

VIRTUAL 3D: SISTEMA MULTI-AGENTE DE VISUALIZAÇÃO COM CONTROLO INTELIGENTE DE CÂMARA

Sérgio Louro^{1,2}, Luís Paulo Reis^{1,3}, Eugénio Oliveira^{1,3}

sergio_louro@brainwork.pt, lpreis@fe.up.pt, eco@fe.up.pt

¹LIACC – Laboratório de Inteligência Artificial e Ciência de Computadores, Universidade do Porto, Portugal

²TecniDuplo – Departamento Tecnológico, Santarém, Portugal

³FEUP – Faculdade de Engenharia da Universidade do Porto, Portugal

Resumo: O Virtual3D é um sistema que permite a visualização de jogos em tempo real ou off-line do futebol robótico simulado 2D, num ambiente 3D com animação dos objectos e som. É um sistema multi-agente, em que cada câmara é modelizada como um agente inteligente autónomo, com agentes para o realizador, jogadores e público. Dado as percepções (posições e velocidades dos jogadores e da bola), os agentes “cameraman” estão aptos a decidir qual a melhor perspectiva em cada momento do jogo, enviando a informação para o agente realizador que é o responsável por escolher a melhor imagem a mostrar aos espectadores.

Palavras-chave: Agents, Computer Graphics, Virtual Camera, Cinematography, Director, Cameraman, Robotic Soccer.

1. INTRODUÇÃO

O Futebol Robótico simulado é um jogo de futebol, específico e muito atractivo, sendo jogado num ambiente distribuído e em tempo real, suportado por uma arquitectura multi-agente (Kitano et al., 95, 97), incluindo assim um conjunto de problemas científicos, que passam pelo processamento em tempo real, robustez de comunicação com ruído, cooperação de agentes numa arquitectura multi-agente e processamento incompleto de informação. A simulação é assegurada por uma arquitectura cliente/servidor em que o servidor central é designado de soccerserver, no qual se modela um campo de futebol virtual composto por duas equipas de agentes (Noda e Frank, 00). Cada equipa é constituída por onze jogadores tendo como objectivo marcar o maior número de golos para ganhar o jogo à equipa adversária.

Tradicionalmente o soccerserver é usado como ferramenta para estudar sistemas multi-agente na coordenação em tempo real, comunicação, planeamento e aprendizagem. O visualizador Virtual3D é também baseado no conceito de sistema multi-agente utilizado no soccerserver (Kitano et al., 95) da liga de futebol simulado do RoboCup (Kitano et al., 95) (Noda et al., 97). Porém, a contribuição principal está baseada no desenvolvimento do visualizador 3D associado ao sistema multi-agente de câmara, permitindo a visualização dos jogos em

tempo real ou em diferido (off-line). O estado do jogo no servidor inclui as posições, velocidades e direcções de todos os objectos num sistema de coordenadas bidimensional (duas dimensões), que serão convertidas para um sistema tridimensional para que o jogo se torne o mais realístico possível. Com os avanços recentes no desenvolvimento da tecnologia gráfica (Foley et al., 00), é agora possível desenvolver aplicações para mostrar uma visualização tridimensional realística destas partidas de futebol simuladas. É neste contexto que surge o visualizador tridimensional Virtual3D.

Alguns laboratórios de investigação já tinham mostrado anteriormente esta possibilidade. Devido às suas características pioneiras e ao seu método de pesquisa, podemos apontar o Virtual RoboCup da Universidade de Bielefeld (Jung et al., 99), Alemanha, o visualizador The Venue 3D da Universidade de Vrije de Amesterdão (Spoelder et al., 00, 99), Holanda, e o visualizador MagicBox (Xing et al., 02) (Li et al., 03) da Universidade da Ciência e Tecnologia da China. No entanto, o visualizador desenvolvido diverge destes sistemas devido ao acentuado realismo das animações e dos sons e ainda ao sistema multi-agente para controlo de câmara inteligente.

O sistema inclui como principais potencialidades:

- Ambiente 3D incluindo um estádio e animações realistas;
- Controle de câmara inteligente através de um Sistema Multi-Agente;
- Repetições em câmara lenta dos melhores momentos do jogo;
- Geração de sons 3D baseado em agentes emocionais que representam os espectadores de cada equipa, através de um motor de som 3D posicional;
- Capacidade de conexão em tempo real com o *socccserver* e de leitura de *logfiles* (versão 2 e 3), incluído as versões gerada para o sistema operativo NT;
- *Display* de estatísticas de jogo fornecidas por um agente analisador de jogo;
- Modelos 3D (Studio Max) para os jogadores com animação real;
- Suporte DirectX7+ para o sistema operativo Windows 2000 e XP;
- Paradigma de programação, ligação orientada por objectos, C++.

