

Sistema de Orientação para o Interior de Edifícios - Solução Visual de Localização

João Sena Marcos
joaosenamarcos@sapo.pt

José Miguel Salles Dias
Miguel.dias@iscte.pt

António Calado Lopes
Antonio.calado@iscte.pt

ADETTI, Edifício ISCTE, Av. Das Forças Armadas, 1600-082 Lisbon, Portugal,
<http://www.adetti.iscte.pt>

Abstract

Neste artigo apresentamos um sistema para computação móvel em cenários interiores, baseado em PDA, que consegue resolver com técnicas de visão por computador, a localização do utilizador e possibilita a sua orientação dentro de um edifício através de marcas dispostas em locais estratégicos, previamente identificados. As marcas, capturadas com a câmara de vídeo integrada no PDA, são conhecidas pelo sistema, que tendo identificado o local, faz aparecer no ecrã uma seta virtual com a direcção do caminho que se deve seguir, para chegar da forma mais rápida ao destino inicialmente seleccionado. É utilizada a biblioteca de Realidade Aumentada 'ARToolkit' adaptada à plataforma PocketPC, para as operações de reconhecimento de marcas e cálculo da pose da câmara real relativamente à marca. Para o desenho 3D é usada a biblioteca open-source 'Klimt'. O sistema é apresentado nos seus componentes principais, com ênfase no algoritmo de descoberta do caminho mais curto de Dijkstra e na disponibilização de Realidade Aumentada em ambiente de computação móvel com escasos recursos computacionais e com reutilização de marcas de visuais.

Keywords

Realidade Aumentada, Computação Móvel, PDA, Localização Visual, Marcas Visuais, Dijkstra, AR Toolkit

1. INTRODUÇÃO

Neste trabalho, propusemo-nos desenvolver um sistema de localização e orientação, utilizável no interior de edifícios em geral e, no ISCTE, em particular, que funcionasse num dispositivo de computação móvel simples, do tipo PDA (*Personal Digital Assistant*). Não podendo recorrer ao GPS (*Global Positioning System*) por não haver linha de vista para os satélites, nem a *pseudolites* (que correspondem à versão Terra dos satélites GPS, os quais foram descartados neste trabalho devido ao seu custo proibitivo), apresentavam-se-nos duas soluções alternativas: a colocação de uma multiplicidade de sensores no edifício, que comunicassem com o sistema via emissores/receptores infravermelhos, *bluetooth*, rede WI/FI, etc., ou a colocação na parede de marcas visuais conhecidas do sistema, que teriam que ser filmadas e adquiridas pelo dispositivo móvel do sistema, para o respectivo reconhecimento do local. Escolhemos a segunda solução, essencialmente por ser menos onerosa e, ainda, por permitir mais flexibilidade para o fim em vista. De facto, a tecnologia de seguimento da pose (posição e orientação) de um utilizador em movimento, por tecnologia de infra-

vermelhos, está limitada a uso em linha de vista entre dispositivos activos, o que implicaria a disponibilização de um número elevado dos referidos dispositivos de seguimento em áreas consideráveis do edifício (com o consequente aumento dos custos envolvidos). Por outro lado, as outras tecnologias de seguimento referidas apresentam os seus inconvenientes próprios: *bluetooth* está igualmente limitado geograficamente a pequenas áreas e, as redes WI/FI, ainda não se caracterizam pela ubiquidade da disponibilização do respectivo serviço de comunicação nos edifícios em geral e no ISCTE, em particular, onde os testes técnicos do sistema têm vindo a ser executados.

As marcas visuais usadas são identificadas pelo nível de Realidade Aumentada do nosso sistema, executada em modo local no dispositivo PDA e fazem parte de uma solução GPL de Realidade Aumentada, o ARToolkit, desenvolvida por Mark Billinghurst e Hirokazu Kato [Billinghurst03] [Kato03].

Para o desenvolvimento deste sistema, foram de grande utilidade os trabalhos publicados de duas equipas lideradas por Dieter Schmalstieg, da *Vienna University of Technology, Áustria* [Kalkusch02, Wagner03], o último

dos quais diz respeito à disponibilização do *ARToolkit* para ambientes de computação móvel em PDA (em modo local).

