

Animando Humanos Virtuais em Tempo-Real usando Visão Computacional e Redes Neurais

Nelson F. Souza Jr
Mestrado em Computação
Aplicada
Universidade do Vale do Rio
dos Sinos
Av. Unisinos 950, 93022-000.
São Leopoldo, RS
juniornelson@gmail.com

Thiago Kehl
Mestrado em Computação
Aplicada
Universidade do Vale do Rio
dos Sinos
Av. Unisinos 950, 93022-000.
São Leopoldo, RS
thiagokehl@gmail.com

Fernando S. Osório
Mestrado em Computação
Aplicada
Universidade do Vale do Rio
dos Sinos
Av. Unisinos 950, 93022-000.
São Leopoldo, RS
fosorio@unisinos.br

Cláudio R. Jung
Mestrado em Computação
Aplicada
Universidade do Vale do Rio
dos Sinos
Av. Unisinos 950, 93022-000.
São Leopoldo, RS
crjung@unisinos.br

Soraia R. Musse
PUCRS – FACIN - Faculdade
de Informática
Av. Ipiranga, 6681 - Prédio 32
- Sala 609
90619-900 - Porto Alegre - RS
soraia.musse@pucrs.br

ABSTRACT

This paper proposes a model for animating groups of virtual humans in real time. The control is based on hand postures, identified through neural networks. This research also includes topics of computer vision and image processing, that are explored to extract information captured by the camera and feed them to the neural network. The result of the neural network is sent to a client-server system, and they become movement stimuli for tool to animate groups of virtual humans. This paper describes technical details of this model, presents some experimental results, and points towards future work.

RESUMO

Este artigo propõe o desenvolvimento de um modelo para animação de grupos de humanos virtuais em tempo real, controlados através de posturas de mão, que são identificadas por Redes Neurais. O trabalho também inclui pesquisa na área de visão computacional para detecção de características de imagens que são enviadas a redes neurais para treinamento e posterior identificação de posturas. Estas são enviadas, em um sistema cliente-servidor, para serem estímulos de movimento a uma ferramenta de animação de grupos de humanos virtuais. Este artigo descreve detalhes técnicos deste modelo, apresenta os resultados obtidos e discute trabalhos futuros.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Symposium on Virtual Reality '2006 Petrópolis, Brazil
Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...\$5.00.

Keywords

Redes neurais, visão computacional, humanos virtuais.

1. INTRODUÇÃO

A área de Segurança e Defesa, em países como EUA e Inglaterra, recebe grandes esforços de financiamento, principalmente no que tange investimento em tecnologia de simulação e visualização. No Brasil, as iniciativas ainda são bastante incipientes, e necessitam de grande investimento tecnológico. No contexto de visualização e computação gráfica, diferentes cenários e situações podem ser simulados em tempo-real, bem como em treinamentos e capacitação de pessoal, provendo alto grau de realismo. Os principais benefícios são o baixo risco das simulações comparadas com o treinamento e capacitação in loco [2]. Para tanto, é necessário desenvolver ferramentas adequadas para interface homem-máquina com dispositivos de visualização e interação 3D como por exemplo: sistema de visualização estereoscópica (projeção e HMD), mouse 3D (spaceball), luvas 3D, gyromouse, sistemas de rastreamento e resposta tátil, bem como com dispositivos convencionais de E/S. A possibilidade de utilização desses dispositivos em aplicações para simulação de evacuação, além de permitir a visualização realística e portanto, mais eficaz para treinamento, abre um largo espectro de desenvolvimento de aplicações [1].

Este artigo visa descrever um modelo que utilize visão computacional e redes neurais para detecção de posturas de mão, sem necessidade de utilização de quaisquer artefatos como luvas, dedais ou outros padrões. Este modelo é integrado em um sistema cliente-servidor, onde as posturas detectadas são enviadas a uma ferramenta de animação de comportamentos estruturados de humanos virtuais. O restante deste artigo é organizado conforme segue: a seção 2 apresenta alguns trabalhos encontrados na literatura; a seção 3 descreve detalhes do modelo utilizado para detecção de características da imagem e o processamento, utilizando redes neurais, enquanto a seção 4 descreve a ferramenta para animação de grupos de humanos virtuais. Finalmente a seção 5 apresenta a arquitetura do protótipo e alguns resultados obtidos.

