Controle Interativo de Avatares pelo Movimento Humano

Elisson M.F. Meirelles Araújo, Luis A. Rivera, Auberto S. Castro, Fermín A. Tang Montané

Elissonmichael@yahoo.com.br, {rivera, ascv, tang}@uenf.br

Laboratório de Ciências Matemáticas – LCMAT
Universidade Estadual do Norte Fluminense – UENF
Av. Alberto Lamego, 2000, CEP 28015-620
Campos dos Goytacazes, Rio de Janeiro - Brasil

Resumo: *O uso de gestos na interação homem-computador apresenta uma justificativa muito poderosa nas pesquisas de modelagem, análise e reconhecimento de gestos. Nos últimos anos muitos novos meios de interação com o computador foram criados, entre eles, aqueles envolvendo visão computacional, reconhecimento de padrões e inteligência artificial. Neste trabalho desenvolve-se um modelo baseado em primitivas de movimentos controlados por gestos, para movimentar um personagem virtual, conhecido como avatar, em um ambiente virtual através de movimentos de um ator humano em frente de uma câmera*.

Palavras-chave: Reconhecimento de postura, gestos, interação, controle de movimento, avatar.

Abstract: *The use of gestures in human-computer interaction have been widely accepted as a powerful tool in modeling, analysis and gesture recognition research. In recent years, several new ways of interaction with computers were created, including those involving computer vision, pattern recognition and artificial intelligence. The present work develops a model based on motion controlled by primitive movement gestures, used by a human actor in front of a camera to move a virtual character, also known as avatar, inside a virtual environment*.

Keywords: Recognition posture, gestures, interaction, motion control, avatar.

1 Introdução

Recentemente, com o desenvolvimento de computadores mais potentes e com o crescimento do interesse das necessidades humanas por meios de interação natural com os computadores, vêm se desenvolvendo modelos de comunicação homem-computador baseado em técnicas de visão computacional e reconhecimento de padrões. As ferramentas fornecidas por essas disciplinas permitem capturar e analisar sequencias continuadas de informações (imagens, posições no ambiente, sons, voz, etc.), que após extrair as características relevantes, no contexto do processo, são usadas com controle de interação com o sistema. Assim, por exemplo, em sistemas com principio de imersão um ator humano, neste caso do mundo real, pode interagir com elementos do ambiente virtual ou atores virtuais. No contexto deste trabalho um ator virtual é conhecido como um *avatar*, que se movimenta no ambiente virtual segundo movimentos do ator humano.

Neste caso, o interesse é a análise de sequencia de imagens capturadas por uma câmera, tal como a webcam de uma laptop ou uma webcam tradicional, para operar em qualquer ambiente e lugar, sem precisar de outros dispositivos adicionais intrusivos. Existem trabalhos como [01] [02] [03] [04] que usam categorias de dispositivos sofisticados de captura para reconhecimento de gestos. Por exemplo, em [01] é usada uma câmera com sensor de movimento DVS-câmeras (*dynamic vision sensor cameras*). Em [02] [03] são usados câmeras com sensor Kinect com tendências eficientes para reconhecimento de gestos aplicados a jogos e seguranças. Porém, essas câmeras não são de usos comuns como as simples webcams. Os dispositivos intrusivos, como marcas de rastreamento, luvas, roupas especiais, etc são orientados para aplicações específicas não comuns. Mitra et al. [04] analisam métodos de reconhecimento de gestos com diversas abordagens, incluindo dispositivos intrusivos.

Qualquer desses dispositivos poderiam, em certa forma, ser usados para a animação de avatares, porém, ainda não estão ao alcance de todos ou são intrusivos. O que se busca é uma técnica de animar avatares evitando, no possível, dispositivos intrusivos, quando se trata, por exemplo, do uso das técnicas para jogos orientados para crianças, uso em terapias, etc.

No processo de controle de animação de avatares existem dificuldades tais como: a atribuição do conjunto de comportamentos do avatar de forma a conseguir um movimento coerente com o desejado, atribuir o controle do usuário para um movimento, e execução do movimento em tempo real. Os dispositivos mouse e joystick tipicamente indicam posições, velocidade ou ação de comportamento. Um ator humano geralmente gostaria passar os movimentos em tempo real ao avatar através de seus próprios movimentos, como ele mesmo estivesse personificando o avatar, o que é possível com gestos de movimentos capturados por uma webcam.

A proposta deste trabalho tem como principal objetivo o estabelecimento de um modelo para controle de movimentos de um avatar num ambiente virtual baseado no reconhecimento de gestos de um ator humano utilizando uma câmera. Em outras palavras, uma ferramenta que possa ser utilizada em aplicativos fisicamente interativos baseados em princípios de imersão, ou seja, ter a capacidade de fazer com que o usuário sinta-se "dentro" do ambiente virtual mencionado.