2. COMPUTAÇÃO GRÁFICA E CINEMATOGRAFIA

Tipicamente, a computação gráfica a três dimensões tenta simular o processo de visualização do mundo real, sendo os objectos modelados geometricamente num sistema de coordenadas devidamente normalizado. O efeito de luz nas superfícies dos objectos é simulado através de transformação das superfícies a três dimensões em planos a duas dimensões. O conceito geral da computação gráfica é a ideia de uma *pipeline gráfico*¹, em que a entrada de dados para visualização é uma cena gráfica a 3-Dimensões constituída por objectos (conjunto de polígonos), câmaras e luzes, em que a saída dos dados é uma imagem da cena a 2-Dimensões para ser apresentada no ecrã.

A câmara virtual é um conceito importante na computação gráfica, pois representa o ponto de vista sobre o qual os objectos pertencentes ao mundo virtual são projectados no ecrã. Por definição, a parte crucial de qualquer ambiente virtual é a forma como é vivido, dependendo em grande parte da visão da câmara. Em geral são considerados sete graus de liberdade pelos quais uma câmara virtual é controlada: três graus para mover no plano gráfico cartesiano (x, y, z), três graus para rotação (*pan, tilt e roll*) e um grau para mudar o campo de visão (*zoom*) (Nieuwenhuisen e Overmars, 02). Em Computação Gráfica recorre-se normalmente ao conceito de câmara virtual (Watt, 93) para definir um novo referencial, denominado Sistema de Coordenadas da Câmara (*Eye or Camera Coordinate System*), cuja origem coincide com o centro de projecção (posição da câmara) e cujo eixo dos ZZ é normal ao plano de

visualização (*View Plane*) e representa a direcção de observação.

O recente desenvolvimento dos computadores pessoais conduziu a novas áreas como a multimédia, jogos interactivos e realidade virtual. Para estas novas áreas a interactividade e a animação em tempo real tornou-se fundamental. Como método tradicional para animar personagens articuladamente temos o *Keyframe*. Este método consiste, numa primeira fase, em criar a personagem num programa de modelação (ex. 3D Studio Max) que permita criar animações. Com alguma facilidade o modelador² adiciona, altera ou remove as *frames* que fazem parte da animação da personagem. O movimento em tempo real é obtido através da interpolação das poses necessárias para o estado actual da personagem (Bruderlin e Calvert, 89), (Chung e Hahn, 99), (Marselas, 00), (Watt, 93) e (Witkin e Popovic, 95).

Cinematografia é a arte de fazer filmes. O problema central do realizador é saber como capturar para a filmagem um conjunto de eventos que ocorrem no mundo em simultâneo de modo a que o espectador os consiga perceber com toda a clareza, dando a continuidade pretendida. Arijon tem um papel muito construtivo em que nomeia um conjunto de descrições técnicas para como bem filmar qualquer tipo de situação cinematográfica (Arijon, 76).

Na generalidade os eventos são filmados por várias câmaras em diferentes pontos de vista, sendo o resultado do filme a junção de todas as cenas filmadas. A este processo chama-se edição. Na terminologia cinematográfica um pedaço de filme ininterrupto, filmado por uma única câmara, é designado de *shot ou take*. Os *shots* editados em conjunto formam uma cena, logo um filme é formado por *shots* (Mascelli, 98). O problema que se coloca a um realizador é saber como dispor as câmaras e o equipamento de iluminação para que quando o filme seja editado, tudo junto descreva coerentemente os eventos exigidos. O problema tem duas partes. O primeiro, que pode ser chamado o problema semântico, é assegurar que são incluídas todas as acções do *shot* na ordem correcta. O segundo problema é pragmático, relaciona assuntos de percepção humana e consiste em decidir que propriedades visuais devem estar invariavelmente do outro lado dos cortes para assegurar que o espectador perceba a continuidade.

3. ROBOCUP FUTEBOL ROBÓTICO SIMULADO

O Futebol Robótico (RoboCup) é um campeonato do mundo de futebol para robôs que se realiza todos os anos e que tem como objectivo promover a investigação e integração das áreas da Inteligência Artificial e Robótica nas suas tarefas comuns, nomeadamente na evolução de algumas teorias, algoritmos, aprendizagem e arquitecturas de sistemas multi-agente (Asada et al., 00) (Kitano et al., 95, 97). O campeonato do Futebol Robótico é dividido em

¹ Processo básico de gerar imagens por computador para aplicações em tempo real ou interactivas.

