

# Realidade Aumentada com Dados Científicos em Dispositivos Móveis

José Nunes Pedrosa, Maria Beatriz Carmo, Ana Paula Cláudio,  
Ana Paula Afonso, António Ferreira, Paula Redweik, Cristina Catita  
Faculdade de Ciências, Universidade de Lisboa  
Campo Grande, 1749-016 Lisboa  
fc37417@alunos.fc.ul.pt, {bc,apc,apa,asfe}@di.fc.ul.pt,  
{predweik,cmcatita}@fc.ul.pt

---

## Sumário

*A Realidade Aumentada (RA) tem vindo a atingir grandes desenvolvimentos aplicativos na última década, em parte devido à expansão da indústria de dispositivos móveis e ao bom compromisso que apresentam entre a portabilidade, capacidade de processamento, e recursos de entrada e saída de dados. Neste artigo descrevemos a extensão de uma biblioteca para a plataforma Android que permite visualizar dados científicos em RA, com base na qual desenvolvemos uma aplicação para tablets que mostra níveis de radiação solar sobre edifícios. Os dados são projetados em tempo real com diferentes modos de visualização, como superfícies ou glifos, e com a possibilidade de serem exibidas transições animadas que facilitam comparações entre dados relacionados. São sucintamente expostos os desafios e opções tomadas no desenvolvimento da biblioteca.*

## Palavras-chave

*Realidade aumentada, visualização, dados científicos, computação móvel.*

---

## 1. INTRODUÇÃO

Uma aplicação de Realidade Aumentada (RA) recorre a imagens captadas do mundo real para a estas sobrepor representações virtuais que proporcionam ao utilizador informação adicional relativa ao seu contexto. No entanto, a quantidade de aplicações com uma utilidade prática tem sido em número reduzido até há pouco tempo, em parte devido às anteriores limitações tecnológicas e por outro lado devido aos elevados requisitos de *software* e *hardware* que estas requerem, especialmente no contexto móvel [Kim12].

Recentemente, com a massificação dos *tablets*, abriram-se novas possibilidades ao desenvolvimento de aplicações de RA. Estes dispositivos móveis possuem um ecrã de dimensões mais generosas, que permitem uma melhor visibilidade e leitura de informação sem penalizar demasiado a portabilidade. Adicionalmente, possuem os mesmos sensores que outros tipos de dispositivos móveis, tais como acelerómetro, giroscópio, bússola, localização por GPS, conectividades 3G, Wi-Fi ou Bluetooth, e uma ou mais câmaras fotográficas de alta resolução. O desempenho e poder de processamento destes dispositivos tem vindo a aumentar, sendo já bastante comum encontrar *chipssets* com dois ou quatro núcleos com elevadas frequências de relógio, que apesar de tudo são otimizados para um (relativamente) baixo consumo de bateria.

A capacidade destes dispositivos, combinada com os progressivos avanços e maturidade dos sistemas de realidade aumentada, permite o desenvolvimento de diversas apli-

cações de grande utilidade nas áreas da ciência, saúde, engenharia ou entretenimento.

O objetivo deste trabalho foi a extensão de uma biblioteca para a construção de aplicações de visualização de dados científicos em RA, dirigidas a dispositivos móveis com o sistema operativo Android, e de uma aplicação, recorrendo a essa biblioteca, para a consulta de informação de exposição solar sobre edifícios.

Os dados para a aplicação foram fornecidos pelo Departamento de Engenharia Geográfica, Geofísica e Energia (DEGGE) da Faculdade de Ciências da Universidade de Lisboa (FCUL) e consistem nos níveis de radiação solar anual dos edifícios do *campus* da FCUL, pelo que são valores escalares associados a localizações espaciais. A aplicação utiliza informação de geolocalização e orientação, bem como imagens captadas pela câmara, para projetar em tempo real esses mesmos dados sobre a imagem do(s) edifício(s). Recorre ainda a diversos modos de visualização de dados para consultar a informação. Os mecanismos para conseguir o alinhamento dos dados com o cenário recorrem aos sensores do aparelho.

Na secção 2 apresentamos o trabalho relacionado com o descrito neste artigo. Na secção 3 descrevemos mais detalhadamente a extensão da biblioteca, indicando as opções de desenho e as funcionalidades implementadas. Na secção 4 apresentamos a aplicação desenvolvida para a visualização de dados sobre radiação solar, a qual utiliza os recursos da biblioteca e descrevemos a interface com o utilizador e as opções de visualização disponíveis. Na

secção 5 apresentamos as principais conclusões e sugerimos melhoramentos futuros.

