Aplicação Colaborativa em Cenário Real para Animação de um Cenário Virtual

Telmo Gomes
ISEL - Instituto Superior de Engenharia de Lisboa
Rua Conselheiro Emídio Navarro, 1 - 1959-007 Lisboa, Portugal
(+351) 218 317 000
telmogomes190987@hotmail.com

Carlos Júnior
ISEL - Instituto Superior de Engenharia de Lisboa
Rua Conselheiro Emídio Navarro, 1 - 1959-007 Lisboa, Portugal
(+351) 218 317 000
c_junnior@hotmail.com

Rui Manuel Feliciano de Jesus
ISEL - Instituto Superior de Engenharia de Lisboa
Rua Conselheiro Emídio Navarro, 1 - 1959-007 Lisboa, Portugal
(+351) 218 317 000
rjesus@deetc.isel.ipl.pt

Pedro Mendes Jorge
ISEL - Instituto Superior de Engenharia de Lisboa
Rua Conselheiro Emídio Navarro, 1 - 1959-007 Lisboa, Portugal
(+351) 218 317 000
pjorge@deetc.isel.pt

Resumo

As aplicações mais populares de mistura de cenários reais com cenários virtuais têm sido as aplicações de realidade aumentada e de realidade mista (realidade aumentada + virtualidade aumentada). Este trabalho mistura a realidade com ambientes virtuais numa aplicação colaborativa para duas pessoas. O projeto tem como objetivo o desenvolvimento de uma aplicação para navegação em ambientes virtuais através da interpretação de movimentos reais de uma pessoa. Estes movimentos são capturados por uma câmara, interpretados com algoritmos de processamento de vídeo e transmitidos para a interface de animação e lógica de jogo do *Blender* [1] *(GameLogic* [2]*)*. Este artigo descreve a aplicação e apresenta resultados experimentais que ilustram o seu desempenho.

Palavras-chave

Animação Virtual, Blender, Gamelogic, Processamento de Vídeo e Realidade Mista.

Introdução

Atualmente observa-se uma evolução tecnológica no que diz respeito à simbiose do mundo virtual com o real. Esta ligação traduz-se pelo aparecimento de imensas aplicações de Realidade Aumentada (RA), Virtualidade Aumentada (VA) e Realidade Mista (RM) [3].

Este projeto insere-se na área de VA, pois utiliza informação útil retirada do mundo real para reprodução de movimento no mundo virtual.

O objetivo principal consiste no desenvolvimento de uma aplicação colaborativa entre dois intervenientes que é baseada na interpretação do tipo de movimento captado por uma câmara e reproduzida num ambiente virtual construído em Blender.

A colaboração entre os dois intervenientes é um ponto fulcral, visto que, a cada elemento está associado uma função específica. Um dos elementos funcionará como guia, e o outro elemento será guiado conforme as informações do guia. A comunicação entre ambos é realizada através de um dispositivo móvel, independente da implementação desenvolvida.

O elemento guiado encontra-se num espaço amplo equipado com uma câmara que filma o ambiente real, e o elemento que guia visualiza a interpretação do movimento real sob forma de movimentos em *first-person-shooter* (simulação do ponto de vista do protagonista, como se o jogador e personagem do jogo fossem o mesmo observador) num ambiente 3D.

As imagens captadas são processadas e traduzidas, sendo extraída informação sobre o tipo de movimento que é efetuado pelo elemento guiado. Esta interpretação é representada como movimentos de uma câmara virtual que permite ao guia a visualização da animação no cenário virtual.

Pretende-se que tanto o elemento guia como o guiado possam desfrutar de uma experiência de entretenimento onde a comunicação entre ambos seja imperativa para a boa usabilidade da aplicação.

Métodos

A aplicação desenvolvida é composta por dois blocos principais: i) Reconhecimento de Movimento (Fig. 1) e ii) Representação de Movimento (Fig. 2). Cada um destes blocos é composto por várias fases que são descritas a seguir.