² Pessoa que implementa o modelo e animação num software específico.

várias categorias, sendo que destas todas elas, excepto uma, envolvem o uso de robots reais. A excepção consiste numa plataforma cliente - servidor onde os jogadores são programas que se ligam a um servidor central, o *soccerserver* (Cheny et al., 02), onde toda a partida é simulada. É na categoria simulada que se baseia este visualizador tridimensional. Os clientes do Futebol Robótico são programas que representam os jogadores, que se conectam ao servidor *soccerserver*. As percepções e acções dos jogadores são mapeadas em mensagens enviadas e recebidas através do protocolo de comunicação, num ambiente de rede. Esta arquitectura permite que os vários clientes possam ser escritos em qualquer linguagem de programação, desde que suporte este tipo de comunicação (Noda e Stone, 00).

O *soccerserver* é um servidor que controla um jogo de futebol num campo virtual de acordo com um conjunto de regras do jogo, simulando os movimentos da bola e dos jogadores, e é responsável pela comunicação entre os diferentes agentes do jogo, numa arquitectura de cliente - servidor. Cada cliente controla apenas um jogador que está ligado ao servidor pela rede, usando para isso uma ligação via *socket* UDP/IP (Kitano et al., 95). Todas as acções do agente são tomadas no cliente, sendo a informação enviada por comandos para o servidor. Por sua vez o agente recebe do servidor informação visual, auditiva e física através dos seus sensores de informação (Noda e Stone, 00). O servidor é constituído por três grandes módulos que permitem assim a comunicação perfeita com os clientes (Noda et al., 97). Os módulos são:

- Simulador de campo - módulo que calcula o movimento dos vários agentes e analisa as colisões entre os vários intervenientes no campo;
- Gestor de mensagens - módulo que controla a comunicação de mensagens entre o servidor e os clientes;
- Árbitro - módulo que assegura que as regras do jogo são cumpridas.

O desenvolvimento de visualizadores 3D para as diversas competições do futebol robótico tem sido um desafio de investigação estimulado ao longo dos últimos anos. O domínio preferencial da aplicação dos visualizadores tem sido a liga de simulação. No entanto, nos últimos anos, após o aparecimento do *RoboCup Rescue*³, visualizadores 3D para este domínio de simulação têm também sido desenvolvidos. No que respeita aos visualizadores 3D para a liga de futebol simulado tem existido um evolução no que respeita ao ambiente gráfico, no entanto essa evolução não sido efectuado no controlo de câmara para visualização dos jogos.

³ RoboCup Rescue – Campeonato de salvamento que tem como objectivo estimular a aplicação da investigação, a domínios socialmente mais úteis, no caso, missões de salvamento e resgate em grandes catástrofes.

4. SISTEMAS DE CONTROLO DE CÂMARA INTELIGENTE

Os movimentos das câmaras têm sido estudados em diferentes contextos da computação gráfica, nomeadamente animação, exploração, manipulação e apresentações. As câmaras virtuais são normalmente descritas através de transformações de visualização (Foley et al., 00). Usando a relação geométrica entre o espaço virtual e o plano de visão, Blinn desenvolveu uma técnica de controlo da câmara que automaticamente calcula a posição, orientação e o factor zoom da câmara, através da especificação de dois objectos posicionados no plano de visão (Blinn, 88). Um outro método para controlar uma câmara foi sugerido por (Phillips et al., 92), para ser utilizado na manipulação do sistema Jack. A câmara é posicionada num local visível e num ângulo apropriado para uma manipulação da câmara, utilizando os vários controlos. Karp e Feiner descrevem um sistema para gerar apresentações automáticas, mas eles não consideram no controlo interactivo da câmara (Karp e Feiner, 90).

Seligmann e Feiner usam as restrições de visibilidade não para controlar o posicionamento da câmara, mas para melhorar a *renderização*⁴ da imagem final (Seligmann, 93) (Seligmann e Feiner, 91). Gleicher e Witkin criam um sistema chamado de “*Through-the-lens camera control*” para controlar o movimento da câmara baseado no sistema de projecção do objecto do mundo para o ecrã (Gleicher e Witkin, 92). O sistema CINEMA era uma tentativa inicial a unificar estas várias noções de controlo de câmara. Era um método procedimental para especificar o movimento da câmara baseado nos objectos que fazem parte do mundo virtual (Ducker et al., 92).