O trabalho se organiza da seguinte forma: na Seção 2 abordam-se reconhecimentos de gestos e trabalhos relacionados. Na Seção 3 se formula o modelo de reconhecimento de gestos para a interação no ambiente

virtual; na Seção 4 analisa-se o modelo proposto do ponto de vista da porcentagem de acertos, e finalmente na Seção 5 conclui-se com a indicação de trabalhos futuros.

2 Reconhecimento de gestos (posturas)

Uma *postura* é uma representação estática de uma ação do corpo humano ou parte dele. Uma pessoa pode passar uma mensagem para um observador através de um movimento chamado de gesto. Então, computacionalmente falando, um *gesto* é uma sequência finita de posturas, neste caso imagens, onde o observador é o computador capturando um conjunto de imagens de um movimento.

Por exemplo, o gesto *caminhar para a direita* é uma sequência finita de posturas que ilustram a evolução de dar um passo para a direita. Uma sequência de passos coerentes permitirá ao corpo se deslocar para direita caminhando. Nessa perspectiva, uma sequência de posturas que gera um passo simples para a direita será uma primitiva de *caminhar-direita*. A análise do conjunto da sequência das posturas que define esse único passo determinará a ação da primitiva *caminhar-direita*. Por tanto, um individuo caminhando ou dando *n* passos para a direita será equivalente à ação da primitiva *caminhar-direita n* vezes. Da mesma forma são definidos outros gestos do comportamento básico de um indivíduo.

Existem três problemas no esquema de detecção de gestos a partir de uma sequência continuada de movimentos: primeiro, a segmentação do movimento continuado em uma sequência de gestos; segundo, o estabelecimento do número de posturas relevantes que definem um gesto; e, terceiro, a eleição de um gesto apropriado para uma postura que pode pertencer a outros gestos. Existem várias abordagens para resolver esses problemas, tratados amplamente em [04]. Uma alternativa bem sucedida é a abordagem baseada em Modelo Oculto de Markov (HMM – *Hidden Markov Model*), usados com sucesso na identificação de gestos gerados com diferentes partes do corpo [05][06][07]. Um dos trabalhos pioneiros em reconhecimento de sequência de posturas e gestos foi desenvolvido por Yamato et al. [08] para reconhecer seis gestos de movimentos de jogo de tênis usando HMM. Chen et al. [09] usam HMM, com descritores de Fourier combinados com movimentos, para reconhecer vinte gestos de mão a partir de movimento contínuo. Nianjun et al. [10] reconhecem 26 letras através de gestos da mão, extraindo atributos da geometria e trajetória dos movimentos, usando HMMs.

Outros métodos de reconhecimento também usados eficientemente como Truyenque [11] que propôs um modelo que utiliza gestos da mão para interagir com slides Powerpoint, controlar um jogo simples e desenhar no Paint Brush, substituindo o mouse, e utilizando máquinas de estados finitas para representação de inferência de gestos baseando-se nas quantidades de dedos detectados na silhueta. Para detectar um dedo constrói-se uma linha utilizando o método de mínimos quadrados. Scandaroli e Melo [12] utilizaram uma câmera USB genérica para detectar os olhos do usuário, após a detecção do rosto, e então transmitir a posição do usuário em relação à câmera, movimentando uma câmera no ambiente virtual de acordo com a posição dos olhos do usuário. Desenvolveram um ambiente virtual imersivo, onde a maneira como o usuário visualizava o ambiente virtual dependia da posição de onde ele olhava para o monitor. Para a detecção do rosto e dos olhos foi utilizado o método de Viola e Jones [13], baseado em características do tipo Haar utilizando AdaBoost e filtro de Kalman para fazer a correção e predição da posição da face na imagem.

Chen et al. [14] criaram uma ferramenta para a detecção de quatro posturas de mão, a posição com dois dedos, a palma da mão, o pulso e a posição com o dedo mínimo em destaque usando características Haar-like baseado no algoritmo desenvolvido por Viola e Jones [13]. Bretzner et al. [15] criaram um sistema para reconhecimento de gestos da mão, onde os gestos são representados em termos de características de hierarquias de imagens em cores em multi-escala, posição, orientação. A mão é representada por um modelo que consiste na representação da palma da mão e dos cinco dedos. O reconhecimento do gesto é realizado através de métodos estatísticos.

Lee et al. [16] desenvolveram um aplicativo Virtual Office Environment System (VOES) que permite controlar o movimento de um avatar com gestos da mão. O avatar é usado para navegar e interagir com os outros participantes. O reconhecimento de gestos contínuos da mão é modelado através de um autômato finito, de forma a controlar de forma intuitiva movimento do avatar. Os movimentos básicos são organizados como primitivas num banco de dados de movimentos do VOES.