O Reconhecimento de Movimento começa com a aquisição de imagens a partir de uma câmara que pode estar instalada num dispositivo móvel. Após a aquisição, existe uma fase de conversão para melhor manipulação das imagens. Em seguida, extraem-se os melhores pontos de referência [4] para a fase de seguimento (processo conhecido por *tracking* [5]). Com base no cálculo do fluxo ótico [6] é possível obter informações sobre a posição desses pontos em imagens consecutivas. Os pontos que existiam na imagem anterior e não existem na imagem atual são removidos e os restantes são usados para os cálculos de preparação (deteção do movimento).

Com base na estimação da matriz de transformação afim [7] é possível reconhecer tipos de movimentos básicos (esquerda, direita, cima, baixo, *zoom in* e *zoom out*) a partir dos pontos de interesse observados. Seguidamente são aplicados vários filtros de modo a remover deteções de movimento indesejadas e a organizar hierarquicamente cada movimento de acordo com o seu peso.

Estes movimentos são enviados por mensagem para o servidor [8], para que este possa começar o processo de Representação do Movimento. Enquanto isso, o cliente processa a próxima imagem e fica à espera que o servidor esteja desocupado.

Blender PT Conference 2013, Porto, Portugal
ISBN: A preencher pela comissão organizadora.

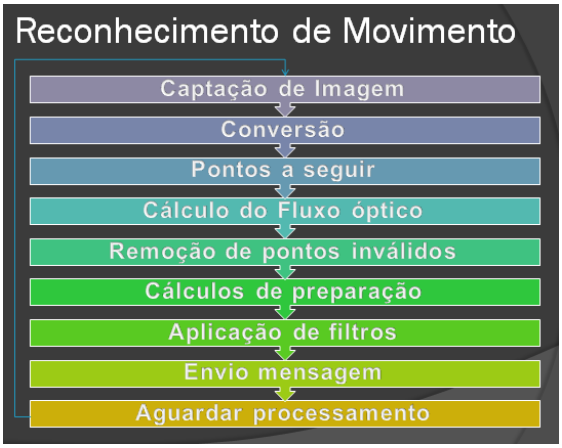

Fig. 1. Processo de Reconhecimento de Movimento

O processo de Representação de Movimento é efetuado no servidor após a receção da mensagem. Durante este processo, o servidor encontra-se no estado ocupado.

Ao receber a mensagem, o servidor interpreta-a e ativa o método correspondente. Por sua vez, esse método permite a ativação de um determinado atuador existente na interface de jogo do Blender (*gamelogic*). O atuador age sobre a câmara existente no cenário virtual de modo a que esta efetue o movimento correspondente à mensagem recebida.

Após a representação do movimento no cenário virtual, o servidor fica livre para que o cliente envie a próxima mensagem.

Fig. 2. Processo de Reconhecimento de Movimento

Resultados

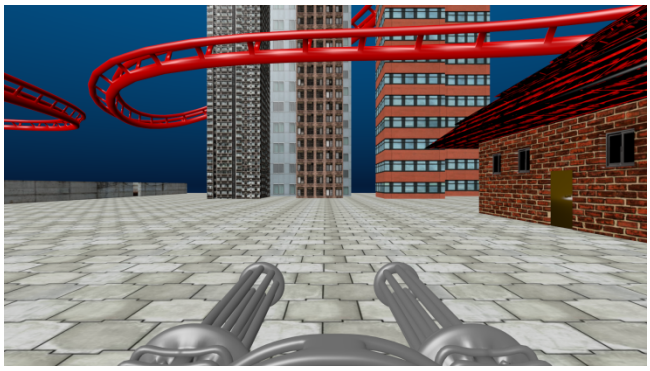

Fig. 3. – Ambiente virtual desenvolvido em Blender (vista da câmara)

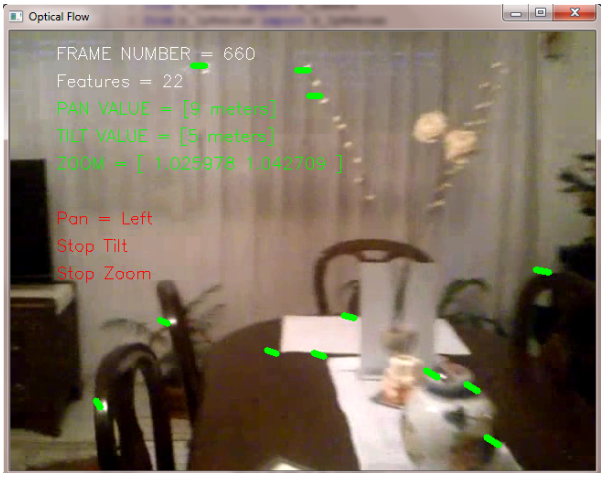

Fig. 4. Fluxo ótico em ambiente fechado (dentro de casa)