2. TRABALHOS RELACIONADOS

A identificação de padrões em imagens tem sido assunto bastante abordado pela comunidade científica. Pesquisadores de Visão Computacional e de Inteligência Artificial tem se esforçado no sentido de propor soluções para problemas cotidianos. Muitos trabalhos destas duas linhas de pesquisa são conhecidos, e relacionados à finalidade de extrair gestos e/ou posturas de mãos com algum propósito específico.

Nos últimos anos inúmeras técnicas integrando Redes Neurais Artificiais e Visão Computacional vêm sendo utilizadas em larga escala. São aplicadas em diagnóstico médico, na robótica, em sistemas de segurança patrimonial, na classificação de imagens, entre outros. Isso se dá, pela facilidade da aplicação das técnicas de inteligência artificial em tarefas de classificação.

Em particular, o escopo de interesse deste trabalho concentra-se em gestos e/ou posturas de mãos capturadas em vídeo, e processadas a ponto de possibilitar a interação com multidões, em ambiente virtual. Algumas dessas pesquisas baseadas em gestos possuem elementos de marcação nas mãos. Luvas de realidade virtual e outros tipos de marcadores [13, 15] são usados pelo usuário a fim de detectar gestos e posição da mão em relação a pontos de referência. De fato, pesquisas relacionadas a dispositivos óticos unidos a sensores que ditavam posição e movimentação com boa performance foram exaustivamente estudados no passado. Dada a extensa bibliografia na área, cita-se apenas alguns trabalhos que visam a cobrir a abrangência da área.

Musse e Osório, 2000 [7], apresentam um sistema para fornecer a interação facilitada do usuário com multidões de humanos virtuais, e detectam as posturas das mãos com a ajuda de uma *DataGlove*, onde a Rede Neural Artificial é responsável pela classificação e reconhecimento da postura.

Outros trabalhos são realizados sem a necessidade de vestir-se as mãos com dispositivos quaisquer. Um exemplo é o método proposto por Sato et al [14], onde descreve-se um modelo para seguir a movimentação da mão do usuário, em 3D, e reconhecer o gesto da mão em tempo real, sem o uso de dispositivos conectados. Este trabalho utiliza múltiplas câmeras determinando a posição e a orientação da mão do usuário. O método identifica gestos pré-determinados usando uma rede neural treinada anteriormente.

Mackie e McCane, 2004 [5] apresentam um sistema capaz de reconhecer gestos e posturas em mãos livres. É um sistema de estimativa de posições que pode ser aplicado a Interfaces Virtuais e Interfaces de Realidade Aumentada. Utilizam árvores de Decisão [9] em um ambiente de aprendizado Weka¹.

Este artigo visa capturar posturas de mãos, a partir de uma única câmera de topo. Em seguida, pré-processar as imagens obtidas a fim de extrair delas as entradas que deverão ser processadas por uma Rede Neural que, devidamente treinada, será capaz de classificar as diferentes classes de imagens obtidas. Uma vez as posturas de mão capturadas, estas imagens e suas classificações deverão ser entrada de um sistema de simulação aplicado a serious games, para controlar o movimento de non-player characters (NPCs).

3. DETECÇÃO DE POSTURAS USANDO VISÃO COMPUTACIONAL E REDES NEURAIS

Esta seção descreve o modelo apresentado neste artigo para a detecção de posturas de mão utilizando técnicas de visão computacional para a extração de feições e redes neurais para a classificação das posturas.