Lee et al. [17] estabeleceram uma forma de animar um avatar no espaço 3D a partir de captura de corpo humano em movimentos. Os movimentos básicos, gerados previamente como scripts (subprograma) de movimentos, são armazenados em forma de árvore. Os nós dá árvore são grupos de sequência de movimentos possíveis, de forma a serem ativadas constantemente a partir da interpretação do movimento do corpo humano. Eles usaram marcas retro-refletivas aderidas ao corpo humano para a simplificação na determinação dos atributos das posturas, e HMM para a representação de baixo nível. Ni et al. [18] avaliaram um framework considerando aspetos temporal e espacial para controlar um avatar em movimentos em jogos 2D a partir das mímicas de movimento humano.

3 Modelo de controle de avatar

Um avatar é a representação de um ator humano real em um ambiente virtual. Ele deve realizar movimentos controlados pelo humano através de gestos [19]. O ator humano se movimenta, realizando gestos, em frente de uma câmera a fim de que o avatar realize os mesmos movimentos ou similares.

Detecção de movimentos complexos geralmente demanda o uso de mais de uma câmera, possivelmente, de detectores de profundidade e adesão de marcas anatómicas no corpo. No caso de movimentos simples, como controle de avatares em jogos para crianças, apenas uma câmera simples é necessária.

Uma sequência de imagens capturadas pela câmera, a cada instante de tempo, permitirá identificar o tipo de movimento que o ator humano está realizando. Uma vez identificado o gesto, dependendo do contexto do movimento, se procederia a realizar a sequência de movimentos do avatar, para produzir o que manda o gesto; por exemplo, dar um passo de caminhar. Produzir um passo de caminhar é produzir uma sequência de pequenas variações do corpo. Isto implica se esse passo é com pé direito ou esquerdo, movimentos braços, etc. Produzir esses movimentos dinamicamente recarga de trabalho ao processador. Mais conveniente, para esses casos, é gerar um banco de primitivas de movimentos na etapa de modelagem. As primitivas de movimentos são scripts (subprograma) de gestos, onde cada gesto é uma ativação de uma sequência de frames, que processada passa a ser uma sequência de posturas, constituindo um gesto básico.

Cada sequência de imagens, que define um possível movimento aceito, será analisada em relação aos movimentos representados, em um processo de treinamento prévio, em um sistema de redes definidos, neste caso, por HMMs. Uma vez que o movimento for aceito através reconhecimento de gesto, o motor de controle de movimentos de avatar ativa a primitiva correspondente ao gesto reconhecido a partir de um banco de primitivas para serem exibidas como a movimentação do avatar via a interface gráfica. O processo descrito está ilustrado na Figura 1.

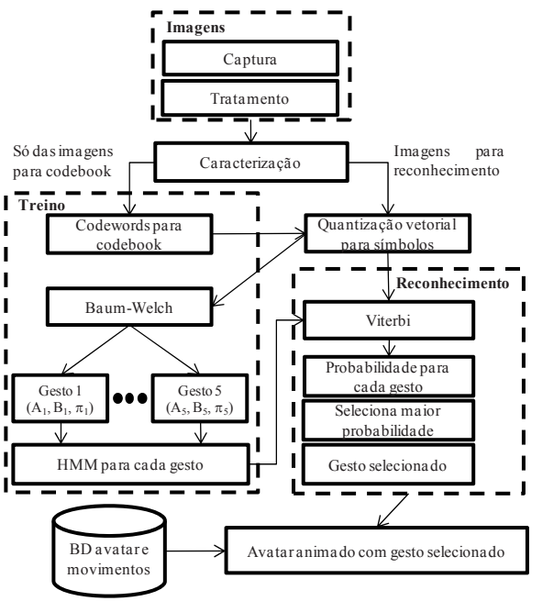

Figura 1: Arquitetura de controle de movimento de um avatar através de gestos.

3.1 Primitivas de movimento

Na modelagem do sistema considerou-se um módulo de geração de primitivas a partir de grupos de subconjuntos de posturas de um avatar em 3D. Para esse objetivo, utiliza-se um protótipo de corpo humano, conhecido como armature, com as articulações e as partes respectivas estabelecidas, tal como ilustrada pela Figura 2. Com a movimentação da armature, segundo as especificações das rotações, translações das partes do modelo e local dos membros, são geradas sequências de 30 frames por segundo. Esses frames são salvos no banco de scripts. Por exemplo, as três imagens da Figura 2 mostram os frames 1, 15 e 30, respectivamente, de um gesto "abrir os braços estando parado".

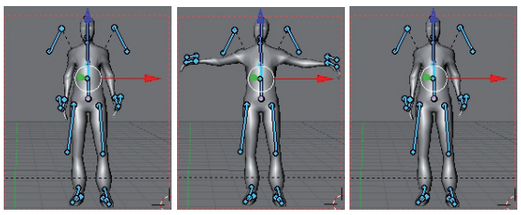

Figura 2: Gerando animações de armatures com Blender.