## 2. TRABALHO RELACIONADO

Uma das principais problemáticas da RA é como alinhar as representações virtuais com a imagem real sem recorrer a marcas fiduciais e independentemente do objeto de estudo. Em [Honkamaa07] procurou-se implementar um sistema capaz de ser executado em tempo real em dispositivos móveis de tipo mini-PC. Recorrendo à leitura de valores do acelerómetro, consegue-se gerar uma estimativa do movimento da câmara do utilizador à medida que este se desloca em torno da cena. Para o posicionamento e alinhamento do objeto tridimensional com a cena, foram consideradas duas abordagens: a semiautomática e a automática. A abordagem semiautomática estabelece que é o utilizador a indicar manualmente a posição do objeto (em particular a distância e a orientação), o que pode ser realizado, por exemplo, através de interação tátil. Posteriormente a aplicação mantém a visualização alinhada através de estimativas de deslocação da câmara. Na abordagem automática, utiliza-se o GPS do aparelho móvel em conjunção com a informação geográfica recolhida do Google Earth para realizar o cálculo da posição do objeto a visualizar. O carregamento do mesmo é feito diretamente a partir de um ficheiro KML, permitindo obter de uma só vez a informação geográfica e geométrica do modelo do edifício a apresentar. Estas duas abordagens de alinhamento foram adotadas no nosso trabalho.

São também de particular relevância estudos em espaços urbanos, uma vez que os dados do nosso trabalho são aplicados a edifícios. Em [Hakkarainen09], o trabalho acima referido é expandido no sentido de tornar a aplicação mais modular e reutilizável, abordando ainda a questão do carregamento e armazenamento de dados e da implementação de transições animadas. Neste, o objetivo foi desenvolver uma aplicação para auxiliar em trabalhos de construção civil e edificação no âmbito de um projeto designado *Augmented Reality for Building and Construction (AR4BC)*. Procurou-se uma implementação que permitisse a visualização do local de construção, bem como a sua evolução ao longo do tempo. Devido ao elevado nível de rigor exigido, recorreu-se a técnicas de localização mais precisas como bússola eletrónica e sensores de inércia. Foi desenhado um novo sistema de visualização, baseado numa arquitetura cliente-servidor, em que ambos os componentes podem ser executados na mesma máquina (se esta possuir capacidades de *hardware* para tal), ou separadamente em duas máquinas. Neste caso a informação de localização e orientação é enviada do cliente para o servidor, que por sua vez calcula a imagem resultante e a devolve ao cliente para ser apresentada. Os algoritmos usados para o alinhamento são importados de uma biblioteca de RA designada ALVAR (*A Library for Virtual and Augmented Reality*). Esta biblioteca foi estendida na tentativa de otimizar o seu funcionamento em dispositivos móveis e encontra-se documentada com maior detalhe em [Woodward11]. Um projeto semelhante é apresentado em [Gimeno10], que procura adicionar uma componente de RA aos tradicionais sistemas de CAD, sendo nesse

caso utilizada uma câmara acoplada a uma grua durante a fase de construção.

Um outro projeto, o *SiteLens* [White09], centra-se no desenvolvimento de uma plataforma e conjunto de técnicas que permitem a chamada *situated visualisation*, isto é, a visualização de dados relevantes contextualizados num espaço físico. O objetivo é auxiliar nas tarefas de planeamento urbano, em que tipicamente são necessárias visitas ao local para averiguar os níveis de tráfego automóvel, vegetação, ou densidade populacional. O *SiteLens* dá ao utilizador a capacidade de analisar dados no local, tais como a saúde dos moradores ou as condições do ar. O mecanismo suporta visualização em dispositivos fixos, móveis ou mesmo *head-mounted-displays*. Recorre-se à conversão de dados no formato KML, permitindo a interoperabilidade com o Google Earth e o Google Maps. A informação que não tiver uma associação espacial é afixada num canto do ecrã, enquanto a restante informação é sobreposta dinamicamente à vista principal. No *SiteLens* foram também desenvolvidas experiências com ícones naturais (exemplo: nuvens de fumo negras para representar a densidade de CO). Existe ainda a opção de “congelar” (ou “trancar”) a imagem, interrompendo a visualização de tempo real para uma melhor consulta da informação, e fazer um *zoom* semântico sobre cada ponto de dados. Estas duas soluções revelaram-se bastante úteis e intuitivas em testes com utilizadores. A opção de “trancar” a imagem foi também adotada no nosso trabalho.

Em [Heuveline11] é apresentado um projeto que procura resolver a questão de como selecionar os dados pretendidos de um grande volume de dados, bem como gerar uma visualização que realize uma filtragem inteligente da informação perceptível ao utilizador. Neste caso, é utilizado um modelo virtual tridimensional de várias ruas e edifícios a analisar, cuja informação geométrica é tida em conta ao fazer a modelação dos dados (nesse caso, correntes de ar). O processamento dos dados e geração de imagens são realizados em servidores remotos de alto desempenho, recorrendo em parte à biblioteca VTK e ao *software* ParaView. As imagens resultantes são enviadas para o cliente, prontas a ver no local, incluindo já a oclusão causada pelos edifícios. Para minimizar os custos de processamento, bem como a latência e consumo da rede na transmissão de dados, a visualização no dispositivo móvel aplica uma técnica detalhada em [Helfrich-Schkarbanenko12], que recorre à criação de imagens com diferentes perspetivas baseadas na posição e orientação da câmara. Assim, para cada posição do utilizador, é recebido um pacote de imagens pré-calculadas que são apresentadas conforme o deslocamento da câmara e o ângulo de visão. Dados os objetivos específicos do nosso trabalho, incluindo o facto de no estado atual de desenvolvimento apenas produzir a visualização dos dados sobre um edifício, optámos por uma abordagem mais simples, em que todo o processamento é feito no cliente.