¹<http://www.cs.waikato.ac.nz/ml/>

3.1 Extração de Feições usando Visão Computacional

A idéia desta etapa é capturar imagens de mãos postadas sobre uma superfície em um ambiente controlado, através de câmeras fotográfica digitais ou webcams, e extrair um vetor de feições para cada postura de mão para posterior reconhecimento. No ambiente controlado, a superfície deve ser branca (ou clara), e o usuário deve utilizar uma camisa de manga comprida de cor branca. A câmera está instalada acima da superfície, provendo uma visão de topo da cena. Exemplos de posturas a serem reconhecidas pelo sistema proposto são ilustradas na Figura 1.

Uma vez capturada a imagem colorida, é necessário diferenciar a região da mão do restante da imagem. Para tal, foi utilizado apenas o canal de cor azul da câmera, que fornece uma baixa reflectância da cor de pele com relação ao fundo claro. Assim, há uma boa discriminação entre as intensidades da pele e do restante da imagem, indicando que um simples limiar é suficiente para identificar a região da mão. Foi utilizado o limiar de Otsu [8], que é calculado adaptativamente com base no histograma da imagem adquirida.

Embora o canal azul forneça uma boa discriminação entre a cor da pele (mão) e o restante do fundo, sombras e ruído geram intensidades parecidas com o tom da pele em regiões espúrias da imagem (particularmente, nas bordas entre o braço e a pessoa). Para remover tal ruído, foi aplicado um fechamento morfológico com elemento estruturante de formato diamante, com raio de três pixels (o tamanho do elemento estruturante foi escolhido com base nas dimensões esperadas da mão capturadas com resolução de 320×240 , que é a resolução utilizada na captura das imagens).

Mesmo com o fechamento morfológico, podem restar algumas pequenas regiões da imagem erroneamente classificadas como mão. Entretanto, tais regiões são bem menores do que a área compreendida pela mão, e então a mão pode ser considerada como o elemento conexo da imagem com maior área. A partir daí, pode-se extrair o *bounding box* da mão, que será utilizado para a obtenção do vetor de feições para a classificação das posturas. Os passos descritos para a extração da imagem binária da mão são ilustrados na Figura 2. Figura 2(a) mostra o canal azul da imagem adquirida, e a Figura 2(b) ilustra a binarização. O resultado do fechamento morfológico é exemplificado na Figura 2(c), e o resultado final (restrito ao *bounding box*) é mostrado na Figura 2(d).

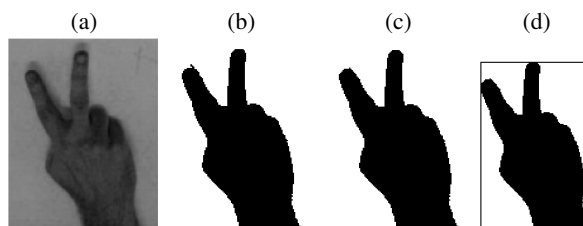


Figura 2: Exemplo de extração de feições. (a) Intensidade do canal azul. (b) Resultado da binarização. (c) Resultado do fechamento morfológico. (d) Imagem binária final restrita ao *bounding box*.

O último procedimento desta etapa é então obter os vetores de feição para cada postura. O importante nesta etapa é que as feições escolhidas sejam similares entre si para posturas da mesma classe, e ao mesmo tempo discriminantes para posturas distintas. Como a propriedade mais discriminante entre as posturas é a posição dos dedos, optou-se por calcular as projeções vertical e horizontal da



Figura 1: Exemplos de posturas a serem reconhecidas.

imagem binarizada resultante internamente ao bounding box. Foram utilizados nos experimentos dez intervalos na discretização das projeções verticais e horizontais, gerando então dez valores para cada projeção (em x e em y). Além disso, as dimensões do bounding box (largura e altura) também são incluídas no vetor de feições, principalmente para discriminar a postura de mão fechada das demais. A Figura 3 ilustra as projeções vertical e horizontal para a imagem binária mostrada na Figura 2(d).