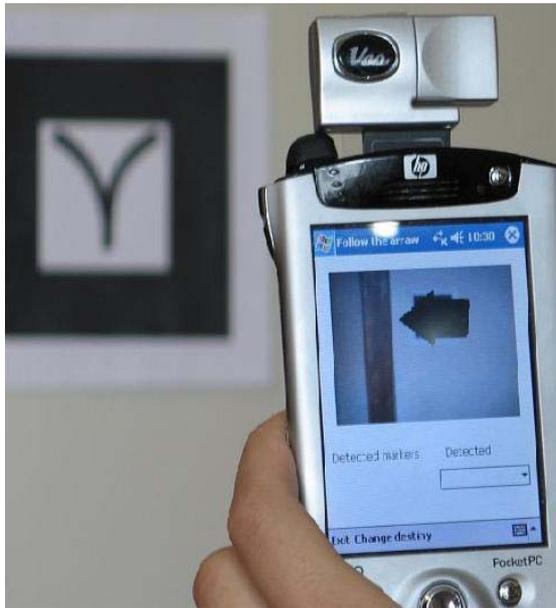


Figura 1. O Sistema reconhece a marca colocada num nó do percurso no interior do edifício e, conhecendo o caminho mais curto para chegar ao destino, sobrepõe uma seta na marca, a qual indica que o caminho devido, segue para a esquerda.

O artigo está organizado da seguinte forma: na secção 2, descrevemos a arquitectura modular e o funcionamento global do sistema. Na secção 3, abordamos o esquema de distribuição de marcas pelo interior do edifício, bem como a problemática da reutilização das marcas em ambientes computacionais de recursos limitados. A secção 5, foca a interface de utilização do sistema, bem como o equipamento utilizado. Finalmente, na secção 5, extraímos algumas conclusões e apontamos algumas direcções para trabalho futuro.

2. FUNCIONAMENTO GLOBAL DO SISTEMA

O nosso sistema desenvolveu-se a partir de um requisito fundamental: a necessidade de indicar ao utilizador o caminho a seguir para um dado destino, no interior de um edifício. A satisfação deste requisito, exige, em primeiro lugar, o conhecimento da localização do mesmo. Depois, terá que descobrir qual o caminho mais curto desde o local actual até ao destino, com base em informação geométrica e topológica do respectivo edifício (disponível a partir da informação em planta) e, finalmente, indicar a sua direcção.

No esquema que se mostra na Figura 2, podemos observar os diferentes módulos do sistema e o respectivo fluxo de informação.

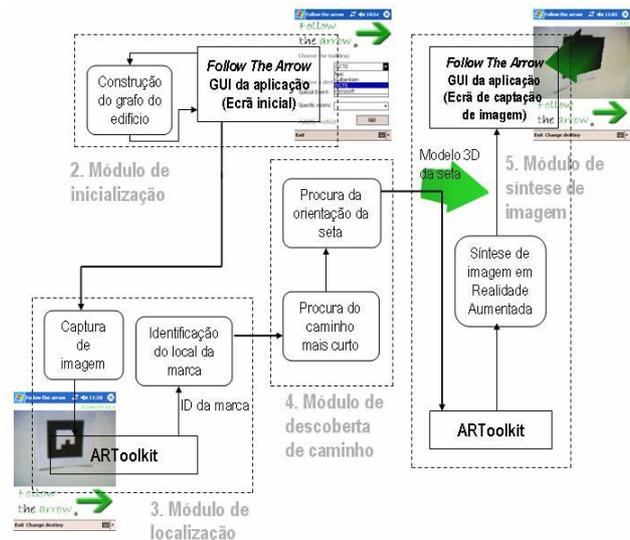


Figura 2. Módulos do sistema e fluxo de operações

2.1 MÓDULO DE INICIALIZAÇÃO

O módulo de inicialização do sistema, constrói um grafo a partir da informação geométrica e topológica do edifício seleccionada pelo utilizador, a qual está guardada num ficheiro de entrada 'adjacentVertices.txt'.

Uma forma prática que encontramos para representar a geometria e topologia da planta de edifícios foi a construção de grafos, em que os nós do grafo são os pontos de interesse nas planta e o custo entre nós, a distância estimada ou arbitrada entre pontos adjacentes. Cada grafo corresponde a uma ou mais plantas do edifício e é construído a partir de um ficheiro de pontos de interesse, onde estão registadas as distâncias e as relações espaciais entre pontos adjacentes.