Realizámos também uma pesquisa de plataformas de RA para Android cujas funcionalidades fossem ao encontro das necessidades do nosso trabalho, nomeadamente o alinhamento baseado em localização (sem marcas fiduci-

ais) com suporte para gráficos tridimensionais e o fácil acesso e modificação do código-fonte.

Algumas das plataformas consideradas foram o Wikitude, Metaio e Layar, que suportam georreferenciação e reconhecimento de imagem (ou marcas fiduciais), mas que são de código fechado e focados em pontos de interesse, em que o suporte de gráficos 3D resume-se à importação de modelos estáticos a representar sobre esses pontos [WDK, MC, LDK]. Dados os objetivos do trabalho, o recurso a estas plataformas não foi considerada viável. Outras alternativas de código aberto analisadas foram o Mixare, que se baseia exclusivamente na georreferenciação de pontos de interesse [Mixare], e o AndAR, que recorre ao OpenGL ES para carregamento de modelos 3D, mas que possui apenas alinhamento baseado em marcas fiduciais [AndAR]. Assim, optámos por recorrer à biblioteca, RUBI [SilvaP11b, Montez12], que temos vindo a desenvolver, realizando as extensões necessárias para a visualização de dados científicos cumprindo as funcionalidades acima referidas.

A RUBI é uma biblioteca de visualização de dados em RA para o sistema operativo Android [SilvaP11a]. Esta permite a recolha e pré-visualização de imagens por meio da câmara de vídeo e ainda a localização por meio de GPS, e a orientação através da bússola e acelerómetro. Realiza também o carregamento de dados remotos em formato JSON. Esta biblioteca foi estendida [Montez12], passando a permitir o ajuste da visibilidade de símbolos em função da imagem de fundo. Esta versão estendida da biblioteca RUBI serviu de base à Rubi Glare (*GL Augmented Reality*), que se apresenta na secção seguinte.

### 3. BIBLIOTECA RUBI GLARE

#### 3.1 Motivação e Desafios

Os requisitos essenciais que um sistema de RA procura atingir são *interatividade*, *fidelidade*, *escalabilidade* e *robustez* [Luo11]. A representação virtual deve correr em tempo real, e idealmente sem grandes restrições quanto à quantidade e qualidade dos dados apresentados, bem como ser tolerante a distúrbios na leitura de dados que possam ser causados por fatores externos. Também é necessário ter em conta, para dispositivos móveis, a diversidade de arquiteturas-alvo, capacidades do *hardware* limitadas em comparação com *desktops* e *laptops*, e requisitos de poupança de energia.

A biblioteca RUBI já possuía capacidades de combinação de imagens da câmara com símbolos virtuais, leitura de dados GPS e de acelerómetro [SilvaP11a, Montez12]. No entanto, ainda não possuía mecanismos para desenhar gráficos 3D e os apresentar sobre a imagem, nem módulos para visualização de dados científicos. Um dos desafios deste trabalho consistiu em desenhar e concretizar esses módulos, tendo já em mente o desenvolvimento de uma aplicação para visualizar dados de radiação solar sobre edifícios.

Outro desafio foi a otimização da biblioteca para permitir que a visualização de gráficos 3D executasse em tempo real. Atendendo ao tipo de aplicação a desenvolver, recorreremos a um *tablet*, pois possui um ecrã com maiores dimensões que um *smartphone*. Neste caso o dispositivo

foi um Asus TF300T com o sistema operativo Android 4.2 *Jelly Bean*, com um ecrã LCD de 10 polegadas, 16 milhões de cores e 1280x800 pixels de resolução, e um *chipset* Nvidia Tegra 3 *quad-core* de 1.3 GHz.

#### 3.2 Arquitetura Modular

A integração de capacidades de RA em dispositivos móveis adota uma estrutura modular. A informação obtida através dos sensores de orientação (bússola, acelerómetro e giroscópio) e de localização (GPS) do aparelho permite que a captação da imagem seja associada uma posição com 6 graus de liberdade (translação e rotação segundo um sistema de coordenadas com 3 eixos). A biblioteca RUBI Glare tem um módulo de *input/output* que lê dados da memória do dispositivo ou de um servidor remoto e que os passa para o motor de gráficos, que os representará no ecrã. Os gráficos tridimensionais são assim combinados com as imagens da câmara em tempo real.