William Bares, conjuntamente com James Lester (Bares e Lester, 99), desenvolveu um sistema chamado de *ConstraintCam* (Controlo da câmara através de restrições) que permite aos utilizadores seleccionarem em tempo real quais os objectos a serem vistos, especificar quais as restrições aplicadas a cada objecto e seleccionar qual o tipo de estilo cinematográfico. A medição do grau de satisfação das restrições avaliadas ou resolvidas é importante para determinar se um determinado posicionamento da câmara satisfaz cada uma das restrições. O sistema desenvolvido por He e Christianson em 1996 (Christianson et al., 96) (He et al., 96) chamado de *Linguagem Declarativa de Controlo de Câmara* (DCCL⁵) baseia-se num base de dados de idiomas cinematográficos predeterminados. Um modelo alternativo de idiomas cinematográficos para ser executado em tempo real, nomeadamente em jogos ou simulações, foi desenvolvido por Amerson e Kime (Amerson e Kime, 01) com o nome de *FILM* (*Film Idiom Language and Model*).

⁴ Renderização - Termo aplicado para as diversas formas e metodologias de cálculo de textura, sombra e cores em um conjunto de objectos vectoriais.

⁵ DCCL - *Declarative Camera Control Language*.

O sistema DCCL, onde o controlo de câmara é feito por idiomas que obedecem às restrições, é a grande base conceptual do sistema FILM, apesar do sistema DCCL não ter sido desenhado para trabalhar em tempo real. O modelo de Bares e Lester (Bares e Lester, 99) dá o seu contributo na selecção e composição dos *shots* em tempo real. Para representar a cena através de uma árvore foi usada a noção do sistema Virtual Cinematographer de He, Cohen e Salesien, que explora a cena hierarquicamente para o modelo cinematográfico em tempo real (He et al., 96).

5. SISTEMAS MULTI-AGENTE

A resolução distribuída e cooperativa de problemas fazem apelo a arquitecturas de sistemas apropriados os quais apontam para a noção de Agente e Sistema Multi-Agente (SMA). As noções de Agente e SMA são aplicadas cada vez mais nas diferentes áreas computacionais. É nestes dois conceitos que assenta o visualizador, como base para modelação e resolução dos diferentes problemas associados ao controlo de câmaras inteligentes.

O conceito de agente é uma metáfora utilizada em inúmeras áreas do conhecimento, desde a Psicologia, Sociologia, até as Ciências de Computação. Foi a comunidade de Inteligência Artificial, pelo facto de basear-se em metáforas psicológicas e sociais como forma de inspiração e de motivação, quer como forma de modelação do processamento (complexo) da informação, que optou por este conceito. Esta primazia está associada a Carl Hewitt pela sua proposta do Sistema de actor (Hewitt, 77). Segundo Pattie Maes, as arquitecturas são especificações de como se pode construir agentes através de módulos e como estes interagem entre si. Os módulos e interacções existentes entre os mesmos devem especificar como, a partir dos dados dos sensores e do estado actual do agente, são obtidas as acções e o novo estado interno (Maes, 96). Um aspecto importante num agente é a sua arquitectura interna. A arquitectura interna de um agente está associada à sua própria definição e mecanismos de decisão do agente que determina a influência de actuação individualmente e na própria arquitectura do sistema multi-agente. A Inteligência Artificial classifica os agentes computacionais em três grandes categorias: agentes deliberativos, reactivos e híbridos (Wooldridge, 98):

- *Agentes deliberativos* – Abordagem clássica onde os agentes actuam com pouca autonomia, com uma visão “*top-down*” e possuindo modelos simbólicos explícitos para verificar informação e gerar acções no seu ambiente através de raciocínio lógico;
- *Agentes reactivos* – Abordagem que procura não utilizar um modelo ou raciocínio simbólico complexo, comportando-se autonomamente no ambiente inserido em que os comportamentos inteligentes emergem da dinâmica das interacções com o contexto (Brooks, 91);

- *Agentes híbridos* – Combinam as características das duas abordagens anteriores, “retirando” os seus pontos fracos.

A arquitectura BDI (“*Belief-Desire-Intention*”) é uma arquitectura essencialmente deliberativa (Rao e Georgeff, 91), na medida em que se baseia em modelos simbólicos explícitos e num raciocínio lógico. O modelo de cognição do agente é descrito através de um conjunto de “estados mentais”: crenças (“*Belief*”), desejos (“*Desire*”) e intenção (“*Intention*”), denominada de arquitectura BDI. A fundamentação filosófica para esta concepção de agentes vem do trabalho de Dennett (Dennett, 87) sobre sistemas intencionais e de Bratman sobre raciocínio prático (Bratman et al., 88).

