5.1. Descrição dos Testes

Nesta secção estão descritas as tarefas, nível de experiência de cada utilizador, apresentados os resultados dos testes e comentários.

##### 5.1.1. Tarefas

Foram criadas quatro tarefas que cada utilizador efectuou e foi registado o seu tempo de execução.

A primeira tarefa consistia em activar o modo de imersão dentro do estádio de Coimbra.

Na segunda tarefa era pedido para entrar dentro de uma das balizas do estádio, à escolha do utilizador.

Na terceira tarefa, foi requerido sair do modo de imersão, activar o modo de mover, navegar no mapa nas quatro direcções possíveis e voltar a entrar, no modo de imersão num local à escolha do utilizador.

A quarta e última tarefa, implica sair do modo de imersão, centrar a Universidade de Coimbra no mapa, activar o modo de imersão, entrar na Universidade de Coimbra e olhar para a Torre.

### 5.1.2. Experiência dos Utilizadores

A experiência dos utilizadores, foi tomada em conta de acordo com sistemas de interacção com que já tenham tido experiência, como por exemplo, *Nintendo Wii* e/ou *Playstation Move*.

Utilizadores com esse tipo de experiência e utilizadores de um computador, foram considerados como 'experientes'. Sem esses requisitos, mas sendo utilizadores assíduos de um computador, foram considerados como 'intermédios'. Se for alguém que esporadicamente utiliza um computador, é considerado como 'pouco experiente'.

Dos seis utilizadores, quatro deles pertencem ao nível 'experiente', um utilizador é nível 'intermédio' e um tem um nível 'pouco experiente'.

### 5.1.3. Tempo de Execução

Como se verifica, à medida que a tarefa ia sendo mais complexa, em média, mais tempo se demorava. Existe uma variação elevada nos resultados da quarta tarefa, que será explicada na conclusão dos testes.

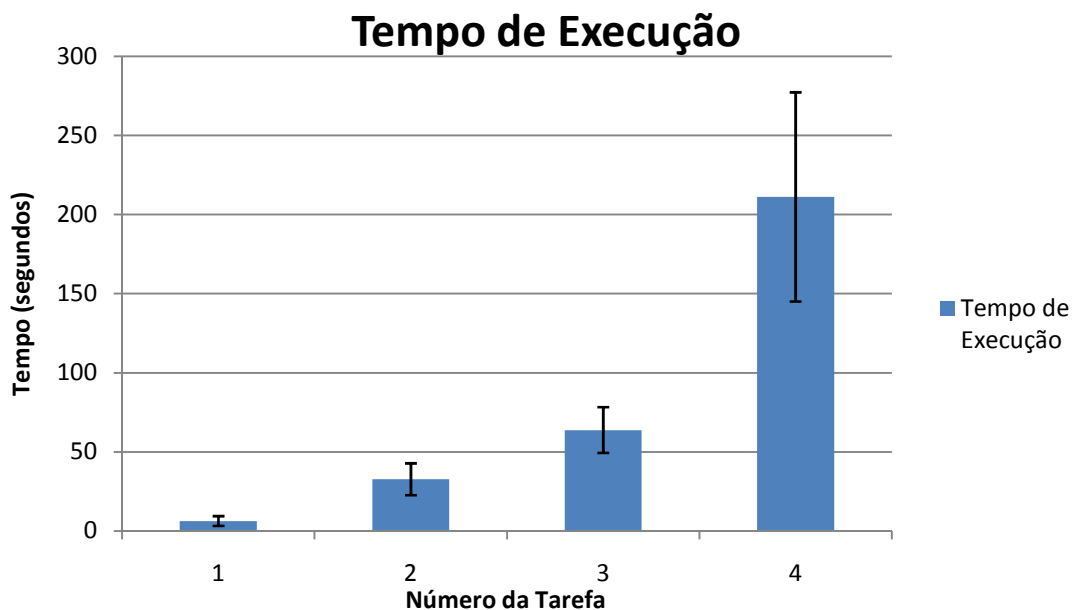


Figura 42: Gráfico da média de tempos de execução

#### 5.1.4. Níveis de Dificuldade

As tarefas foram concluídas com sucesso, pelos seis utilizadores.

Dentro do sucesso, categoriza-se o nível Fácil, Médio e Difícil. Para a construção do gráfico, fácil corresponde o valor 1, médio o valor 2 e difícil o valor 3.