Os pontos de interesse tanto podem ser divisões do edifício como locais onde é necessário escolher uma direcção a seguir, em virtude de serem zonas de quebra de linha de vista de um utilizador em andamento, relativamente ao percurso a seguir. Nesses pontos de interesse estarão colocadas marcas como a que se vê na Figura 1. A associação entre os identificadores das marcas (IDs) e os locais onde elas se encontram é guardada num ficheiro denominado 'assignedMarkers.txt'.

A estrutura de uma ou várias plantas de um edifício representada como um grafo, torna-se útil na altura de descobrir o caminho mais "curto" entre dois pontos da mesma. De facto, existem alguns algoritmos clássicos de pesquisa do caminho mais curto em grafos. Um deles, o algoritmo de Dijkstra [Cormen90], foi o utilizado no nosso sistema.

Para mais fácil edição e manutenção dos ficheiros acima referidos, desenvolveu-se um editor gráfico simples, denominado *Adjacent Points Editor – for Buildings Representation*.

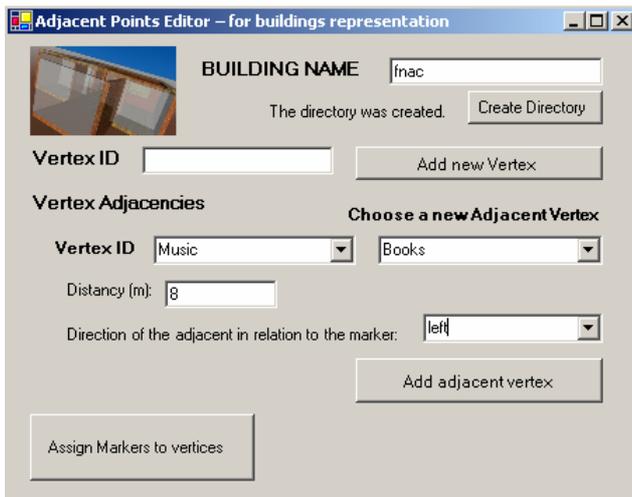


Figura 3. Exemplo de execução da aplicação ‘Adjacent Points Editor – for building representation’

No módulo de inicialização constrói-se assim a estrutura de dados que representa o grafo que representa o edifício. As duas formas mais comuns de representar computacionalmente um grafo são, ou como uma matriz de nós adjacentes, ou como um conjunto de listas de nós adjacentes. Seja o grafo $G = (V, L)$, em que V são os vértices (nós) do grafo e L o conjunto de ligações entre vértices. $|V|$ é o número de vértices e $|L|$ o número de ligações entre vértices. A representação do grafo como uma matriz de nós adjacentes é normalmente usada quando o grafo é *denso* - $|L|$ está próximo de $|V|^2$ - ou quando se pretende saber de uma forma rápida se existe uma ligação entre quaisquer dois vértices. Contudo, é mais usada a representação de um grafo como um conjunto de listas de nós adjacentes, pois consegue-se representar grafos *esparsos* - aqueles nos quais $|L|$ é muito menor que $|V|^2$ - ocupando muito menos memória¹. É este o tipo de grafo que representa as ligações entre zonas de um edifício. Por esse motivo a representação de grafo que usámos foi a última que se referiu.

A representação do grafo $G = (V, L)$ como lista de adjacências consiste numa matriz de *Adjacências* com $|V|$ listas, uma para cada vértice em V . Para cada $u \in V$, a lista de adjacências *Adjacências*[u] contém todos os vértices v para os quais há uma ligação $(u, v) \in L$, isto é todos os vértices adjacentes a u em G .¹

No exemplo de utilização da Figura 3, acima, na preparação da planta do “espaço *fnac*”, foi inserida a distância (8 metros) entre a zona da Música (Music) e a zona dos livros (Books) e está a ser inserida a relação espacial entre as duas zonas: a zona dos livros fica à esquerda (left) da zona musical. Obviamente esta relação espacial tem haver com o local onde se vai colocar a marca na zona da música.

¹ Cfr. [Cormen90], capítulo 23.1 Representation of graphs.

2.2 MÓDULO DE LOCALIZAÇÃO - Uma Solução Visual de Localização

A cada nó do grafo que representa um edifício, está associada uma marca visual (*fiducial marker*) que será interpretada e reconhecida pelo nível de Realidade Aumentada do nosso sistema.