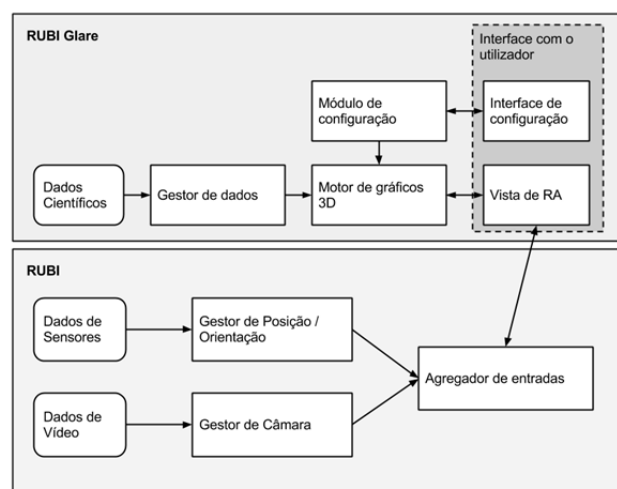


Figura 1: Módulos da biblioteca RUBI Glare.

Para o desenvolvimento do motor de gráficos considerámos inicialmente o VTK (*Visualization Toolkit*), por este estar otimizado para a leitura e processamento de dados científicos e por já termos experiência anterior de utilização. O VTK é baseado em OpenGL e implementado na linguagem C++, mas não é compatível com o Android, pois este suporta apenas um subconjunto das funcionalidades correspondentes à especificação OpenGL ES. Esta incompatibilidade do VTK, e de outras bibliotecas semelhantes do nosso conhecimento, com o Android levou à necessidade de desenvolvermos de raiz um módulo capaz de gerar visualizações tridimensionais de dados. O resultado desse trabalho foi uma extensão da biblioteca RUBI, designada RUBI Glare (*GL Augmented Reality*). A Figura 1 mostra uma vista de alto nível da estrutura de módulos da biblioteca, identificando também alguns fluxos de dados entre estes.

#### 3.3 Representações Gráficas

Começámos por desenvolver um algoritmo que permitisse gerar primitivas geométricas em OpenGL ES, tais como triângulos e quadrados. De seguida criámos código para gerar representações baseadas em glifos e superfícies coloridas.



Para a representação de glifos, jogámos com a capacidade de gerar primitivas geométricas para desenvolver um modo de visualização que utiliza ícones em forma de cubos, que podem ser individualmente posicionados e coloridos. Em cada ponto onde se conhecem valores da variável a observar é desenhado um cubo centrado nessa posição.

Numa fase seguinte, desenvolvemos um algoritmo para a construção de superfícies coloridas (que podem ou não ser planas), constituídas por uma malha de polígonos cujos vértices podem ser coloridos individualmente através do ajuste de parâmetros RGBA. O resultado é a criação de uma superfície colorida com transparência ajustável e que faz a interpolação automática de cores por meio de *vertex shading* graças à biblioteca OpenGL ES do Android. O algoritmo parte da estrutura de pontos de cada grelha regular e constrói um vetor de vértices com as coordenadas (X,Y,Z) de cada ponto, um segundo vetor de cores com a informação RGBA para cada ponto e um terceiro vetor com a ordem (índice) de desenho de cada ponto. Estes três vetores são carregados para a memória gráfica do dispositivo para gerar uma superfície colorida. Note-se que embora a topologia e a proporção (largura e altura) de cada superfície devam ser regulares, a quantidade de pontos e a geometria podem ser arbitrárias.

Adicionalmente, é possível aplicar uma deformação na superfície gerada. Isto é realizado calculando a orientação do vetor normal à superfície e aplicando uma translação em cada ponto segundo esse vetor, que é multiplicada pelo valor da variável que está a ser representada e um fator de escala ajustável.

### 3.4 Alinhamento

O alinhamento das representações gráficas geradas pelo *renderer* do OpenGL ES com a imagem captada pela câmara baseia-se na posição e orientação do observador. A biblioteca RUBI permite-nos aceder a estes valores através dos sensores do *tablet*. O GPS permite obter em tempo real a latitude e longitude do observador, enquanto a bússola digital (baseada no acelerómetro e magnetómetro) fornece a orientação/rotação do dispositivo. Finalmente, com a ajuda do acelerómetro, é possível também calcular a inclinação vertical do mesmo. Uma vez que as características do acelerómetro e da bússola variam ligeiramente entre dispositivos, houve necessidade de fazer ajustes no código da biblioteca de modo a melhorar a qualidade dos dados obtidos pelos sensores. Apesar de tudo, no caso da bússola digital esta revelou-se muito suscetível a falhas e a interferências de campos magnéticos gerados por outros dispositivos eletrónicos. Não foi possível averiguar com clareza se isto se deveu a um defeito em particular do Asus TF300T, ou se é generalizado. A imprecisão da bússola levou à necessidade de acrescentar mecanismos de alinhamento manual.