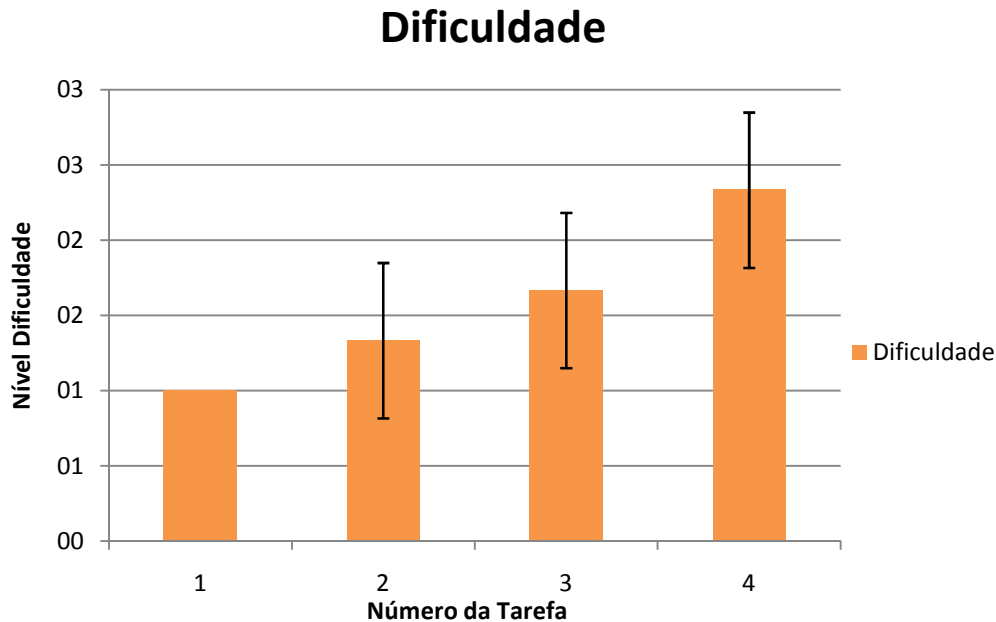


Figura 43: Gráfico da média dos valores da dificuldade encontrada

#### 5.1.5. Comentários

Os comentários dos utilizadores, ajudaram na remoção de erros de interface e até mesmo a nível de sugestões.

O comentário dominante, referido por todos os utilizadores, foi: "ser necessário um tempo de habituação".

"O *Microsoft Kinect* é um dispositivo insuficiente para este tipo de ambientes, pois está sempre a tremer. Este dispositivo não serve para substituir a precisão manual do rato, apenas serve para ambientes altamente interactivos."

"Alguma dificuldade em estabelecer o indicador da mão como ponteiro do rato. Desconforto no braço."

"Cor das letras do interface muito claras, o que leva a um esforço ocular maior e desnecessário."

## 5.2 Conclusão dos Testes

De acordo com os comentários obtidos, a aplicação foi sendo corrigida.

Relativo ao tempo de habituação do interface, cerca de 20 minutos é o tempo suficiente para um utilizador ter uma percepção aceitável da aplicação.

As limitações do *Microsoft Kinect*, mais precisamente da câmara de profundidade, deixaram um utilizador insatisfeito com a sua precisão, mas com o auxílio do *smartphone*, dá para compensar esse erro.

Para eliminar o desconforto do braço do utilizador, foi criada a zona de alterar o *Tilt* do sensor durante a calibração.

Dois utilizadores experimentaram a aplicação numa fase intermédia. Esses valores não foram aqui registados, mas foram tomadas em consideração as suas sugestões. Uma das sugestões foi a mudança de cor do fundo das letras. Vermelho para desligado, verde para ligado. Esta sugestão provou ser eficaz.

A outra sugestão foi ter o cursor do rato a perseguir o cursor virtual que a componente do *Microsoft Kinect* cria. Essa sugestão aumentou a precisão durante a operação de navegar fora do modo de imersão.

O facto de ter o cursor do rato a seguir o cursor virtual, permitiu poder resolver o problema relacionado com o último comentário; mudar a cor das letras para preto e retirar as transparências nos botões.