Para o propósito deste trabalho foram consideradas 6 gestos: *default relax, caminhar para esquerda, caminhar para direita, levantar braço esquerdo, levantar braço direito*, e *levantar os dois braços*. Assim, cada gesto básico é um movimento básico, composto pela respectiva subsequência de frames, definido como primitiva. A Figura 3 ilustra as 6 posturas compondo as primitivas. Neste caso, cada primitiva é definida por quatro posturas chaves.

Posturas chaves são as posturas representativas selecionadas de uma sequência de maior de frames que define um gesto. Assim, existiram posturas próximas (similares) às posturas chaves que não aparecem na movimentação do avatar.

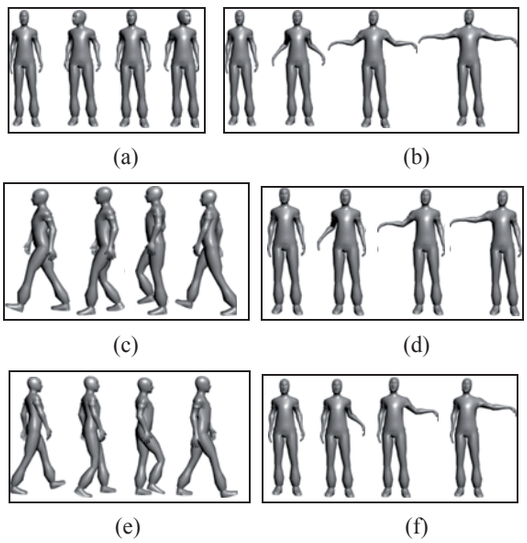

Figura 3: Gestos com quatro posturas: (a) *default-relax*; (b) *ambos-braços*; (c) *caminha-esquerda*; (d) *braço-esquerdo*; (e) *caminha-direita* e (f) *braço-direito*.

3.2 Captura e segmentação

No processo de captura e segmentação foi utilizada a biblioteca OpenCV para obtenção de imagens a partir de uma webcam e para processamento dessas imagens com o objetivo de obter imagens binárias, neste caso com o fundo em preto e o corpo do humano detectado em

branco. Os frames capturados possuem altura A de 640 pixels e largura L de 480 pixels.

No início da captura, é registrada uma imagem de fundo do ambiente. Ela é utilizada durante o processo de extração do fundo. Todos os frames seguintes sofrem uma operação de subtração em relação à imagem de fundo. A imagem resultante passa por um processo de binarização com uma certa tolerância. A Figura 4 ilustra a imagem de fundo, um frame capturado e equivalente binarizada.

Figura 4: Segmentação da imagem (fundo, imagem, binarizada).

É importante realizar alguns pré-processamentos e pós-processamentos nas imagens com o intuito de melhorar a silhueta detectada e remover ruídos e informações desnecessárias. Esses processamentos são operações morfológicas, suavização e eliminação de falsas detecções.

3.3 Caracterização

Esse processo consiste em analisar a imagem pré-processada e buscar pelos padrões de interesse. Nesta fase, a partir de cada imagem binarizada, é obtido um vetor de valores numéricos, conhecido como vetor característico, que representam eficientemente a imagem. O vetor V da Figura 5 é o vetor característico.

A seleção de boas características para o reconhecimento de posturas é uma fase crucial no processo e pode determinar o sucesso ou falha do algoritmo em uso. Esse conjunto de características deve idealmente descrever a postura e/ou gesto de forma única, no sentido que cada diferente posição do corpo deve prover um conjunto de boas e diferentes características para um reconhecimento confiável.

Na literatura existem vários métodos de caracterização de gestos, dependendo da complexidade dos gestos e o modelo de aplicação, como o modelo de Elmezain et al. [05] que utiliza a localização, orientação e velocidade da trajetória de um membro do corpo, neste caso da mão. Os momentos invariantes de Hu [20], também apropriados para caracterizar partes do corpo, extraem informações das imagens eficientemente apenas para um conjunto de sete parâmetros insensíveis às deformações rígidas, como translação, rotação, escala e espelhamento. Yamato, Ohya e Ishii [08] usam segmentação em malha, onde cada cela da malha fornece a frequência relativa da ocorrência de parte do membro do corpo em possível postura. O método de segmentação em malhas é mais apropriado para detecção de movimento do corpo todo, enquanto os outros métodos são mais apropriados para gestos com análise em certo detalhe de um membro do corpo.

Para formação do vetor de características V usou-se características por segmentação em malha, por sua capacidade de descrever complexos padrões em duas dimensões com sucesso, cada imagem de $L \times A$ pixels contendo a região de interesse é dividida em segmentos de malha de $L_s \times A_s$ pixels. Se o número de pixels brancos no segmento (i, j) da malha for $B(i, j)$, então a proporção de pixels brancos nesse segmento é calculada como

$$V(i \times A_s + j) = \frac{B(i, j)}{L_s \times A_s},$$

Ocupando esse valor na posição $(i \times A_s + j)$ do vetor V.