A investigação em Sistemas Multi-Agente derivou da Inteligência Artificial Distribuída, constituindo actualmente o núcleo deste campo. Inclui a investigação em diversas áreas científicas, sendo extremamente abrangente. A investigação está focada para o desenvolvimento de princípios computacionais, construção de modelos para descrever, implementar e analisar as formas de interacção e coordenação de agentes em sociedades de reduzida ou elevada dimensão (Lesser, 99). Um sistema Multi-Agente é um sistema computacional no qual dois ou mais agentes interagem ou trabalham em conjunto num conjunto de tarefas ou satisfazem um conjunto de objectos a que são propostos. (Lesser, 99). Os Sistemas Multi-Agente incluem diversos agentes que interagem ou trabalham em conjunto, podendo compreender agentes homogéneos ou heterogéneos. Uma agente no sistema é considerada uma parte da resolução do problema, operando assincronamente com uma certa autonomia respeitando sempre os outros agentes do sistema. O SMA contém vários agentes, cada qual com diferentes capacidades de percepção e acção no mundo, tendo uma esfera de influência distinta sobre o ambiente, sendo capaz de influenciar outras organizações ou partes do ambiente (Jennings, 99).

5. ARQUITECTURA DO SISTEMA

5.1 Descrição

O sistema foi desenvolvido numa arquitectura Multi-Agente. Para implementar esta arquitectura foi desenvolvido uma ferramenta designada de MAS (*Multi-Agent System*), utilizando o paradigma de *object oriented*. São cinco as componentes que suportam esta arquitectura: renderização gráfica 3D utilizando a biblioteca gráfica RenderWare (Criterion, 95), motor de som 3D para público, árbitro e jogadores, comunicação, animação 3D e agentes. Esta arquitectura é apresentada na figura 1.

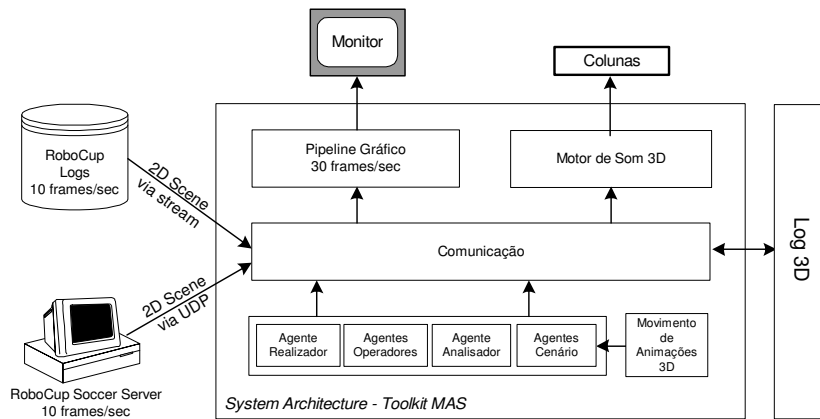


Fig. 1. Arquitectura do sistema

5.1 Controlo de câmara inteligente

O controlo da câmara baseia-se no modelo matemático de Blinn em que as câmaras estão automaticamente a focar a bola no centro do ecrã (Blinn, 88). Para posicionar e orientar as câmaras (*pan, tilt, zoom*) de forma a acompanhar a direcção da bola no mundo virtual é utilizado um conjunto de primitivas de câmara procedimentais sugeridas por Ducker (Ducker e tal., 92). O sistema desenvolvido utiliza um conjunto de restrições sugeridas por William Bares que permite especificar as restrições aplicadas aos objectos e a projecção em tempo real (Bares e Lester, 99). O modelo de percepção e intenções da filmagem segue o modelo de Amerson e Kime (Amerson e Kime, 01) em que o realizador tenta que sejam cumpridas as intenções da filmagem, que serão enviadas para os *cameraman* via mensagem, em que este é o responsável pelas configurações das câmaras.

Para medir o grau da qualidade da imagem projectada por cada câmara foi criada uma fórmula que tem por base o princípio da medição do grau de satisfação das restrições sugerida no trabalho de William Bares. Esta fórmula permite integrar a prioridade do objecto (P_i) a ser visualizado, os objectos com maior relevância têm prioridade mais alta para que sejam contemplados na cena a visualizar (valor atribuído pelo realizador); o factor (D_i) quantifica a distância do objecto ao ponto de focagem (normalmente a bola). Qualidade de visualização (Q_i) do objecto no ecrã, quantifica como o objecto ocupa o ecrã na visualização. O objecto pode estar descentrado, mas o que interessa neste caso é se o objecto está a ser visualizado com qualidade.

$$Qualidade\ da\ imagem = \sum_{I=1}^N P_i * D_i * Q_i$$

n = número de objectos a analisar.