Um utilizador é esguerdino, dessa forma foi também possível comprovar que esse factor não influencia os resultados, pois conseguiu completar algumas tarefas mais depressa que os utilizadores dextros.

Um aspecto curioso na realização destes testes, foi o facto do utilizador com menor experiência, ter classificado como fácil o controlo em modo de imersão, em comparação com o controlo do rato. O que pode provar que alguns utilizadores estão habituados de tal forma ao tradicional, que perante a inovação criam alguma resistência.

O resultado a nível do tempo da quarta tarefa, deve-se a vários factores. Uns utilizadores sentiram-se enganados com o *Google Earth*, pois tinham como referência a Universidade de Coimbra perto da Sé Nova e a legenda deste não coincide. Outros, simplesmente têm um sentido de orientação fraco e no modo de imersão perdem um pouco a noção de onde se encontram.

De um modo geral, todos os testes tiveram sucesso e os utilizadores acharam interessante o modo de interacção com a aplicação.

## Capítulo 6

### Conclusões

No início do ano, o projecto começou a ser delineado a nível de objectivos e começou a ganhar forma, de modo a que, fosse continuado com motivação e empenho de acordo com o planeamento previsto.

O projecto encontrava-se definido para prosseguir, embora tivesse havido alguma indefinição por parte do magnetómetro, que foi resolvido posteriormente, (referenciado no capítulo 4, em Análise Complementar e Implementação).

Como foi referido várias vezes ao longo deste documento, o projecto consiste em duas componentes; o *smartphone* e o *Microsoft Kinect*. A componente que necessitou de uma maior atenção foi o *Microsoft Kinect*. Embora o *smartphone* também tivesse um impacto elevado.

De um modo geral, ao longo do desenvolvimento deste projecto, todos os objectivos foram atingidos. Como se trata de um projecto cujos resultados são visualmente concretos, foi muito interessante e motivador o decurso do seu desenvolvimento.

Foram adquiridas novas capacidades de raciocínio e de perspectiva, necessárias para as várias abordagens necessárias na implementação das várias funcionalidades.

#### 6.1. Principais Obstáculos

No decorrer deste projecto, o maior obstáculo encontrado foi o tentar manter o planeamento definido, o que por vezes se tornou impossível e teve de ser redobrado o esforço para voltar ao ponto certo.

A nível do *smartphone*, a indecisão sobre como enviar os dados para o computador, causou um gasto de tempo considerável, na constante tentativa de enviar os dados via *bluetooth*.

Perante o *Microsoft Kinect*, teve de existir uma persistência muito grande, devido às falhas de hardware existentes. O dispositivo, por vezes, fica muito lento e precisa de ser desligado um pouco porque a ventilação gerada por ele próprio, é insuficiente perante a utilização durante um dia completo. A câmara de profundidade tem uma precisão que deixa muito a desejar, o que foi causa de comentários depreciativos sobre o sensor em si, durante os testes de usabilidade.

## **6.2. Trabalho Futuro**

Nos tempos que correm, sente-se cada vez mais o efeito da crise económica, e como tal, muitas pessoas são impossibilitadas de viajar. Com este projecto, obtém-se a ideia de podermos andar de modo virtual e livre, dentro de um local de sonho pretendido pelo utilizador, o que é visto como algo bastante agradável e apelativo.

Também poderá ter aplicações a nível de ensino, como por exemplo, apoio a aulas de Geografia e, possivelmente História.

Este projecto tem potencial para ser continuado. Nesta fase está apto para cumprir o seu objectivo principal.

Apesar do *Microsoft Kinect* estar a começar a entrar em desuso e a ser ultrapassado por outras tecnologias alternativas, para alguns problemas menos complexos ainda se apresenta como uma boa solução.

Nota:

Este documento não foi redigido consoante o Acordo Ortográfico da Língua Portuguesa (1990).