A malha utilizada no começo do sistema tinha tamanho 3×3, gerando um vetor de características de tamanho 9. Esse tamanho não gerou informações suficientes para diferenciar corretamente algumas posturas como, por exemplo, estar virado para direita ou para esquerda, gerando bastante confusão quando se estava nessas posições, aumentar a malha resolveu esse problema. Apesar de Yamato, Ohya e Ishii [08] terem dividido os segmentos da malha em partes de 8×8 pixels, gerando um vetor de características de dimensão 625, esses valores pareceram excessos desnecessários, ter dobrado a altura e largura da malha no sistema foi suficiente para os problemas de confusão nas posturas. A Figura 5 exibe um exemplo de malha de características de tamanho 6x6, utilizada no sistema final, ela gera um vetor de 36 elementos. Cada retângulo da malha gera um valor, que finalmente é colocado em um vetor V. Observa-se que a célula preta gera um valor 0.0 ou próximo a zero e célula contendo parte do corpo gera um valor próximo a 1.0.

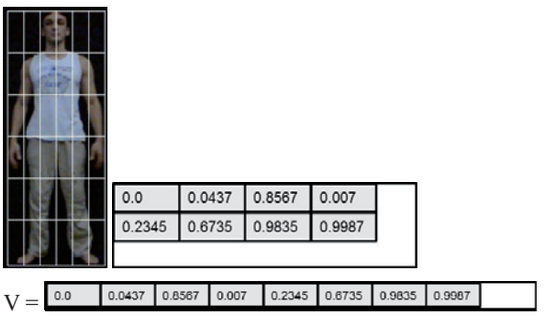

Figura 5: Características da imagem em vetor V.

Os vetores de características não operam diretamente com os HMMs, pois eles ainda contêm muitos elementos, neste caso 36, em ponto flutuante. No processo de treino de HMM deve-se fornecer característica da imagem (observação "obs") e o que a imagem representa como resultado (emitido "emit"); na classificação também se fornece ao HMM a sequência de observação para receber como resultado os emitidos. Então, é necessário encontrar um identificador de vetores de característica conhecido como *codebook*.

A cada vetor característico das posturas chaves, definidas na seção anterior, associa-se um código em número inteiro. Das 24 posturas que compõem os 6 gestos (Figura 3), as primeiras quatro posturas são do gesto *defaut-relax*, ou seja uma ação respeito a gestos não identificados obtêm *code* 0, resto das posturas adquirem *codes* de 1 a 20, como segue: *code* 1, 2, 3 e 4 para a sequência de posturas do gesto *caminhar-direita*, *code* 5, 6, 7 e 8 para

posturas do gesto *caminha-esquerda*, *code* 9, 10, 11, e 12 para posturas do gesto *braço-direito*, *code* 13, 14,15 e 16 para posturas do gesto *braço-esquerdo*, e finalmente, *code* 17, 18, 19 e 20 para posturas do gesto *ambos-braços*. Na realidade, são cinco gestos que devem ser identificados.

Para o processo de treinamento são geradas sequências de frames dos cinco gestos considerados neste trabalho. Cada gesto é realizado várias vezes, com estilos diferentes e por pessoas diferentes, coletando-se, desse modo, um conjunto grande de frames. Desse conjunto de frames são selecionadas quatro posturas chaves (ou *frame-keys*) mais representativos para cada um dos cinco gestos. Podia ter sido considerado mais *frame-keys*, mas por razões computacionais só se considerou quatro, porque para geração de movimentos básicos basta um mínimo de posturas diferentes de forma que exibidos, em forma de animação, mostre a realização do gesto.

Dado um conjunto de todos os frames, isto é, seus respectivos vetores característicos de um gesto, deve-se associar cada um dos frames ao qual dos quatro frames-chave se aproxima. Pode-se dizer que, o conjunto foi clusterizado em quatro subconjuntos disjuntos conhecidos como *clusters*, onde cada subconjunto é representado pelo seu frame-chave. Assim, todos os frames do conjunto do gesto recebem um *code* que corresponde ao *frame-keys* de seus respectivos subconjuntos.

Como agrupar todos os frames em base seus vetores característicos em subconjuntos? Esse processo se realiza pelo método de clusterização, nesta caso usa-se o método K-means [21] semi-manual, com k = 4 e os quatro meios iniciais são os quatro *frame-keys* escolhidos. O processo de k-means, durante as iterações, melhora os *frame-keys* mais representativos, de forma que a distância entre *frame-keys* e os elementos do cluster seja mínima.