Reconhecida uma marca, o sistema descobre o local onde o utilizador se encontra. Este sistema de localização é consideravelmente menos oneroso, quando comparado com o recurso a sensores activos, pois apenas são necessárias marcas impressas e uma câmara comercial integrada no PDA.

O módulo de reconhecimento de marcas, estimação da pose (posição e orientação) da marca relativamente à câmara de vídeo e de desenho 3D em Realidade Aumentada, corresponde a uma *dll* realizada em *Embedded C++*. Esta biblioteca dinâmica, a que chamámos *ARModule.dll*, encapsula uma versão modificada do *Artoolkit* especificamente adaptada para plataformas móveis (PocketPC e Symbian) [Wagner04]. Para o desenho 3D foi utilizada a biblioteca GPL *Klimt*, igualmente criada por Daniel Wagner [Klimt03]. A comunicação entre a aplicação principal e esta *dll* é feita através de uma técnica da plataforma .NET, o *Interop*. Resolvemos dividir o sistema nestas duas componentes, uma vez que desta forma conseguimos aliar a facilidade de desenvolvimento e programação característicos da linguagem C#, ao superior desempenho do *Embedded C++*.

O *ARToolkit* constituiu, pois, a base da nossa solução visual de localização do utilizador. O seu funcionamento processa-se da seguinte forma [Billinghurst03]: Primeiro, a imagem vídeo captada pela câmara é convertida numa imagem binária (a preto e branco), com base em valores limiares de luminância. Nesta imagem são procuradas regiões quadradas. O *ARToolkit* encontra todos os quadrados na imagem binária, muitos dos quais não são as marcas de localização. Para cada quadrado é capturado o padrão nele desenhado, produzida uma assinatura DCT que é comparada com os padrões conhecidos do sistema. Se houver uma correspondência, então o *ARToolkit* encontrou uma das marcas de localização. Esta biblioteca determina ainda a pose da câmara relativamente à marca. Envia então o ID unívoco da marca encontrada a uma função que identifica o local a que pertence a marca, quer seja por associação directa ou pelo conhecimento que tem do percurso já percorrido, conforme o esquema de distribuição de marcas que tenha sido usado, como se explicará na secção 3.

2.3 MÓDULO DE DESCOBERTA DO CAMINHO

Como já fizera Kalkusch no seu sistema de navegação para interiores, o *SignPost* [Kalkusch02], usámos igualmente o algoritmo de Dijkstra para descobrir, em tempo real, o caminho mais “curto” entre os locais de origem e destino do utilizador.

Não foi utilizado o algoritmo *breadth-first search*, um dos mais simples algoritmos de pesquisa em grafos, pois este algoritmo trabalha em grafos sem *pesos*, isto é grafos nos quais se pode considerar que cada ligação tem *peso*

(*custo*) unitário². No caso dos grafos que representam edifícios todas as ligações têm um *custo*: a sua distância (real ou arbitrada).

Necessitávamos de um algoritmo que, dado um nó de origem e outro de destino, encontrasse a sequência de nós cuja soma dos *custos* das ligações que os unisse fosse a menor. Este é um problema do tipo *single-source shortest path problem*, que tem como solução clássica, para o caso em que nenhuma ligação tenha *custo* negativo, o algoritmo de Dijkstra. O algoritmo de Bellman-Ford é usado no caso mais geral em que as ligações podem ter *custos* negativos³. Os procedimentos do algoritmo de Dijkstra descritos no livro *Introduction to Algorithms* [Cormen90], foram por nós concretizados na plataforma .NET, em C#, como aliás, grande parte do nosso sistema.

Para testar o algoritmo, desenvolvemos uma aplicação que constrói o grafo com os nós que vai importar a um ficheiro. A aplicação desenha os nós em posições relativas, que têm que ser especificadas nos ficheiros de nós, e permite ao utilizador escolher um nó de origem e outro de destino a fim de se procurar a sequência de nós com *custo* mais baixo entre esses dois pontos, correspondente ao caminho de “*custo*” mais baixo, ou caminho mais “curto”. Na Figura 4, podemos observar um exemplo de execução do algoritmo de Dijkstra, a determinar o caminho mais “curto” entre dois nós (b e f) de um grafo.