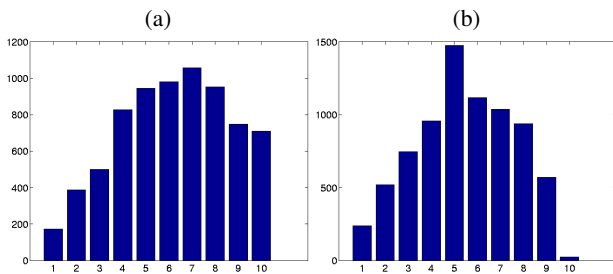


Figura 3: Elementos que formam o vetor de feições. (a) Projeção vertical. (b) Projeção horizontal.

Tendo armazenadas as informações sobre as dimensões x e y do bounding box, e os vinte valores extraídos através das projeções vertical e horizontal da imagem, pode-se concatená-los em um vetor colecionando vinte e duas informações. Este vetor deve ser submetido à Rede Neural Artificial como entradas, que são resultantes do pré-processamento das posturas das mãos adquiridas em tempo real através de uma webcam.

3.2 Classificação das Posturas através de Redes Neurais

A partir das informações de interesse extraídas de cada uma das imagens (10 projeções verticais, 10 projeções horizontais e as dimensões da largura e altura do bounding box), pode-se então criar a base de treinamento do classificador, contendo estas 22 entradas e a respectiva classe (1 dentre as 6 posturas pré-definidas) a qual pertence cada uma das posturas de mão capturadas. A fim de criar um classificador automático para as imagens de posturas de mão foi adotado o uso de uma ferramenta de Machine Learning (aprendizado de máquina): Rede Neural Artificial - RNA [10, 6].

Normalmente as Redes Neurais Artificiais (RNAs) são utilizadas em aplicações de aprendizado indutivo supervisionado (através de exemplos), sendo amplamente adotadas em problemas de reconhecimento e classificação de padrões. As Redes Neurais Artificiais são inspiradas no funcionamento biológico dos neurônios humanos, onde possuem pesos sinápticos que se adaptam de forma a reforçar ou enfraquecer as ligações entre as entradas e saídas

dos neurônios. Os pesos sinápticos permitem estabelecer “ponderações” sobre como a rede irá classificar os padrões apresentados em suas entradas. As RNAs possuem diversas características, dentre elas a capacidade de aprender relações complexas não lineares, além de sua capacidade de generalização do aprendizado, obtido a partir do ajuste dos seus pesos de modo a se adaptar aos padrões de treinamento.

Neste trabalho, foi utilizada durante os experimentos uma Rede Neural Artificial de arquitetura MLP - Multi Layer Perceptron, com conexões do tipo feed-forward, por ser a um dos modelos mais adequados ao treinamento e reconhecimento/classificação de padrões [12, 3]. Uma rede MLP possui pelo menos três camadas, sendo uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. O aprendizado da rede é baseado nas informações de entrada e na utilização de um algoritmo de aprendizagem. Este algoritmo de aprendizagem utiliza os padrões dados como entradas (informações extraídas das imagens) e as saídas desejadas (classes), de modo a ajustar os pesos da rede através de um mecanismo de correção de erros. Os erros são estimados obtendo a diferença entre o resultado atual fornecido pela RNA e o resultado que era esperado (saída desejada). Através da medida deste erro é possível estimar o desempenho da rede, bem como realizar o ajuste dos seus pesos sinápticos de modo a buscar a minimização deste erro.

A Rede Neural Artificial foi modelada a partir do uso do Software JavaNNS [4]. Este programa possibilita uma melhor visualização da arquitetura de rede neural, auxiliando graficamente na configuração dos parâmetros utilizados nas simulações, na realização do aprendizado e facilitando na visualização dos logs e dos gráficos do erro de aprendizado (e.g. MSE - Mean Squared Error, SSE - Sum of Squared Error).