3.4 Modelagem de HMM para gestos

O Modelo de Markov Oculto, segundo Rabiner [RABINER], é definido como um grafo dirigido de N estados e $N \times N$ arcos $\{a_{ij}\}$ que definem a probabilidade de transição do estado i para outro estado j, com a condição da soma de todas as probabilidades de transição saindo do estado i para os outros estados seja 1. A característica de um HMM é que qualquer estado i tem a probabilidade de emissão $b_i(k)$ do símbolo v_k, para $k = 1,..., M$, como resultado. Dessa forma o estado emissor é oculto, porque o estado emissor pode ser qualquer em um instante t.

No processo de treinamento, um grafo genérico de topologia desejada, calibra eficientemente os valores das probabilidades de transição a_{ij} e as probabilidades de emissão b_k, a partir das sequências válidas de observações $\{o_i\}_{t=1,...,T}$ e seus respectivos possíveis símbolos de emissões, e $\{v_k\}_{k=1,...,M}$, são fornecidas. Para exigir que o processo inicie do estado 1, é necessário estabelecer $\{\pi_i\}_{i=1,...,N}$, tal que $\pi_i = P[1 = i]$.

Formalmente, um HMM é definido como $\lambda = (A, B, \pi)$, onde $A = \{a_{ij}\}_{N \times N}$, $B = \{b_i(k)\}_{N \times M}$ são parâmetros do modelo e $\{\pi_i\}_{i=1,...,N}$ é a probabilidade do estado inicial. Para uma sequência de símbolo de observação $O = \{o_i\}_{t=1,...,T}$ .e um conjunto de símbolos a serem emitidos $V = \{v_k\}_{k=1,...,M}$.

Para estabelecer a topologia geral de HMM, tal como recomendam Liu et al. [23], Elmezain et al. [05] e Tataru et al. [24], usou-se a topologia LRB (left Right Banded), como modelo na HMM que apresenta a melhor taxa de reconhecimento para esta categoria de gestos. Então são gerados quatro HMMs de topologia LRB, um para cada gesto, considerando que cada modelo é composto de quatro estados e cinco emissões (v0 caso relax, de v1 a v4 para posturas válidas), com arcos indo de estado 1 a 4, como recursão no mesmo estado. Os quatro estados com possibilidade de emitir os quatro símbolos associados às primitivas do respectivo gesto. A Figura 6 ilustra o modelo genérico de HMM, já com os respetivos treinos, os cinco HMMs adquirem λ_1,λ_2, λ_3, λ_4 e λ_5.

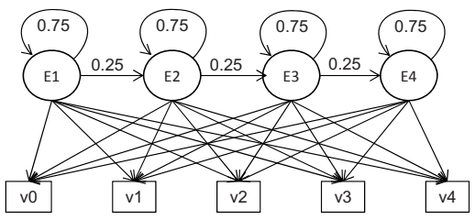

Figura 6: Topologia LRB inicial de um HMM genérico antes do treino.

No treino, os parâmetros de λ=(A, B, π) são calculados usando-se os símbolos observados O de forma que $P[O \mid \lambda]$ seja maximizado. O treinamento de um modelo estatístico não é, em geral, um problema simples e a existência de um algoritmo eficiente para esse problema é condição fundamental para a aplicabilidade desse modelo estatístico. Esse o caso de HMM, pois existe o eficiente algoritmo de Baum-Welch [25], que é caso particular de EM (Expectation-Maximization), que é usado neste trabalho.

3.5 Reconhecimento de gestos

Dados os modelos HMMs λ_i, para $i = 1,...,4$, devidamente treinados, isto é, os parâmetros A_i e B_i respectivos computados pelo método Baum-Welch, a seguinte etapa é o reconhecimentos em tempo real de toda sequência de posturas de um possível gesto. Para isto, primeiramente deve se identificar qual dos modelos λ_i se ajusta mais ao possível gesto, isto é melhor ajuste de $P[\lambda_i \mid O]$. Depois, por último reconhecer o gesto com o modelo λ identificado.

Nesta etapa, as posturas são capturadas em ordem de 30 quadros por segundo, em forma continuada, através de uma webcam. Cada frame é segmentado, pré-processada, caracterizada e associada apropriadamente a um dos clusters para adquirir seu respectivo *codebook*. Cada uma de essas posturas, colocadas em sequências continuadas,

passam a ser os símbolos observados $O = O_1 O_2 ,... O_T$ que são alimentadas os modelo λ para calcular $P[O|\lambda]$, como a probabilidade de ocorrência da observação O no modelo λ. Esse problema é resolvido com alto grau de aceitação pelo algoritmo de Viterbi [25].

Uma vez reconhecido o gesto são ativados os script de movimentos do avatar e exibidos em um ambiente virtual, como imagem de fundo, tal como se observa na sequência de situações da Figura 7.