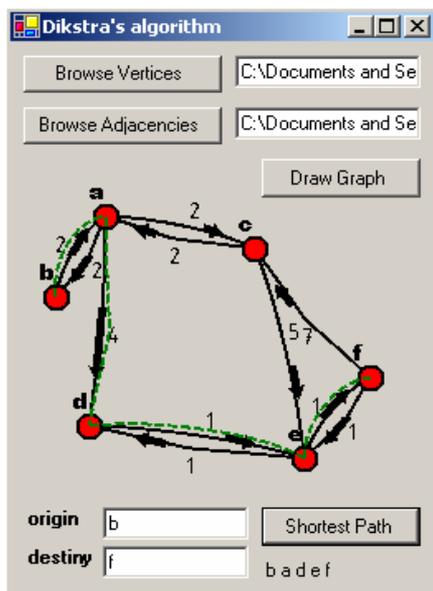


Figura 4. Aplicação de teste do algoritmo de Dijkstra.

2.4 MÓDULO DE SÍNTESE DE IMAGEM

Uma vez descoberto o caminho mais curto para o destino, o módulo de procura do caminho vai descobrir qual a relação espacial entre o *nó* onde o utilizador actualmente se encontra e o próximo *nó* desse caminho. Sabida essa

² Cfr. [Cormen90], capítulo 23.2 *Breadth-first search*.

³ Cfr. [Cormen90], capítulo 25 *Single-Source Shortest Paths*.

relação, a orientação do modelo virtual da seta é enviada para o módulo de síntese de imagem.

Na figura 5 pode ver-se dois exemplos de síntese de imagem de objectos virtuais na imagem real capturada pela câmara. Na imagem da esquerda o modelo indica que se devem descer as escadas e na imagem da direita que se deve seguir em frente.

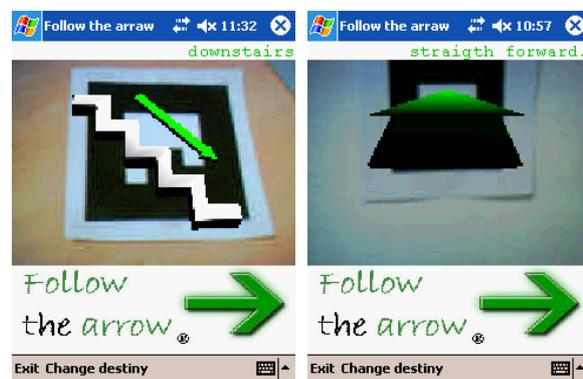


Figura 5. Dois exemplos de síntese de imagem de objectos virtuais na imagem real. Os modelos virtuais indicam o caminho que se deve seguir

3. ESQUEMA DE DISTRIBUIÇÃO DAS MARCAS PELO EDIFÍCIO

O número possível de marcas visuais a utilizar foi limitado pela versão do *ARToolkit* para PDA disponível à data da realização deste projecto. A nova versão entretanto publicada, o *ARToolkitPlus* [Wagner05], consegue analisar mais de 512 marcas, a uma taxa de 150 imagens por segundo. Como a nova versão ainda não estava disponível na altura da concretização do sistema, a solução usada foi reutilizar as marcas disponíveis, visto que para muitos edifícios haverá mais locais do que marcas disponíveis.

A vantagem de reutilizar marcas é que quanto menor for o número de marcas, menor será o número de comparações entre a assinatura DCT (*Discrete Cosine Transform*) da marca detectada pela câmara e as marcas registadas no sistema e, por outro lado (e mais importante), também será menor a semelhança e correlação entre as marcas. Estes factores diminuem o tempo e os erros na identificação das marcas e no registo de informação gráfica 3D nas mesmas.

A utilização de marcas únicas em cada local colmata por sua vez a grande limitação da reutilização de marcas: neste caso, o utilizador tem obrigatoriamente de filmar, senão todas, pelo menos algumas marcas que identificam zonas, consoante o esquema de reutilização usado, pois caso contrário o sistema não conseguirá discernir onde é que aquele se encontra. O sistema precisa de conhecer o contexto da utilização.

Na nossa opinião esta é uma limitação muito grande, visto que o utilizador com muita facilidade pode passar por algumas marcas sem as filmar, quer inconscientemente

quer conscientemente, quando já conhece parte do edifício ou por qualquer outro motivo presume um dado caminho.

Apesar desta limitação, nos casos em que as marcas disponíveis são insuficientes é necessário colocar marcas iguais em diferentes sítios, sendo nesse caso um dos requisitos da utilização do sistema que todas as marcas do caminho têm que ser filmadas.