6. COMPONENTES DO SISTEMA

6.1 Componente de agentes

A componente de agentes tem os modelos de três tipos de agentes: reactivos, deliberativos e emocionais. Nos agentes deliberativos temos os operadores de câmara (*cameraman*), que utilizam um arquitectura BDI. O *cameraman* tem como função tentar focar o jogo na melhor perspectiva de visualização, através de primitivas de focalização da câmara. Cada *cameraman* calcula as várias medidas de qualidade da visualização, sendo enviada a medida da qualidade para o agente deliberativo Realizador, que é o responsável por escolher qual a melhor câmara para visualizar o cenário num determinado momento do jogo. É o agente realizador que envia as coordenadas de posicionamento das câmaras como as regiões do campo de maior prioridade para a filmagem no decorrer do jogo.

Os agentes do cenário são compostos por jogadores, público e bola. Os jogadores são modelados como agentes emocionais, pois são emocionalmente afectados pelo seu comportamento no campo (durante o jogo). Eles também são agentes reactivos simples porque são guiados pelas coordenadas bidimensionais enviadas pelo servidor *soccerserver* que são convertidas em coordenadas tridimensionais. O público também é modelado como agente emocional. Isto permite implementar a satisfação emocional por equipa, sendo a satisfação expressa através do som. A bola é um agente reactivo simples que converte o seu sistema de coordenadas bidimensionais para tridimensionais.

6.2 Movimento e animações 3D

O sistema de animação 3D é criado tendo por base a posição, velocidades, direcção, energia e orientação do pescoço de todos os jogadores que estão a correr no mundo virtual a duas dimensões do servidor *soccerserver* em tempo real ou diferido. O sistema de animação dos jogadores utiliza a aproximação *keyframe* que já foi explicada anteriormente. Várias animações foram previamente criadas para cada situação do jogador: parado, andar, correr, chutar com pé esquerdo, etc. É de salientar que também

foram criados os estados intermédios entre os diferentes movimentos para que a animação não apresente “saltos” nas transições de movimentos. A sequência de animação a executar depende da acção do jogador e da sua envolvente e percepção do jogo.

6.3 Componente gráfica

É a componente gráfica que permite criar ambientes virtuais 3D cada vez mais realistas. Para responder à necessidade da visualização utilizou-se uma biblioteca gráfica RenderWare (Criterion, 95), que permite criar cenários em tempo real com alguma qualidade e com um desempenho muito bom, acima das 30 imagens por segundo. A componente gráfica tem a responsabilidade de gerar o *output* 3D do cenário a partir dos vários estados dos agentes envolvidos e objectos estáticos do mundo virtual. Ver figura 2 e 3.



Fig. 2. Imagem do visualizador utilizando uma câmara de topo tipo *fly*.



Fig. 3. Imagem do visualizador utilizando uma câmara estática lateral.

6.4 Componente de comunicação

Actualmente o tempo de ciclo do servidor do futebol simulado, *soccerserver*, é de 100 milissegundos (Cheny et al., 02). Para melhorar o tempo de comunicação entre os agentes existe a necessidade de criar uma componente de comunicação, independente do servidor *soccerserver*, que permita aos agentes

trocarem mensagens entre si e que não dependam do tempo de amostragem dos movimentos referentes ao servidor. Para uma boa comunicação recomenda-se um *frame rate* aplicacional superior a 30 imagens por segundo. A componente serve assim para a troca de mensagens entre os vários agentes, como também para as diferentes mensagens aplicacionais.

6.5 Motor de som tridimensional

A geração de sons do jogo depende dos agentes jogadores, público e árbitro. O sistema emocional dos vários agentes permite assim criar um ambiente sonoro o mais realista possível tendo em conta o desenvolvimento do próprio jogo. O motor de som posicional tridimensional é suportado pela tecnologia DirectX (Root e Boer, 99).

7. RESULTADOS E DISCUSSÃO

Os espectadores confirmaram que os visualizadores do RoboCup da liga simulada com controle de câmara inteligente lhes dá uma sensação de assistir a uma partida real de futebol como na televisão. Acreditamos que estes resultados são motivados pelo ambiente virtual tridimensional que combina um ambiente realístico (inclusive um estádio), animações (dos jogadores, bola e público), efeitos sonoros e controle da câmara inteligente utilizando um arquitectura multi-agente, como acontece na realidade de uma filmagem.