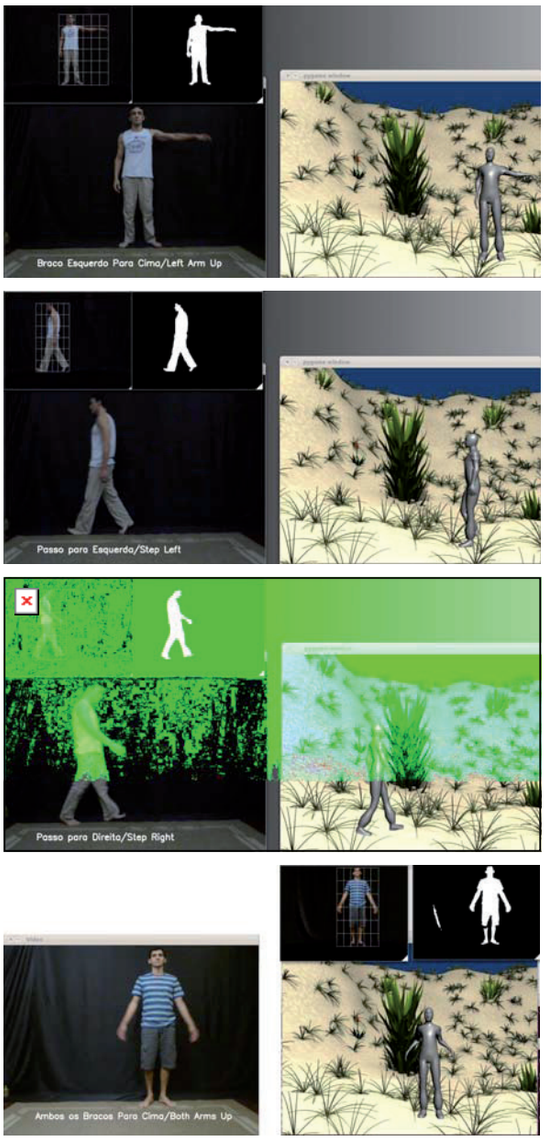

Figura 7: Sequência de controle de movimentos de um avatar através de gestos de corpo.

4 Análise do modelo

Em condições normais de ambiente e luz, o modelo se comporta com uma porcentagem alta de acertos para sequências de gestos definidos. Isto é, quando a captura de vídeo inicia com o ator humano realizando um dos movimentos do gesto definido. Porém, se observa tem segmentos de movimento, tipo transições entre dois gestos diferentes, que o modelo não se comportou de forma esperada. Por exemplo, quando um gesto de andar para direita ou para esquerda acontecia seguido da postura parado de frente para a câmera. Isso aconteceu principalmente porque os vídeos que treinaram o sistema não continham um exemplo desse gesto, em outras palavras, as HMMs treinadas para reconhecerem os gestos de passos, nunca começavam com o símbolo 0, que é a *codeword* para a postura parado de frente, então o gesto de caminhar só era reconhecido se o ator humano já estivesse de lado no início das sequências desses gestos. Essa situação foi resolvida, injetando alguns símbolos 0 nas sequências de treinos dessas HMMs.

Existem também, casos que o reconhecimento de um gesto foi iniciado quando com sequência parcial de observações, obrigando a considerar as probabilidades de transição para estados anteriores, quebrando um pouco a topologia LRB. Com essas variações complementares o modelo responde razoavelmente para o propósito de movimentação do avatar obedecendo os gestos estabelecido neste trabalho.

5 Conclusões e trabalhos futuros

Neste trabalho foi implementado um avatar em um ambiente virtual que recebe comandos do ator humano através de visão computacional e técnicas de reconhecimento de padrões. O modelo baseou-se em algoritmos de segmentação no processo de captura e extração da silhueta do ator humano a partir de uma câmera. É importante ter um ambiente onde exista luz controlada, evitando-se assim variações na iluminação e muitos ruídos no sinal da silhueta extraída.

O ambiente virtual é composto de um avatar que fica aguardando comandos de entrada que decidirão qual movimento será executado, esses gestos são guardados em uma fila no momento em que são detectados e saem da fila a toda vez que o avatar conclui uma animação.

Como trabalhos futuros sugere-se a extensão do modelo para reconhecimento de um conjunto maior, e completo, de movimentos naturais que realiza um humano. Melhorar a caracterização de forma a capturar detalhes redundantes com a combinação com outros métodos, incorporando variações de movimentos em ambiente real tri-dimensional, possivelmente incorporando os acessórios Kinect em conjunto com um sistema de HMMs. Expandir o número de animações do avatar e criar outros objetos com os quais ele poderia interagir no cenário.

Referências bibliográficas

[01] Ahn, E.Y.; Lee, J. H.; Mullen, T.; Yen, J. Dynamic Sensor Camera Based Bare Hand Gesture Recognition. IEEE Proceedings of Symposium on Computational Intelligence for Multimedia, Signal and Vision Processing (CIMSIVP), 2011. Pags. 52-59.