Caso isso não aconteça o sistema perde a posição actual do utilizador dando-lhe a possibilidade de seleccionar o sítio onde se encontra, caso o saiba.

O nosso sistema está preparado para funcionar com os dois esquemas de distribuição de marcas: a reutilização de marcas e a utilização de marcas únicas, no caso de menos de 100 marcas serem necessárias.

A associação das marcas aos locais é feita automaticamente, através do já referido editor *Adjacent Points Editor – for Buildings Representation*, ficando registada no também referido ficheiro “assignedMarkers.txt”, dentro da pasta do respectivo edifício.

A primeira coisa que o algoritmo de distribuição de marcas faz é contar o número de nós introduzidos no editor. Se esse número for superior a 100, é seguido um esquema de reutilização de marcas e, na primeira linha do ficheiro “assignedMarkers.txt”, é introduzida a expressão ‘MARKERS_SCHEME reusable’. Se o número de nós for inferior a 100 será usado um esquema mais simples, para atribuição de marcas únicas a cada nó. A primeira linha do ficheiro “assignedMarkers.txt” será então ‘MARKERS_SCHEME single_markers’.

Esta expressão, que identifica o tipo de marcas usado, será usada pela função que procura o local a que uma marca pertence. No caso de marcas únicas a identificação é directa.

3.1 REUTILIZAÇÃO DE MARCAS

A estratégia de reutilização de marcas usada foi uma adaptação de um algoritmo que resolve o problema da coloração de grafos. O conhecido *graph coloring problem* [Jensen95, Dirac51], lança o desafio de se conseguir um conjunto mínimo de cores para colorir os nós de um grafo, sem que quaisquer dois nós adjacentes tenham a mesma cor. O mesmo problema se apresenta para colorir o mapa-mundo sem ter dois países adjacentes com a mesma cor. À solução apresentada por P. Gopala Krishna [Krishna98], realizámos as necessárias alterações para que a mesma marca seja usada não apenas com a intercalação de um nó com uma marca diferente, mas sendo também intercalada por dois nós com marcas distintas. Isto é necessário, porque se tivéssemos por exemplo dois nós ‘A’ separados apenas por um nó ‘B’, nunca saberíamos em que sentido se tinha ido de ‘B’ para ‘A’.

Inicialmente tem-se um conjunto de 5 marcas para reutilizar. Se este não for suficiente, o sistema vai acrescentando marcas até que se consiga o resultado pretendido.

O algoritmo assume que os nós do grafo estão ordenados por ordem decrescente de grau do nó (número de nós

adjacentes). Para que isto aconteça utilizámos o algoritmo de ordenação *Selection Sort*.

3.2 ESTRATÉGIA DE RECONHECIMENTO DO SÍTIO A QUE PERTENCE A MARCA

Como a mesma marca está presente em vários locais do edifício, coloca-se a questão de saber a qual deles pertence quando for reconhecida pelo sistema. Pelos motivos já explicitados, assume-se que neste caso são filmadas todas as marcas ao longo do caminho indicado pelo sistema.

Quando uma marca é detectada na imagem, o sistema vai buscar todos os locais onde ela está presente ao ficheiro “assignedMarkers.txt”. Depois, se se trata da primeira marca filmada, confirma-se que é uma marca única correspondente a uma das entradas do edifício. Caso não seja, é pedido ao utilizador que indique onde se encontra.

Se a marca reconhecida pelo sistema já não é a primeira, sabe-se que o caminho para o destino já foi descoberto, pois o caminho mais curto para o destino escolhido pelo utilizador, é calculado assim que o sistema sabe onde é que este se encontra, ou seja, quando a primeira marca é identificada. Deste modo, pode-se procurar entre os locais trazidos do ficheiro “assignedMarkers.txt” se há algum que esteja presente no caminho mais curto calculado e pelo qual ainda não se tenha passado. Caso haja, assume-se que é esse o local onde o utilizador se encontra.

4. APLICAÇÃO FOLLOW THE ARROW

Follow The Arrow é a aplicação que corresponde ao nosso sistema de localização em espaços interiores. Pode-se ver uma imagem do ecrã inicial da mesma aplicação na Figura 6.



Figura 6. Ecrã inicial da aplicação *Follow The Arrow*. O utilizador escolheu o edifício ISCTE e o destino Adetti(HP).