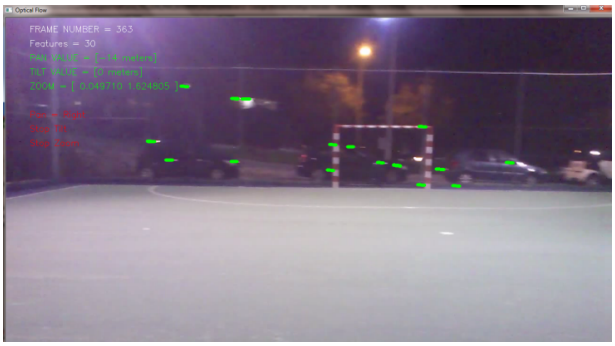

Fig. 5. Fluxo ótico em ambiente aberto (campo futsal)

Nos testes efetuados utilizaram-se dois vídeos: Um em ambiente fechado dentro de uma casa (ver Fig. 4) e outro em ambiente aberto num campo de futsal (ver Fig. 5).

Na figura 3 mostra-se o ambiente virtual desenvolvido em Blender para reproduzir os movimentos capturados pela câmara.

Uma das componentes importantes para o bom funcionamento da aplicação é a estimação do movimento realizado pela câmara. Esta estimação envolve o cálculo dos pontos de referência a seguir e o cálculo da matriz de transformação afim. Nesta secção apresentam-se os resultados da avaliação realizada ao reconhecimento do movimento.

As figuras 4 e 5 mostram uma imagem exemplificativa com os pontos de interesse calculados em cada um dos ambientes.

Para cada ambiente contabilizou-se o número de imagens em que o movimento real ocorreu e a respetiva classificação (tipo de movimento) ao longo das mesmas.

As tabelas 1 e 2 mostram os resultados obtidos para cada sequência de vídeo.

Ambiente Fechado	Tipo de movimento real			
	Left	*Right*	*Zoom In*	*Zoom Out*
Nº Frames	54	115	155	105
Recálculos	4	13	1	0
Mal Classificados	2	0	19	46
Bem Classificados	52	115	136	59
Precisão	96,30%	100%	87,74%	56,19%

Tabela 1 – Percentagem de pontos classificados para ambiente fechado

Blender PT Conference 2013, Porto, Portugal
ISBN: A preencher pela comissão organizadora.

Ambiente Aberto	Tipo de movimento real			
	Left	*Right*	*Zoom In*	*Zoom Out*
Nº Frames	168	144	120	101
Recálculos	2	3	4	5
Mal Classificados	0	4	47	74
Bem Classificados	168	140	73	27
Precisão	100%	97,22%	60,83%	26,73%

Tabela 2 - Percentagem de pontos classificados para ambiente aberto

Nas tabelas, as linhas indicadas como "Recálculos" correspondem a um novo cálculo dos pontos de interesse. Esta situação acontece sempre que o número de pontos onde foi possível determinar o seu seguimento em imagens consecutivas é menor que um limiar considerado relevante para o reconhecimento do movimento. Para estes testes o limiar definido foi de 10 e o valor inicial de deteção foi de 30.Assim, inicialmente a aplicação extraí 30 pontos de interesse numa imagem contudo, ao longo das sequências de imagens o número de pontos seguidos vai diminuindo, e quando existirem somente 10 ou menos é realizado uma nova estimação de 30 pontos (recálculo).

A precisão é calculada dividindo o número de casos favoráveis pelo número de casos possíveis. Os casos possíveis são a totalidade de imagens em que o movimento real ocorreu, os casos favoráveis são o número de imagens onde a classificação foi correta, ou seja, a classificação corresponde ao movimento real.