## 4. APLICAÇÃO PARA VISUALIZAÇÃO DE NÍVEIS DE RADIAÇÃO SOLAR

### 4.1 Pré-Processamento dos Dados

Os dados utilizados pela aplicação consistem em valores de radiação solar sobre o edifício C6 da FCUL, nomeadamente a radiação solar direta, difusa e global, medidas

em kWh/m<sup>2</sup>, bem como o número de horas de sombra anuais. Os valores de radiação estão dispostos numa grelha de pontos regular, georreferenciados através da latitude, longitude e altitude de cada ponto. As coordenadas de cada ponto encontravam-se originalmente no sistema cartográfico ETRS89-TM06/PT, as quais tiveram que ser convertidas para coordenadas geográficas no *datum* WGS84, de modo a serem compatíveis com o referencial usado pelo GPS do *tablet*. Cada ponto é identificado por um código numérico único que identifica a fachada do edifício a que pertence.

O pré-processamento dos dados inclui também a remoção de *outliers* através da análise da topologia de cada grelha de pontos para identificação e correção de irregularidades, por forma a gerar um ficheiro de dados com grelhas de topologia regular e formatado de acordo com o módulo de gestão de dados da nossa aplicação. Uma parte deste pré-processamento teve de ser manual.

O módulo de gestão de dados abre o ficheiro resultante do pré-processamento para leitura (neste momento em formato CSV) e carrega, linha a linha, a informação de cada ponto para memória. O ficheiro de dados possui os pontos agrupados por fachada do edifício e ordenados por coluna, de forma a otimizar o restante processo de geração dos gráficos.

### 4.2 Interface e Funcionalidades

A interface gráfica da aplicação foi desenhada de modo a ser simples e rápida de utilizar, permitindo um acesso intuitivo e direto às funcionalidades disponibilizadas ao utilizador. Seguimos os padrões de desenho estabelecidos para o sistema operativo Android [ADP], tendo em conta esta ser uma aplicação de RA para ser usada em ambientes exteriores, com *tablets* (Figura 2).

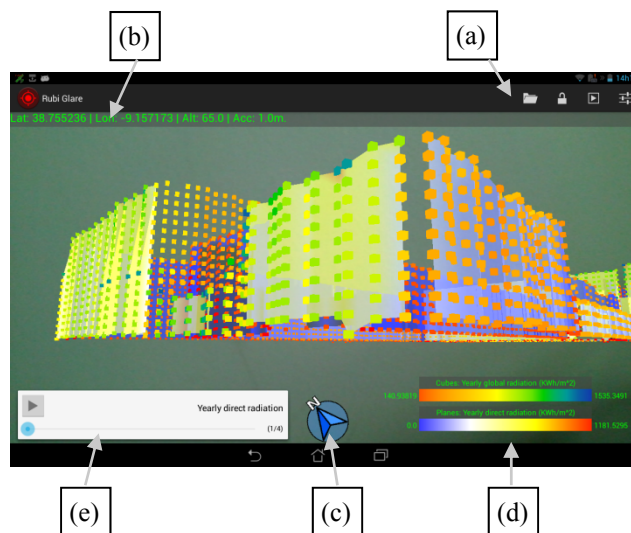


Figura 2: Interface com o utilizador, com a vista de RA em execução.

Respeitando as normas Android, na parte superior do ecrã encontra-se a barra de ações (Figura 2a) que dá acesso aos diferentes menus e funções da aplicação que estão disponíveis: abrir ficheiro de dados, trancar/destrancar vista, abrir controlos da transição animada de dados e

aceder às opções de configuração. À esquerda desta encontra-se a barra de estado do GPS (Figura 2b) com informação se o GPS está ou não em funcionamento, a latitude, longitude, altitude, e margem de erro obtidas pelo mesmo. O indicador da bússola (Figura 2c) encontra-se centrado na parte inferior da interface. A legenda (Figura 2d), ou legendas, são desenhadas à direita da bússola e o menu de controlo da transição animada de dados (quando esta é invocada) situa-se à esquerda da bússola (Figura 2e). O posicionamento destes itens foi pensado para minimizar a sobreposição dos mesmos com a visualização do edifício.

Existe ainda um navegador de ficheiros para seleccionar o ficheiro a abrir, e um menu de opções de rápido acesso desenhado para *tablets*, onde todos os parâmetros de configuração da visualização podem ser ajustados, bem como as configurações adicionais da aplicação.

#### 4.2.1 Modos de visualização

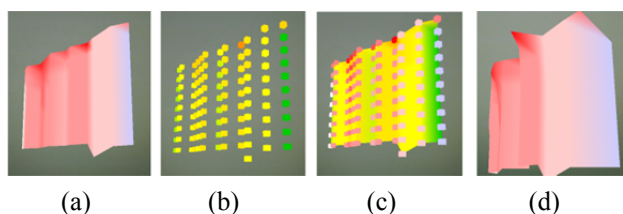


Figura 3: Exemplo dos quatro modos de visualização, aplicados sobre a mesma face do edifício.

