VRemote - Interface Humano-Computador utilizando reconhecimento de gestos com Hidden-Markov Models

Bernardo Cunha Vieira
Universidade Federal de Minas Gerais
Visão Computacional
Belo Horizonte, MG
bcvieira@dcc.ufmg.br

Resumo

Este trabalho implementa uma interface humano-computador baseada em reconhecimento de gestos. A interface visa controlar a passagem de transparências durante apresentações. Geralmente utiliza-se um controle remoto, com botões com funções próxima/anterior. A gramática deste trabalho é constituida de dois gestos simples de mesma função dos botões. O reconhecimento é dado de maneira contínua, em tempo real e cada gesto é identificado através de um modelo escondido de Markov.

1. Introdução

A utilização de gestos para controle de máquinas vêem sendo amplamente estudado e comentado. Artigos já de 1996, como [12], proviam meios de localização e rastreamento de pessoas para criação de softwares. Os softwares vão desde de softwares para criação de interface-humano computador até análise de padrões de comportamento. Uma aplicação do trabalho Pfinder foi [2], que também utiliza modelos escondidos de Markov.

Os modelos escondidos de Markov são a base matemática, provavelmente, mais utilizada para sistemas de reconhecimento de padrões da fala [4]. Um modelo deste tipo, assume que o processo seja um processo de Markov, com paramêtros desconhecidos. O desafio é determinar os parâmetros escondidos a partir dos parâmetros observáveis, veja a figura 1. Devido à habilidade de identificação de padrões através de treinamento da rede, esta ferramenta passou a ser utilizada em larga escala também em aplicações de visão computacional, como [10], que incluem qualquer análise de padrões, desde a identificação de objetos até identificação de uma conjunto de gestos. Recentemente tem-se utilizado o algoritmo Condensation descrito em [6], também para identificação de gestos no tempo [3]. Para saber mais sobre HMMs na identificação de gestos veja [10].

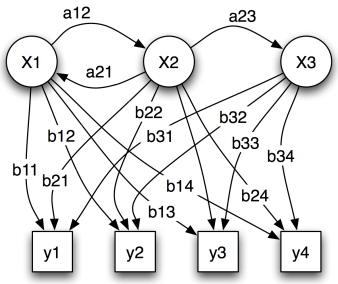

Figura 1. Modelo Escondido de Markov - Retirada da Wikipedia

$$x \rightarrow hidden\ states$$
$$y \leftarrow possible\ observations$$
$$a \leftarrow state\ transition\ probabilities$$
$$b \leftarrow output\ probabilities$$

2. Trabalhos Relacionados

Este trabalho é parecido com a tese de mestrado em [9]. Aproveita-se da tese a discretização das observações possíveis, mostrados na figura 2, e o número de estados da cadeia de Markov, dado pela fórmula a seguir.

$$N_E = \max\left(T_{min}, \frac{T_{avg}}{2}\right)$$

A parte do modelos escondidos de Markov é derivada de [7], com alguma alterações para rodar no windows (funções isnan, isinf).

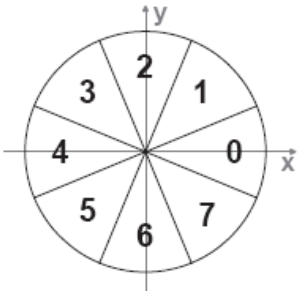

Figura 2. Discretização do espaço de observações

O código fonte final do projeto pode ser baixado em [11].

O artigo [12] mostra bem o problema de localização das pessoas e partes de seu corpo. O problema que os autores pretendem resolver é o rastreamento em tempo real do corpo humano. Eles pretendem desenvolver um sistema que em tempo real rastreie as pessoas e interprete seu comportamento. O sistema utiliza uma única câmera e funciona para uma única pessoa. Pretende-se achar onde se encontram o corpo e as partes do corpo humano, como mãos e cabeça. Este trabalho tem as mesmas restrições que o artigo. Ele diferentemente do artigo usa o domínio de cores (H,S,V), segmentação por distribuição probabilística da cor(componente H), identificação do centro com CamShift [1].

Este trabalho é mais próximo de [10], uma vez que baseia-se no mesmo fundamento matemático, os HMMs. Foi deste artigo que teve-se a inspiração para este trabalho. O artigo é mais profundo e utiliza outro método de pré-processamento e extração de features.

Para a análise de gestos manuais (como levantar/abaixar os dedos da mão) recomenda-se os trabalhos [8], [13]. As duas resenhas tem um mesmo core que é a identificação dos modelos a serem utilizados a partir dos problemas que deseja-se resolver (cardboard, wireframe), e o modelo de cinemática. Há também a identificação da mão (principalmente através de cores, como o espaço RGB normalizado), identificação dos valores dos parâmetros dos modelos para melhor descrever o que está acontecendo, e posterior identificação de gestos.

3. Metodologia

Tendo em mente o controle do progama de apresentações, define-se que 2 gestos são suficientes. Para facilitar a extração de features utiliza-se uma luva da cor amarela para a segmentação e o tracking do gesto. Utilizando a abordagem de [10], um sistema de reconhecimento de gestos pode ser divido em 3 partes:

- Pré-pocessamento
- Extração de features
- Classificação estatística.