No problema tratado neste artigo, o conjunto de dados é muito grande e contém uma grande quantidade de exemplos. Foram gerados 600 vetores de características (seiscentas imagens de posturas de mão), em dois momentos distintos. No primeiro momento, são capturadas as 420 posturas relativas a 70% do conjunto total dos dados para compor a base de treino da rede neural. Os outros 30% dos dados, composto por mais 180 imagens relativas à base de dados de validação, são adquiridos em momento posterior e mantidos separados dos dados de treino. As imagens de treino foram usadas na fase de ajuste dos pesos da rede, e as imagens de validação foram usadas somente para avaliar a rede (cross-validation), uma vez terminado o aprendizado. Cada uma das 6 posturas (classes) possui um número igual de exemplos em cada uma das bases, onde na base de treino temos 6×70 imagens de cada classe = 420 exemplos.

Neste estudo adotou-se uma rede 22-22-6, por ter apresentado os melhores resultados em relação às outras topologias testadas previamente. Nesta rede, 22 neurônios estão dispostos em sua camada de entrada, 22 neurônios na sua camada oculta ou intermediária e seis neurônios formam a saída da rede neural artificial. A camada

de saída possui um neurônio especializado em identificar cada uma das 6 classes. Para o processo de aprendizagem foi escolhido o algoritmo de aprendizagem supervisionado RPROP - Resilient Propagation [11], baseado no conhecido Back-Propagation, por apresentar melhor desempenho em um número menor de épocas. Para estimar e validar significativamente as taxas de acertos/erros da rede foi utilizado o método de validação cruzada, conhecido como cross-validation, onde usamos uma base de treino com 420 exemplos e uma base separada de validação com 180 exemplos.

A Rede foi treinada pelo software JavaNNS a partir da base de dados de treinamento, utilizando-se apenas 4000 épocas. Como o processo de treinamento é não determinístico, foram rodadas dez execuções diferentes, cada uma utilizando uma semente randômica diferente. Este procedimento deu origem as 10 redes neurais treinadas, que foram submetidas posteriormente à uma avaliação dos seus resultados pelos softwares SNNS e Analyze. A topologia da rede é ilustrada na Figura 4, e a evolução do erro médio quadrático para as 2000 épocas iniciais é mostrada na Figura 5.

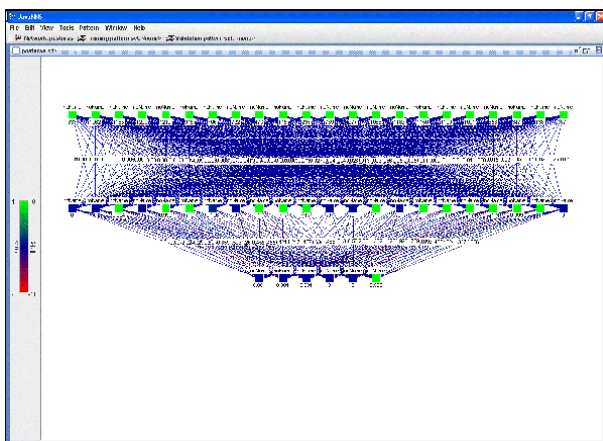


Figura 4: Topologia da rede 22-22-6 criada a partir do JavaNNS

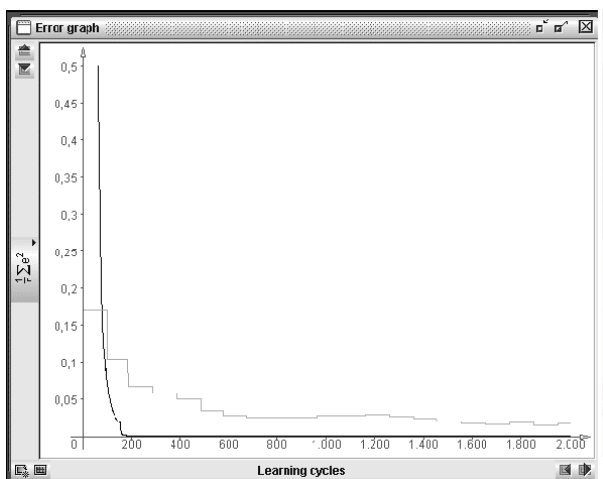


Figura 5: Gráfico da evolução do erro médio quadrático para as 2000 épocas iniciais.