A aplicação possui quatro modos de visualização, como ilustra a Figura 3. Os dois modos principais, superfícies coloridas e glifos coloridos, permitem a visualização de dados de uma só variável, enquanto os outros dois modos, superfícies com glifos e superfícies coloridas com deformação, permitem visualizar em simultâneo duas variáveis ou a mesma variável com duas representações distintas.

- **Superfícies coloridas** – Os dados são projetados segundo malhas poligonais criadas a partir de grelhas com topologia regular. Uma cor é atribuída a cada valor e existe interpolação de cor entre pontos, resultando numa superfície compacta com um efeito contínuo de gradação de cores. É possível escolher a variável e a escala de cores associada à visualização (Figura 3a).
- **Glifos coloridos** – Sobre cada posição do espaço de dados é desenhado um glifo, em forma de cubo, posicionado e colorido individualmente (Figura 3b). O volume destes cubos pode ser aumentado até se tornarem adjacentes, gerando uma superfície de mosaicos coloridos, ou diminuído de forma a distinguir cubos e posições individuais, com espaçamento entre si. Na sua dimensão mínima permite visualizar os dados como nuvens de pontos (Figura 4). Tal como no caso anterior, é possível escolher a variável e a escala de cores associada à visualização
- **Superfícies com glifos** – As visualizações de superfícies e glifos podem ser combinadas de forma a

mostrar duas variáveis em simultâneo (Figura 3c). A variável e escala de cores para cada tipo de visualização podem ser seleccionadas independentemente, sendo os glifos sobrepostos às superfícies coloridas.

- **Superfícies coloridas com deformação** – A visualização de superfícies coloridas pode ser complementada com a aplicação de uma deformação, de acordo com uma variável de dados especificada pelo utilizador (Figura 3d). A deformação consiste em deslocar os pontos da malha poligonal segundo um vetor normal à superfície e que aponta para o semi-espaco onde se encontra o utilizador. É possível também ajustar a intensidade da deformação, sendo que esta já sofre um ajuste automático conforme a variável seleccionada. Devido à dificuldade na percepção do relevo das superfícies, este modo requer ainda melhoramentos para ser eficaz no contexto da realidade aumentada.

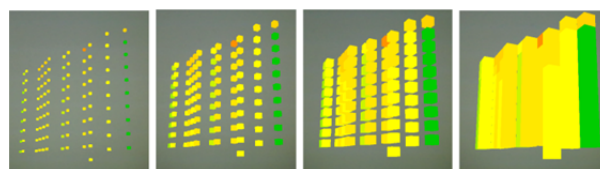


Figura 4: Comparação da visualização de glifos em diferentes dimensões.

#### 4.2.2 Opções adicionais

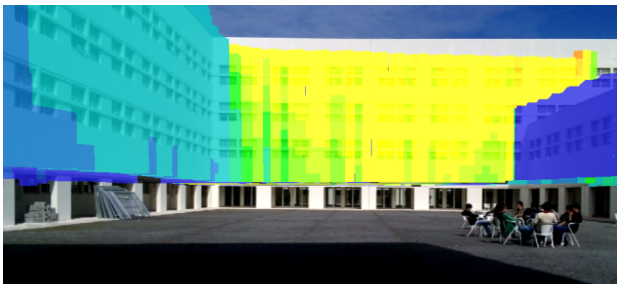
Para cada modo de visualização é possível seleccionar a escala de cor a associar à variável em estudo. Para cada escala de cor, existe também uma escala inversa, considerando que alguns dos valores relativos aos dados são referentes à quantidade de sombra em vez da radiação solar. As escalas de cores existentes são baseadas em [SilvaS11], e escolhidas pela sua utilidade na análise de radiação solar. Estas são apresentadas na Figura 5.



Figura 5: Escalas de cor utilizadas na aplicação.

Adicionalmente, é possível ajustar os níveis de opacidade dos gráficos. No contexto da RA, esta opção é muito importante, uma vez que deve existir um balanço entre a visibilidade dos dados e a visibilidade do edifício sobre o qual estes são projetados. Dependendo das condições do ambiente exterior, e das características da visualização, o nível de opacidade adequado pode variar, e assim, acrescentou-se a opção para o ajustar.

Outra funcionalidade oferecida é a opção de desligar ou ligar a oclusão que controla se um gráfico mais à frente oculta ou não outro mais atrás. O ajuste da opacidade e da oclusão são essenciais para a usabilidade da aplicação de RA, permitindo ao utilizador enquadrar mais facilmente a visualização de dados sobre o contexto do espaço físico em que este se encontra (Figura 6).