Conclusões

Este artigo propõe uma aplicação colaborativa que combina ambientes reais com cenários virtuais utilizando o Blender para construir o cenário virtual e uma câmara de vídeo para capturar a realidade. Analisando o comportamento da aplicação nos dois ambientes reais, verifica-se que a precisão de deteção de movimentos para a esquerda e para a direita é quase igual, situando-se muito próximo dos 100%. Assim sendo, conclui-se que a deteção deste tipo de movimentos é eficaz no que diz respeito ao desempenho da aplicação.

Relativamente aos movimentos do tipo Zoom In e Zoom Out, em ambiente fechado temos um valor médio de aproximadamente 72%, e em ambiente aberto o valor situa-se nos 44%. Esta diferença nos resultados está relacionada com o facto de em ambiente aberto os pontos de interesse para a deteção de movimento situarem-se em zonas muito mais distantes, de tal modo que estes movimentos da câmara são mais dificilmente detetados. Em ambiente fechado os objetos estão próximos da câmara e qualquer movimento é mais facilmente detetado.

O facto de existir um decréscimo de precisão dos movimentos Zoom In e Zoom Out relativamente aos movimentos para a esquerda e para a direita de deve-se a dois fatores. Primeiro, porque o movimento da pessoa que segura a câmara não é linear mas sim ondulatório, e segundo porque a aplicação tem uma grande capacidade de deteção de movimentos do tipo esquerda e direita. A união destes dois fatores leva a que quando a pessoa se mova para a frente ou para trás, alguns deslocamentos ondulatórios possam ser interpretados de outra maneira.

Verifica-se ainda um decréscimo de precisão de cerca de 32% do Zoom In para o Zoom Out. Isto acontece porque as pessoas andam naturalmente para a frente e muita raramente para trás. Assim sendo, o movimento ondulatório para trás é mais acentuado e acidentado do que o movimento ondulatório para a frente, originando erros de deteção.

Assim conclui-se que, apesar de algumas heurísticas existentes na aplicação, os movimentos de Zoom têm alguns erros. No entanto, estes erros podiam ser minimizados se o teste fosse feito utilizando, por exemplo, um robot para garantir mais estabilidade na aquisição do vídeo do ambiente real. É de referir que a câmara utilizada foi a de um telemóvel e não uma câmara de filmar, normalmente equipada com sistemas de estabilização de imagem. Em alternativa, poder-se-ia utilizar uma câmara *stereo* que permite a estimação da profundidade do campo de visão contribuindo para reduzir a percentagem de erros.

Referências

[1] Blender 2.62.1 (2012). Blender Documentation contents. Amesterdam.

[2] Blender 2.62.1 (2012). The Blender Game Engine Python API Reference. Amesterdam.
URL:
http://www.blender.org/documentation/249PythonDoc/GE/index.html Última consulta: 27 de Dezembro de 2012.

[3] Milgram, P. e Kishino, F. A taxonomy of mixed reality visual displays. IEICE Transactions on Information Systems, Vol E77-D, No.12 December, 1994,
URL:
http://etclab.mie.utoronto.ca/people/paul_dir/IEICE94/ieice.html
Última consulta: 27 de Dezembro de 2012

[4] Shi, J. e Tomasi, C. Good Features to Track. IEEE Conference on Computer Vision and Pattern Recognition (CVPR94) Seattle, June, 1994.

[5] Tomasi, C. e Kanade, T. Detection and Tracking of Point Features. United States Technical Report CMU-CS-91-132, Carnegie Mellon University, April, 1991.

[6] Stavens, David. The OpenCV Library: Computing Optical Flow. Stanford Artificial Intelligence Lab. URL: http://sourceforge.net/projects/opencvlibrary/. Última consulta: 27 de Dezembro de 2012.

[7] Shapiro & Stockman. CSE/EE 576: Computer Vision, February 2, 2001.

[8] Brutto AVT (2008). Sistema cliente-servidor em Python.
URL: http://brutto.proceduralbase.org/2008/10/26/sistema-cliente-servidor-em-python/

Blender PT Conference 2013, Porto, Portugal
ISBN: A preencher pela comissão organizadora.