A tabela 1 mostra os resultados obtidos através das dez exe-

cuções nas etapas de treinamento da RNA, obtendo assim 10 diferentes redes neurais treinadas.

Após a obtenção das redes neurais devidamente treinadas, foi submetida a cada uma delas a base de dados de teste/validação. O objetivo foi o de verificar a capacidade de generalização das redes e assim contabilizar o número de erros e acertos a partir dos padrões previamente definidos que não haviam ainda sido apresentados à rede na fase de treinamento.

A tabela 2 mostra os resultados obtidos através das dez execuções na etapa de validação/testes da RNA.

Observando os resultados conseguidos através das diferentes rodadas de execução pode-se concluir que a RNA utilizada nesta técnica obteve excelente capacidade de aprendizado/identificação dos padrões submetidos a ela. A taxa média de aprendizado (T_a) obtida na etapa de treinamento foi de 99,952 % e a taxa média de erro se manteve abaixo de 0,05 %. Cabe destacar que a taxa de acertos tanto no treino quanto na validação se mostrou muito próxima a 100 %, o que indica o sucesso do aprendizado e a grande capacidade de generalização do método proposto.

Após a análise dos resultados obtidos a rede identificada na rodada 6 foi selecionada para ser colocada em produção pelo método, uma vez que apresenta o melhor desempenho tanto na fase de treinamento quanto na validação (menor MSE em ambos e 100% de acerto em ambos).

4. MODELO PARA PROVER COMPORTAMENTOS ESTRUTURADOS DE HUMANOS VIRTUAIS

Com o intuito de simular-se diferentes movimentações de humanos virtuais em ambientes povoados, definiu-se uma série de comportamentos estruturados, como filas e formações geométricas que os grupos de humanos podem adotar em função do tempo. Para isto, este trabalho propõe os “locais de ação” (AL - Action Location), que são estruturas simples que descrevem a dinâmica das formações e movimentações dos humanos nos ambientes virtuais.

Definição 1 Os locais de ação (AL's) são delimitados por polígonos, descritos através de um conjunto de pontos (x, y, z) , conforme ilustrado na figura 6.



Figura 6: Vista de cima de uma formação de humanos virtuais, visualizando o polígono.

Os AL's são instanciados em locais específicos no ambiente virtual, e descritos através das coordenadas (x, y, z) do seu centro.

Base dados de treino - 420 padrões			
Execução	Percentual de aprendizado	Padrão não reconhecido	Erro
1	100.00 % (420 pattern(s))	0.00 % (0 pattern(s))	0.837033
2	100.00 % (420 pattern(s))	0.00 % (0 pattern(s))	0.084951
3	99.76 % (419 pattern(s))	0.24 % (1 pattern(s))	1.231.246
4	100.00 % (420 pattern(s))	0.00 % (0 pattern(s))	0.137863
5	100.00 % (420 pattern(s))	0.00 % (0 pattern(s))	0.185359
6	100.00 % (420 pattern(s))	0.00 % (0 pattern(s))	0.009651
7	100.00 % (420 pattern(s))	0.00 % (0 pattern(s))	0.017347
8	100.00 % (420 pattern(s))	0.00 % (0 pattern(s))	0.035172
9	100.00 % (420 pattern(s))	0.00 % (0 pattern(s))	0.507792
10	99.76 % (419 pattern(s))	0.24 % (1 pattern(s))	1.302.725