[02] Gonçalves, N.; Rodrigues, J.; Costa, S. e Soares, F. Automatic detection of stereotypical motor movements. Procedia Engineering, 47 (2012), pag. 590-593.

[03] Mahbub, U.; Imtiaz, H.; Roy, T. Rahman, S., e Ahad, A. R. A template matching approach of one-shot-

learning gesture recognition. Pattern Recognition Letters, 2012, http://dx.doi.org/10.1016/j.patrec.2012.09.014

[04] Mitra, S. Acharya, T. Gesture recognition: A surveys. IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and reviews, v37, n3, 2007.

[05] Elmezain, M.; Al-Hamadi, A.; Appenrodt, J.; Michaelis, B. A hidden Markov model-based isolated and meaningfull hand gesture recognition. World academy of science, engineering and technology, 41, 2009, 393-400.

[06] Mohandes, M.; Deriche, M.; Johar, U.; e Ilyas, S. A signer-independent Arabic sign language recognition system using face detection, geometric features, and a hidden Markov models. Computer and Electrical Engineering, 38, (2012), 422-433.

[07] Al-Rousan, M.; Assaleh, K.; e Tala'a, A. Video-based signer-independent Arabic sign language recognition using hidden Markov models. Applied Soft Computing, 9, (2009), 990-999.

[08] YAMATO, J.; OHYA, J.; ISHII, K. Recognizing human action in time- sequential images using hidden markov model. Yokosuka, Japan, 1992.

[09] Chen, F.; Fu, Ch.; e Huang, Ch. Hand gesture recognition using a real-time tracking method and hidden Markov models. Image and Vision Computer, 21, (2003), 745-758.

[10] Nianjun, L.; Lovell, B. C.; Kootsookos, P.J. Evaluation of hmm training algorithms for letter hand gesture recognition. Proceedings of the IEEE International Symposium on Signal Processing and Information Technology, The Institute of Electrical and Electronics Engineers, v. 1, n. 1, 2003, pag. 4-7.

[11] Truyenque, M. A. Q. Uma Aplicação de Visão Computacional que Utiliza Gestos da Mão para Interagir com o Computador. Dissertação (Mestrado em Informática), Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2005.

[12] Scandaroli, T. G.; Melo, D. D. Detecção de usuário e estimação de posição para interface com realidade virtual. Brasilia/DF: [s.n.], 2009.

[13] Viola, P.; Jones, M. Robust real-time object detectin. Cambridge Research Laboratory Technical Report Series CRL2001/01, 2001, 1-24

[14] Chen, Q.; Geoganas, N.D.; Petriu, E.M. Real-time vision-based hand gesture recognition using haar-like features. Instrumentation and Measurement Technology Conference – IMTC, Polonia, (2007).

[15] Bretzner, L.; Laptev, I.; Lindeberg, T. Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. IEEE Proceedings of Fifth International Conference on Automatic Face and Recognition, 2002, 405-410.

[16] Lee, ChanSu; Ghyme, SangWon; Park, ChanJong; e Whojn, KwangYun (1998). The control of avatar motion using hand gesture, ACM: VSRT'98, Taipai – Taiwan, pag. 2-5.

[17] Lee, Jehee; Chai, Jinxiang; Reitsma, Paul; Hodgins, Jessica; e Pollard, Nancy (2002). Interactive control of avatars animated with human motion data. SIGGRAPH'02. Pags. 491-500.

[18] Li, Na; Noraveji, Neema; Kimura, Hiroaki; e Ofek, Eyal (2006). Improving the experience of controlling avatars in camera-based games using physical input. ACM: MM'06, oct. 23-27, Santa Barbara – California, USA. Pag. 73-76.

[19] Li, Na; Chen, Chun; Wang, Qiang; Song, Mingli; Tao, Mingli; e Li, Xuelong (2008). Avatar motion control by natural body movement via camera. Neurocomputing 72 (2008), pag. 648-652.

[20] [Hu62] Hu, M.K. Visual pattern recognition by moment invariants. IEEE Transactions on Information Theory, v8, n2, 1962, 179-187.

[21] Bishop, C. M. Pattern Recognition and Machine Learning. [S.l.]: Springer, 2006, Pag. 424-430.

[22] Liu, H.; Yu, X. Application research of k-means clustering algorithm in image retrieval system. Proceedings of the Second Symposium International Computer Science and Computational Technology, 2009, p. 274-277.

[23] Liu, N. et al. Understading hmm training for video gesture recognition. TENCON 2004. IEEE Region 10 Conference, Conference Publications, n. 1, 2004, pag. 567-570.

[24] Tataru, V.; Vieriu, R.-L.; Goras, L. On hand gestures recognition using hidden markov models. Acta Technica Napocensis Electronics and Telecommunications, v. 51, n. 3, 2010, pag. 29-32.

[25] Rabiner, L. R. A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE, v. 77, n. 2, 1989, pag. 257-286.