**Figura 6: Dados sobrepostos à imagem da câmara, com transparência a 50% e oclusão ativada.**

Relativamente ao alinhamento dos gráficos com a imagem da câmara, existem também opções disponíveis ao utilizador, cuja principal utilidade é colmatar as limitações do *hardware*. Assim, adicionámos um modo de ajuste manual dos gráficos, semelhante ao mecanismo utilizado em [Honkamaa07], que permite retificar a posição destes arrastando-os com o dedo sobre o ecrã tátil. Esta funcionalidade é essencial em situações em que a bússola digital não funciona corretamente devido a interferência eletromagnética ou pelo facto de não estar calibrada. A sensibilidade e o sentido do arrastamento podem também ser ajustados.

Uma vez que a medição de altitude através do GPS é demasiado imprecisa, a aplicação utiliza uma altitude fixa que pode ser ajustada pelo utilizador.

Também desenvolvemos um modo de “trancar” a vista de RA, inspirado na solução apresentada em [White09]. Através de um botão é possível “congelar” totalmente a imagem apresentada pelo aparelho (quer a vista da câmara, quer os gráficos 3D) para mais confortavelmente consultar a informação sem a necessidade de apontar o aparelho constantemente na direção pretendida.

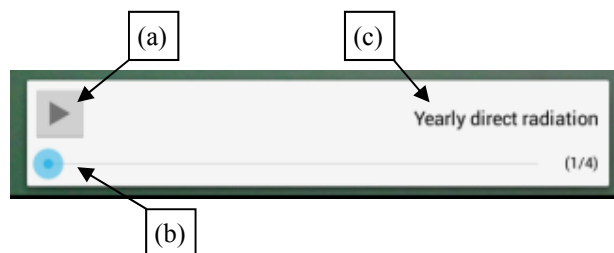
#### 4.2.3 Transição animada de dados

Um dos objetivos a que nos propusemos foi a possibilidade de gerar transições animadas que facilitam comparações entre dados relacionados. Para tal implementámos um modo de animação que pode ser iniciado e parado pelo utilizador, e que altera em tempo real os parâmetros de visualização a serem apresentados (funcionando mesmo que a vista esteja “trancada”). O utilizador pode ainda selecionar a sequência de variáveis a apresentar, e a velocidade da transição.

Esta funcionalidade irá ser estendida para animações da mesma variável ao longo do tempo, mas uma vez que os dados fornecidos não possuem informação temporal associada, fizemos a demonstração de conceito com a transição entre diferentes variáveis. As variáveis a incluir na animação, bem como o intervalo de transição, podem ser ajustadas pelo utilizador.

Ao invocar o menu de controlo da animação (através da barra de ações) este passa a ocupar o canto inferior esquerdo do ecrã. O botão do lado esquerdo permite iniciar e parar a animação (Figura 7a). A barra de progresso situa-se na parte inferior e funciona de forma semelhante à de um leitor de vídeo, podendo ser arrastada com o dedo para saltar para qualquer momento da animação (Figura 7b). Do lado direito encontra-se a legenda que indica a

instância da animação em curso (por exemplo, a variável ou instante de tempo atuais). A disposição destes itens impede que qualquer informação fique obstruída pelas mãos ou dedos do utilizador enquanto este interage com a aplicação.



**Figura 7: Menu de controlo da transição animada.**

## 5. CONCLUSÕES E TRABALHO FUTURO

Apresenta-se neste artigo o trabalho decorrente da extensão de uma biblioteca para visualização em RA, aplicando técnicas de visualização de dados científicos. Desenvolvemos um motor gráfico capaz de gerar diferentes modos de visualização, representando uma ou duas variáveis em simultâneo e com execução em tempo real. A biblioteca foi construída de forma modular podendo ser estendida com novas funcionalidades. A capacidade de realizar transições animadas sobre os dados está também implementada, suportando, no futuro, a análise da evolução temporal dos mesmos. Como caso de estudo, foi desenvolvida uma aplicação para visualização de dados sobre a radiação solar relativa a um dos edifícios do *campus* da FCUL.

Uma das limitações com que nos deparamos foi a falta de precisão dos mecanismos de geolocalização e orientação para conseguir o alinhamento rigoroso dos gráficos com a imagem real de fundo, forçando o uso de opções de ajuste manual. Estão em curso testes com utilizadores para recolher opiniões acerca da usabilidade das funcionalidades já implementadas, e posteriormente fazer melhoramentos à aplicação.

Como trabalho futuro pretendemos melhorar e estender a aplicação, otimizando o mecanismo de carregamento de dados e modificando o código atual para obter melhor desempenho gráfico. Procuramos também aperfeiçoar a aplicação no sentido desta apoiar a instalação e posicionamento de equipamentos de captação de energia solar. Algumas funcionalidades novas, como o registo de *snapshots* e animações para posterior análise, deverão de igual modo ser incluídas no futuro. Pretende-se ainda a adição de mais formas de filtragem da informação que permitam uma análise mais minuciosa, e a possibilidade de carregamento ou processamento de dados remotos.

## 6. AGRADECIMENTOS

Agradecemos à FCT (Fundação para a Ciência e a Tecnologia) e ao LabMAG (Laboratório de Modelação de Agentes) o apoio financeiro concedido ao bolsheiro José Nunes Pedrosa, ao abrigo do projeto estratégico PEst-OE/EEI/UI0434/2011.