REFERÊNCIAS

- Amerson, Dan; Kime, Shaun, (2001), “*Real-time Cinematic Camera Control for Interactive Narratives*”, Notas de trabalho para: AAAI Spring Symposium on Artificial Intelligence e Interactive Entertainment, Stanford, California.
- Angel, Edward, 2002, “*Interactive Computer Graphics: A Top-Down Approach with OpenGL*”, 2nd Edition, Addison Wesley.
- Asada, M.; Veloso, M.; Tambe, M.; Noda, I.; Kitano, H., Kraetzschmar, G.K., (2000), “*Overview of RoboCup-98 (2000)*”, notas de leitura na *Computer Science*.
- Bares, William.; Lester, J., (1999), “*Intelligent Multi-Shot Visualization Interfaces for Dynamic 3D Worlds*”, Proceedings of International Conference on Intelligent User Interfaces.
- Blinn, Jim, (1988), “*A Trip Down the Graphics Pipeline*”, Morgan Kaufmann, pp. 69-82 – Capítulo 8 - “*Where am I? What am I looking at?*”.
- Bratman, Michael E.; Israel, David J.; Pollack, M. E., (1998), “*Plans and resource-bounded practical reasoning*”. *Computational Intelligence*, 349-355.
- Brooks, Rodney A., (1991), “*Intelligence Without Reason*”, Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91).
- Bruderlin, A. e Calvert, T. W., (1989), “*Goal-Directed, Dynamic Animation of Human Walking*”. Proc. ACM, SIGGRAPH 89, Boston.
- Cheny, Mao; Foroughi, Ehsan; Heintz, Fredrik; Huangy, ZhanXiang; Kapetanakis, Spiros; Kostiadis, Kostas; Kummeneje, Johan; Noda, Itsuki; Obst, Oliver; Riley, Pat; Steens, Timo; Wangy, Yi e Yiny, Xiang, (2002), “*RoboCup Soccer Server - for Soccer Server Version 7.07 and later*”.