Inicialmente é pedido ao utilizador que seleccione o edifício onde se encontra e depois o destino para o qual se pretende deslocar. Quando o utilizador carrega no botão ‘GO’ aparece um novo ecrã já com a imagem captada pela câmara do PDA. A partir desse momento o sistema analisa todas as imagens (*frames*) provenientes da câma-

ra de vídeo, através de um módulo do *ARToolkit*, o qual tenta identificar marcas conhecidas.

Quando uma marca é detectada no ecrã, uma seta com a direcção que o utilizador deve seguir, a partir desse ponto físico do edifício, para chegar da forma mais rápida ao destino (figura 5). Na Figura 7 podemos observar um utente a utilizar o sistema.

4.1 HARDWARE UTILIZADO

O PDA que utilizámos para testar a aplicação *Follow The Arrow* apresentada abaixo é um HP iPAQ equipado com um processador Xscale a 400 MHz. A câmara de filmar é uma VEO Photo Traveler 130S.



Figura 7. Utilizando a aplicação *Follow the Arrow*.

5. CONCLUSÕES E TRABALHO FUTURO

A aplicação *Follow The Arrow* revelou-se funcional, apesar da limitação de todas as marcas do caminho terem de ser filmadas. Em breve utilizaremos a nova versão do *ARToolkit*, que permitirá a utilização de um universo extenso de marcas únicas (até 512 marcas). Além disso, nos casos em que mais marcas sejam necessárias, colocaremos duas marcas lado a lado, o que aumentará em muito o número de ‘marcas’ disponíveis.

Mesmo não sendo uma ideia completamente nova [Kalkusch02], a concretização de um sistema de localização para o interior de edifícios pareceu-nos de grande utilidade, tendo em vista a crescente utilização de PDAs e a já consumada integração dos PDAs com os telemóveis.

Um trabalho futuro, mas já em andamento, será a passagem da aplicação *Follow The Arrow* para telemóveis do tipo SmartPhone, as quais suportam o ambiente de desenvolvimento que existe no equipamento PDA utilizado no nosso Projecto. O sistema está ainda a ser generalizado por forma a poder comunicar com um servidor remoto de plantas de edifícios, com a geração automática de grafos topológicos de cada edifício. Prevêem-se ainda, no curto prazo, a realização de estudos de avaliação usabilidade do sistema no apoio a eventos científicos a serem organizados no ISCTE e, ainda, uma demonstração nas instalações da *Microsoft* no *TagusPark*.

6. REFERÊNCIAS

- [Billinghurst03] Billinghurst, M., Shared Space Homepage, Junho 2003
www.hitl.washington.edu/research/shared_space/download/, pesquisa efectuada em Janeiro de 2005.
- [Cormen90] Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; *Introduction to Algorithms*, 2nd edition, 1990, MIT Press and McGraw-Hill Book Company.
- [Dirac51] Dirac, G., "Note on the colouring of graphs", *Math. Z.* 54, 1951.
- [Jensen95] Jensen, R., TOFT, B., *Graph Coloring Problems*, Wiley-Interscience Series in Discrete Mathematics and Optimization, 1995.
- [Kalkusch02] Kalkusch, M., Lidy, T., Knapp, M., Reitmayr, G., Kaufman, H., and Schmalstieg, D. "Structured Visual Markers for Indoor Pathfinding". *Proceedings of the First IEEE International Workshop on ARToolkit (ART02)*, 2002.
- [Kato03] Kato, H.; Billinghurst, P.; Poupyrev, I., *ARToolkit version 2.3.3*, November 2002.
- [Klimt03] <http://studierstube.org/klimt/>, pesquisa efectuada em Janeiro de 2005.
- [Krishna98] <http://www.geocities.com/krishnapg/graphcoloring.html>, pesquisa efectuada em Março de 2005.
- [Wagner03] Wagner, D., Schmalstieg, D. "First Steps Towards Handheld Augmented Reality", To appear in: *Proceedings of the 7th International Conference on Wearable Computers*, White Plains, NY, USA, Oct.21-23, 2003.
- [Wagner04] Wagner, D., Schmalstieg, D. "ARToolkit on the PocketPC Platform".
- [Wagner05] http://studierstube.org/handheld_ar/artoolkitplus.php, pesquisa efectuada em Junho de 2005.