Tabela 1: Resultados da etapa de treinamento/aprendizado da Rede Neural

Base dados de teste/validação - 180 padrões			
Execução	Percentual de aprendizado	Padrão não reconhecido	Erro
1	100.00 % (180 pattern(s))	0.00 % (0 pattern(s))	0.008386
2	100.00 % (180 pattern(s))	0.00 % (0 pattern(s))	0.004745
3	100.00 % (180 pattern(s))	0.00 % (0 pattern(s))	0.004978
4	100.00 % (180 pattern(s))	0.00 % (0 pattern(s))	0.005603
5	100.00 % (180 pattern(s))	0.00 % (0 pattern(s))	0.005603
6	100.00 % (180 pattern(s))	0.00 % (0 pattern(s))	0.004536
7	100.00 % (180 pattern(s))	0.00 % (0 pattern(s))	0.004528
8	100.00 % (180 pattern(s))	0.00 % (0 pattern(s))	0.019297
9	100.00 % (180 pattern(s))	0.00 % (0 pattern(s))	0.005882
10	100.00 % (180 pattern(s))	0.00 % (0 pattern(s))	0.005242

Tabela 2: Resultados da etapa de validação/testes da Rede Neural

Internamente, os AL's são representados através de matrizes cujas células possuem tamanhos variados dependendo da densidade do AL, em limiares mínimos e máximos a serem respeitados. A figura 7 apresenta um mesmo AL com 3 densidades diferentes.

Os AL's são especificados a partir de um arquivo XML, onde também são criados os agentes virtuais. Para cada agente, ou grupo de agentes, são gerados estímulos que ativam formações geométricas através dos ALs. Quando os agentes aplicam o comportamento de um AL, suas movimentações passam a ser controladas de maneira global ao AL (os AL's podem ser animados durante a simulação, através de simples translação do polígono), recebendo uma orientação, posição no AL, ação, velocidade, tempo de permanência e animação (ilustrada na Figura 8).

É possível a transição entre 2 formações (AL's), conforme ilustrado nas figuras 9. Este processo é bastante simples, tratando-se de uma simples interpolação linear entre 2 posições, sendo estas, as atribuídas em cada AL.

Na aplicação deste artigo, os AL's são acionados através de comandos detectados com posturas de mão. A próxima seção apresenta o protótipo e alguns resultados obtidos.

5. ARQUITETURA DO SISTEMA

Para a geração do protótipo, desenvolveu-se um sistema cliente-servidor comunicando-se via sockets. O Servidor, rodando em plataforma Windows, roda o programa MatLab e captura características da mão para geração de dados para o módulo de Redes Neurais. Após a detecção da postura, o seu código é enviado para o cliente (em plataforma Linux) que roda o simulador de formações geométricas de humanos virtuais baseados nos conceitos de AL's.

Para execução deste projeto foi utilizado um computador PC com



Figura 8: Agentes animados caminhando em um AL.

512 Mb de memória Ram, processador de 1,8 Ghz, uma Webcam Genius modelo Slim 321C padrão de mercado (facilmente encontrada nas lojas do ramo) e um pedestal ou tripé para servir de suporte à câmera. O setup do ambiente é ilustrado na Figura 10.

Para aquisição e processamento das imagens foi utilizado o Software MATLAB da Mathworks, que permite uma rápida implementação dos algoritmos propostos. No que diz respeito à RNA, foram utilizados os pacotes JavaNNS v1.1 e o SNNS v4.2 em suas versões para Windows. Além disso, utilizou-se o Dev C++ v4.9.9.2 na customização da rede neural exportada para C++ a partir do SNNS2C e na implementação da comunicação via Sockets entre a plataforma



Figura 7: Diferentes densidades de um mesmo AL.



Figura 9: Algumas possíveis formações.



Figura 10: Ambiente utilizado neste trabalho: 1. Webcam Genius modelo Slim 321C; 2. Tripé para máquina fotográfica/filmadora; 3. Pc com 512 Mb processador

Windows (que extrai a classificação das posturas) e o sistema operacional Linux (onde roda a aplicação que controla os humanos virtuais).

O sistema foi apresentado e validado junto a usuários leigos em termos de aplicações computacionais e mostrou-se bastante amigável e robusto.