- Chung, Shih-kai; Hahn, James K., (1999), "Animation of Human Walking in Virtual Environments".
- Christianson, D.; Anderson, Sean; He, Li-wei, Salesin, D.; Weld, D.; Cohen, M., (1996), "Declarative Camera Control for Automatic Cinematography", Proceedings of the AAAI-96.
- Criterion Software, (1995), "Renderware – Real-time 3D graphics library", API Reference Manual, Canon Company.
- Dennett, Daniel C., (1987), "The Intentional Stance", The MIT Press, Cambridge, MA.
- Drucker, Steven M.; Galyean, Tinsley A.; Zeltzer, David, (1992), "CINEMA: A System for Procedural Camera Movements", Computer Graphics and Animation Group, MIT Media Lab, SIGGRAPH Symposium on 3D Interaction. Cambridge, MA.
- Drucker, Steven M.; Zeltzer, David, (1995), "CamDroid: A System for Implementing Intelligent Camera Control", Computer Graphics and Animation Group, MIT Media Lab, SIGGRAPH Symposium on Interactive 3D Graphics.
- Drucker, Steven M.; Zeltzer, David, (1996), "Intelligent Camera Control in a Virtual Environment", Computer Graphics and Animation Group, MIT Media Lab.
- Foley, James D.; Dam, Andries Van; Feiner, Steve; Hughes, Jonh F., (2000), "Computer Graphics Principles and Pratices", Second Edition in C, Addison Wesley.
- Gleicher, M.; Witkin, A., (1992), "Through-the-Lens Camera Control", Processdings of SIGGRAPH 92.
- He, Li-Wei; Cohen, Michael F.; Salesin, David H., (1996), "The virtual cinematographer: a paradigm for automatic real-time camera control and directing", Proceedings of SIGGRAPH 96, in Computer Graphics Proceedings, Annual Conference Series.
- Carl Hewitt, (1997), "Viewing Control Structures as Patterns os Passing Messages". Artificial Intelligence.
- Jennings, Nick R., (1999), "Agent-Base Software Engineering", Proceedings of the 9th European Workshop on Modelling Autonomous Multi-Agent System Engineering.
- Jung, Bernhard; Oesker, Markus; Hecht, Heiko, (1999), "Virtual RoboCup: Real-Time Visualization of 2D Soccer Games", Universidade de Bielefeld, Alemanha.
- Karp, P., Feiner, S., (1990), "Issues in the Automated Generation of Animated Presentations", In Proceedings of Graphics Interface '90, pp.39-48.
- Kitano, Hiroaki; Asada, Minoru; Kuniyoshi, Yasuo; Noda, Itsuki; Osawa, Eiichi, (1995), "RoboCup: The Robot World Cup Initiative", Proceedings of the First International Conference on Autonomous Agents (Agents'97).
- Kitano, Hiroaki, Tambe, Milind, Stone, Peter, (1997), "The RoboCup Synthetic Agent Challenge 97", International Joint Conference on Artificial Intelligence (IJCAI97).
- Lesser, Victor, (1999), "Cooperative Multi-Agent Systems: A Personal View of the State of the Art", IEEE Transactions on Knowledge and Data Engineering, Vol. 11, Nº 1.
- Li, Qingrui; Huang, Zhanxiang; Sun, Pen ; Chen, Xiaopin, (2003), "WrightEagle 2003 Presentation Description: 3D Monitor with Automatic Commentary and Graphical Debugging System", AI Center, University of Science and Technology of China, Descrição da equipa para o RoboCup-03.
- Maes, Pattie, (1996), "Intelligent Software: Programs That Can Act Independently Will Ease the Burdens that Computers Put on People", IEEE Expert Systems, Vol. 11, No. 6, pp. 62-63.
- Marselas, Herbert, (2000), "Interpolated 3D Keyframe Animation", Game Programming Gems, Charles River Media, pp. 465-470.
- Mascelli, Joseph V., (1998), "The five C's of cinematography" Motion picture filming techniques, Silman-James Press.
- Noda, Itsuki; Matsubara, Hitoshi; Hiraki, Kazuo; Frank, Ian, (1997), "Soccer Server: a tool for research on multi-agent systems", Electrotechnical Laboratory - Umezono, Tsukuba, Japan.
- Noda, Itsuki; Frank, Ian, (2000), "Review: RoboCup Through 2000", Segunda conferência internaional de computação gráfica (CG2000).
- Noda, Itsuki; Stone, Peter, (2000), "The RoboCup Soccer Server and CMUnited Clients: Implemented Infrastructure for MAS Research", Nota no Workshop 13 on Infrastructure for Scalable Multi-agent Systems..
- Nieuwenhuisen, Dennis ; Overmars, Mark H. , (2003), "Motion Planning for Camera Movements in Virtual Environments", Institute of Information and Computing Sciences, Utrecht University.
- Phillips, C. B.; Badler, N. I.; Granieri, J., (1992), "Automatic viewing control for 3D direct manipulation", 1992 Symposium on Interactive 3D Graphics, ACM Press.
- Rao, A. S. e Georgeff, M. P., (1992), "Modeling rational agents within a BDI-architecture", In Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91).
- Root, Michael e Boer, James, (1999), "DirectX Complete", McGraw-Hill.
- Seligmann, D. D. ; Feiner, S., (1992), "Automated of intent based 3D illustrations", Computer Graphics, 25, pp.123-132.
- Seligmann, D. D., (1993), "Interactive Intent-Based Illustration: Language for 3D Worlds", Tese de Douturamento. Graduate School of Arts and Sciences, Universidade de Columbia, EUA.
- Spoelder, Hans J.W.; Renambot, Luc; Germans, Desmond; Bal, Henri E.; Groen, Frans C.A., (1999), "Interactive and Collaborative Visualization of Robot Soccer", Vrije Universidade de Amesterdão, Holanda.
- Spoelder, Hans J.W.; Renambot, Luc; Germans, Desmond; Bal, Henri E.; Groen, Frans C.A., (2000), "Real-time Interaction in VR with a Distributed Multi-Agent System", Vrije Univ. de Amesterdão, Holanda.
- Watt, A., (1993), "3D Computer Graphics" Addison Wesley, 2nd Edição.
- Witkin, Andrew; Popovic, Zoran, (1995), "Motion Warping", SIGGRAPH 95, Los Angeles, COMPUTER GRAPHICS Proceedings.
- Michael Wooldridge. (1998), "Agent-based computing". Interoperable Communication Networks, 1(1):71-97.
- Xing, Xuan; Li, Qingrui; Liu, Chang; Yang, Bin; Chen, Xiaopin, (2002), "An Intuitive 3D Monitor System with Automatic Commentary for RoboCup 2002", AI Center, University of Science and Technology of China, Descrição da equipa para o RoboCup-02.