## Referências

- [1] Android Developer - Position Sensors. Disponível em [http://developer.android.com/guide/topics/sensors/sensors\\_position.html](http://developer.android.com/guide/topics/sensors/sensors_position.html) (2011); Acedido em 04/03/2013.
- [2] Anthony, S. ExtremeTech, '*The Leap: Gesture control like Kinect, but cheaper and much higher resolution*'. Acedido em 10/01/2013, de ExtremeTech: <http://www.extremetech.com/computing/129638-the-leap-gesture-control-like-kinect-but-cheaper-and-much-higher-resolution>
- [3] CogniMem Technologies, Inc. Disponível em <http://www.cognimem.com/products/chips-and-modules/CM1K-Chip/index.html> (2011); Acedido em 09/10/2012.
- [4] CogniMem Technologies, Inc., *Kinect Finger Recognition For Games*, Acedido em 09/10/2012 de Youtube: <http://www.youtube.com/watch?v=NqjopQmqWAE>
- [5] EngineeringToolbox.com - Convert between Cartesian and Polar coordinates. Disponível em [http://www.engineeringtoolbox.com/converting-cartesian-polar-coordinates-d\\_1347.html](http://www.engineeringtoolbox.com/converting-cartesian-polar-coordinates-d_1347.html) (2011); Acedido em 05/06/2013.
- [6] Gomes, I. *Testes de Usabilidade: uma metodologia rápida e fácil*. Acedido em 12/06/2013 de ivogomes: <http://www.ivogomes.com/blog/testes-de-usabilidade-uma-metodologia-rapida-e-facil/>.
- [7] Hwang, S. and Ryu, M. and Yang, Y. and Lee, N. '*Fall detection with Three-Axis Accelerometer and Magnetometer in a Smartphone*'. Acedido em 11/12/2012, de onlinepresent.org: [http://onlinepresent.org/proceedings/vol6\\_2012/10.pdf](http://onlinepresent.org/proceedings/vol6_2012/10.pdf)
- [8] Kinect for Windows. Disponível em <http://www.microsoft.com/en-us/kinectforwindows/> (2012); Acedido em 07/11/2012.
- [9] KinectHacks, Disponível em <http://www.kinecthacks.com>(2011); Acedido em 07/11/2012.



[10] Kirkpatrick, M. *Could Kinect Control Your Internet of Things?*, Acedido em 28/09/2012, de Readwrite:

[http://readwrite.com/2010/11/04/could\\_kinect\\_control\\_your\\_internet\\_of\\_things](http://readwrite.com/2010/11/04/could_kinect_control_your_internet_of_things)

[11] Lázaro y Torres, M. A cidade, um laboratório para a educação geográfica, "*El impulso de las competencias espaciales a través del trabajo de campo virtual en las ciudades*", Conferência 2013.

[12] LEAP Motion. Disponível em <http://www.leapmotion.com> (2012); Acedido em 11/12/2012.

[13] Magnetometers and Accelerometer and Gyros. Disponível em <https://forum.sparkfun.com/viewtopic.php?f=14&t=24226> (2010); Acedido em 08/01/2013

[14] NaturalPoint - TrackIR, SmartNav, OptiTrack. Disponível em <http://www.naturalpoint.com> (2011); Acedido em 11/12/2012.

[15] OpenNI - The standard framework for 3D sensing. Disponível em <http://www.openni.org> (2010); Acedido em 07/11/2012.

[16] Rocha, F. *Heurísticas de Nielsen*. Acedido em 22/05/2013, de filiperocha: <http://www.filiperocha.com/blog/heuristicas-de-nielsen/>.

[17] Schapman, J. DomController, *Control Anything from Anywhere*. Acedido em 09/10/2012 de DomController: <http://www.domcontroller.com/en/>

[18] Stefan. Candescant, *My personal blog: Currently focusing on AI methods applied to Kinect sensor data*. Acedido em 9/5/2013, de <http://blog.candescent.ch/>

[19] Tip Tep Researches. Disponível em <http://tiptep.com/index.php/research> (2013); Acedido em 10/2012

[20] Todd. Autoponics, *Switching from Kinect to Asus Xtion Pro Live*. Acedido em 03/12/2012, de University of Colorado: <http://autoponics.org/?p=160>

[21] VisionTrak. Disponível em [http://polhemus.com/?page=Eye\\_VisionTrak](http://polhemus.com/?page=Eye_VisionTrak) (2006); Acedido em 11/12/2012.

[22] Why You Only Need to Test with 5 Users. Disponível em <http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>(2000);  
Acedido em 17/06/2013

[23] Yoon, A. *Kinect update could greatly improve performance, report suggests*. Acedido em 24/10/2012, de Joystiq: <http://www.joystiq.com/2010/12/19/kinect-update-could-greatly-improve-performance-report-suggests/>