6. CONSIDERAÇÕES FINAIS

No decorrer dos experimentos foi constatado que a técnica apresentada é robusta o bastante a ponto de ser utilizada como interface

em aplicações que demandam movimentação estruturada de humanos virtuais, em tempo real. O processamento inicial das imagens serviu a contento, uma vez que as metas desta etapa são extrair características das posturas para comparação, tendo sempre que o reconhecimento de formas e o pré-processamento destas imagens iniciais pode ser visto ainda como um desafio maior.

No que tange a RNA, os resultados foram surpreendentes. Pode-se afirmar que, é possível realizar satisfatoriamente a classificação de posturas de mãos utilizando Redes Neurais Artificiais. A técnica demonstrada também se mostrou extremamente barata uma vez que os dispositivos utilizados podem ser encontrados em qualquer loja, a preço módico. No que diz respeito ao custo computacional, é exigido o mínimo de recursos de máquina podendo ser colocada em execução em qualquer PC doméstico. Finalmente, no que concerne a aplicação, pode-se verificar que trata de interface amigável e prática, além de barata, e poderia servir de grande utilidade em diversos sistemas de simulação.

Como trabalhos futuros, podemos citar melhorias no processamento das imagens capturadas, podendo ser utilizadas técnicas de remoção ou subtração de background, entre outras. Assim, outras aplicações com humanos virtuais poderão fazer uso desta interface.

7. AGRADECIMENTOS

Este trabalho foi realizado em colaboração com HP Brazil e parcialmente financiado pela FINEP, Projeto Rede Brasileira de Visualização.

8. REFERÊNCIAS

- [1] H. Chen, H. Atabakhsh, C. Tseng, B. Marshall, S. Kaza, S. Eggers, H. Gowda, A. Shah, T. Petersen, and C. Violette. Visualization in law enforcement. In *Proceedings of the 2005 national conference on Digital government research*, Digital Government Research Center, pages 229–230, 2005.

- [2] T. Dunworth. *Information Technology and the Criminal Justice System: A Historical Review*. Sage Publications, Thousand Oaks, CA, 2005.
- [3] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice-Hall, New Jersey, EUA, 2nd ed edition, 1999.
- [4] JavaNNS: Java Neural Network Simulator. University of Tübingen.
- [5] J. Mackie and B. McCane. Finger detection with decision trees. In *proceedings of Image and Vision Computing*, pages 399–403, New Zealand, 2004.
- [6] T. Mitchell. *Machine Learning*. WCB / McGrall-Hill - Computer Science Series, Boston, MA, 1997.
- [7] S. R. Musse, F. S. Osório, F. Garat, M. Gomez, and D. Thalmann. Interaction with virtual human crowds using artificial neural networks to recognize hands postures. In *Proceedings of III Workshop on Virtual Reality*, pages 107–118, Gramado, RS, Brazil, 2000.
- [8] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–69, 1979.
- [9] J. R. Quinlan. Bagging, boosting and c4.5. In *In Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 725–730, Menlo Park, CA, 1996.
- [10] S. O. Rezende. *Sistemas inteligentes: fundamentos e aplicações*. Ed. Manole, São Paulo, 2003.
- [11] M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *Proc. of the IEEE Intl. Conf. on Neural Networks*, pages 586–591, San Francisco, CA, 1993.
- [12] D. Rumelhart, G. Hinton, and R. Williams. *Learning Internal Representations by Error Propagation - In: Parallel Distributed Processing*. Cambridge: MIT Press, vol.1 edition, 1986.
- [13] D. J. Sturman and D. Zeltzer. A survey of glove-based input. *IEEE Computer Graphics and Applications*, 14:30–39, January 1994.
- [14] H. K. Yoichi Sato, Makiko Saito. Real-time input of 3d pose and gestures of a user's hand and its applications for hci. *IEEE Virtual Reality Conference 2001 (VR'01)*, page 79, March 2001.
- [15] T. G. Zimmermann, J. Lanier, C. Blanchard, S. Bryson, and Y. Harvill. A hand gesture interface device. In *In Proceedings ACM Conf. Human Factors in Computing Systems and Graphics Interface*, pages 189–192, 1987.