## 7. REFERÊNCIAS

- [ADP] Android Design Patterns (acedido em 6-9-2013).  
<<http://developer.android.com/design/patterns/index.html>>
- [AndAR] AndAR – Android Augmented Reality (acedido em 12-9-2013).  
<<https://code.google.com/p/andar/>>
- [Gimeno10] J. Gimeno, P. Morillo, S. Casas, M. Fernández. *An Augmented Reality (AR) CAD System at Construction Sites*. In *Augmented Reality: Some Emerging Application Areas*, pp. 15-32. InTech (2010).  
<<http://www.intechopen.com/books/augmented-reality-some-emerging-application-areas>>
- [Hakkarainen09] M. Hakkarainen, C. Woodward, K. Rainio. *Software Architecture for Mobile Mixed Reality and 4D BIM Interaction*. In *Proc. 25th CIB W78 Conference*, Istanbul, Turkey, pp. 517-524. (2009).
- [Helfrich-Schkarbanenko12] A. Helfrich-Schkarbanenko, V. Heuveline, R. Reiner, S. Rittersbusch. *Bandwidth-Efficient Parallel Visualization for Mobile Devices*. In *Proc 2nd International Conference on Advanced Communications and Computation (INFO-COMP)*, Venice, Italy, pp. 106-112. (2012).  
<<http://emcl.uni-hd.de/fileadmin/images/Publications/Preprints/emcl-preprint-2012-07.pdf>>
- [Heuveline11] V. Heuveline, S. Ritterbusch, S. Ronnås. *Augmented Reality for Urban Simulation Visualization*. In *EMCL Preprint Series*, 7-2012. Engineering Mathematics and Computing Lab (EMCL), Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany (2011).
- [Honkamaa07] P. Honkamaa, S. Siltanen, J. Jäppinen, C. Woodward, O. Korkalo. *Interactive Outdoor Mobile Augmentation Using Markerless tracking and GPS*. In *Proc. Virtual Reality International Conference (VRIC)*, Laval, France, pp. 285-288. (2007).
- [Kim12] I. Kim. *An Introduction to Augmented Reality*. (Seminar Report) In *COMPSCI 705 SI C*. Department of Electrical and Computer Engineering, University of Auckland, Auckland, New Zealand (2012).
- [LDK] Laya SDK (acedido em 12-9-2013).  
<<https://www.layar.com/products/custom-solutions/sdk/>>
- [Luo11] X. Luo. *The Cloud-Mobile Convergence Paradigm for Augmented Reality*. In *Augmented Reality: Some Emerging Application Areas*, pp. 33-58. InTech. (2011).  
<<http://www.intechopen.com/books/augmented-reality-some-emerging-application-areas>>
- [MC] Metaio Creator (acedido em 12-9-2013).  
<<http://www.metaio.com/products/creator/overview/>>
- [Mixare] Mixare (acedido em 12-9-2013)  
<<https://code.google.com/p/mixare/>>
- [Montez12] E. Montez. *Visualização de Pontos de Interesse em Dispositivos Móveis Utilizando Realidade Aumentada*. Faculdade de Ciências da Universidade de Lisboa, Lisboa, Portugal. (Relatório final de Projeto em Engenharia Informática, 2012).
- [SilvaS11] S. Silva, B. Santos, J. Madeira. *Using Color in Visualization: A Survey*. In *Computers & Graphics*, 35(2), pp. 320-333. (2011).  
<<http://www.sciencedirect.com/science/article/pii/S0097849310001846>>
- [SilvaP11a] P. Silva. *Transição Entre Ambientes Externos e Internos e Visualização Adaptativa*. Faculdade de Ciências da Universidade de Lisboa, Lisboa, Portugal. (Relatório final de Projeto em Engenharia Informática, 2011).
- [SilvaP11b] P. Silva, P. Pombinho, A. Afonso, T. Gonçalves, Rubi: *An Open Source Android Platform for Mobile Augmented Reality Applications*. In *Workshop on Mobile Augmented Reality: Design Issues and Opportunities (MobileHCI)*, Stockholm, Sweden. (2011)
- [White09] S. White, S. Feiner. *SiteLens: Situated Visualization Techniques for Urban Sites Visits*. In *Proc. ACM HCI 2009*, Boston, MA, pp. 1117-1120. (2009)
- [WDK] Wikitude SDK for Android – Documentation (acedido em 12-9-2013).  
<<http://developer.wikitude.com/documentation/android>>
- [Woodward11] C. Woodward, M. Hakkarainen. *Mobile Mixed Reality System for Architectural and Construction Site Visualization*. In *Augmented Reality: Some Emerging Application Areas*, pp. 115-130. InTech. (2011).  
<<http://www.intechopen.com/books/augmented-reality-some-emerging-application-areas>>