O pré-processamento irá reduzir o ruído utilizando um filtro gaussiano de tamanho 5. A extração de features utiliza basicamente o algoritmo CamShift para obter o centróide e a localização da luva. Para possibilitar o tracking mesmo quando o objeto sai da área da imagem, utilizamos para gerar a janela de procura do objeto do CamShift, o algoritmo descrito a seguir.

Primeiro faz-se a segmentação da componente H do espaço de cores (H,S,V). A segmentação é feita utilizando um histograma normalizado obtido de uma área da luva. O algoritmo que faz a segmentação é o backproject, implementado com o cvCalcBackProject. A figura retornarda pelo backproject é então segmentada em componentes 8-conexos, denominados blobs. O blob de maior área é a luva. A janela do blob é passada ao CamShift que determina o centróide, a orientação e localização da luva.

A partir das posições x, y do centroíde calculamos o valor da velocidade entre dos frames. O espaçamento inicial do cálculo . As velocidades (v_x, v_y) são classificadas utilizando a discretização proposta em [9], mostradas na figura 2.

A classificação estatística utiliza HMMs. Para a fase de treinamento serão utilizadas 10 vídeos para cada gesto. A web cam fornece 30 quadros por segundo. Cada segundo fornecerá uma feature, de 8 valores possíveis para cada (veja figura 2). Vídeos com sequencias de 6 repetições em ordem aleatória foram gerados para avaliação do sistema.

Os critérios de avaliação serão contagem de deleção, inserção e substituição [4]. A taxa de reconhecimento será dada por $TR = 1 - \frac{D+I+S}{N_{real}}$.

A última parte, não implementada ainda, é utilizar os comandos identificados para controlar o programa.

4. Resultados

4.1. Pré-processamento

No pré-processamento, utilizá-se apenas um filtro gaussiano de tamanho 5. Neste caso obtem-se a figura 3.

4.2. Features

A saída do algoritmo backproject é visualizada na figura 4. Este algoritmo utiliza um histograma normalizado e retorna uma imagem com a probabilidade do pixel pertencer ao objeto representado pelo histograma. No caso de certeza

Figura 3. Imagem filtrada com o filtro gaussiano.

o valor é 255. No caso de não pertencer é zero. A graduação representa a próximidade da cor com o histograma. Temos ao final uma imagem com a probabilidade para cada pixel (i, j). Nesta imagem é feito um threshold para binarizar a imagem, a ser utilizada no próximo passo.

Figura 4. Imagem resultante após aplicação do BackProject.

A saída do algoritmo CamShift é visualizada na figura 5.

Repare que as figuras são obtidas na ordem, em um pipeline. Primeiro fazemos a filtragem dos ruídos com o filtro gaussiano de tamanho 5. A saída é então ligada à entrada do BackProject, que utiliza um histograma normalizado de cores. A saída do BackProject é a entrada do CamShift, que acha o centróide (em verde na imagem 5). A localização e a orientação da luva são dadas pela elipse vermelha da mesma figura.

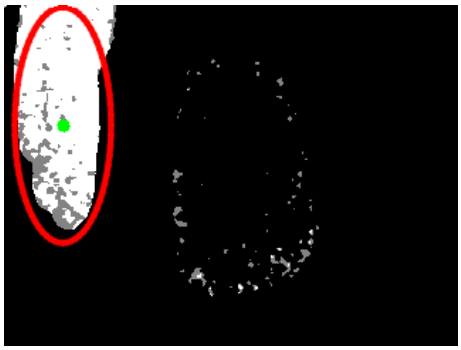

Figura 5. Imagem resultante após aplicação do CamShift.

4.3. Cadeia de Markov

A cada 30 frames fazemos a diferença entre o centroíde anterior e o centroíde atual, obtendo uma feature. Os modelos de cada gesto são obtidos através do treinamento dos HMMs respectivos, com 10 filmes do mesmo gesto. O algoritmo de treinamento utilizado foi o BaumWelch. Os dois gestos foram denominados esquerda/direita, fazendo uma alusão à mão utilizada para cada gesto. O gesto direita é semelhante ao gesto de passar uma página e o esquerda é semelhante ao voltar uma página. Após treinamento, o modelo de cada gesto é salvo e utilizado na fase de identificação. A estrutura do modelo é do tipo **esquerda para direita sem omissões** encontrado em [9].

A iniciação do algoritmo é feita com probabilidade uniformemente distríbuida. Para 2 estados e 8 gestos temos a matriz inicial de transição A, e a matriz inicial de emissão B:

$$A = \begin{pmatrix} 0.5 & 0.5 \\ 0 & 1 \end{pmatrix}$$

$$B = \begin{pmatrix} 0.125 & 0.125 & 0.125 & 0.125 & 0.125 & 0.125 & 0.125 & 0.125 \\ 0.125 & 0.125 & 0.125 & 0.125 & 0.125 & 0.125 & 0.125 & 0.125 \end{pmatrix}$$

4.4. Identificação

Nesta fase, utilizamos um buffer de tamanho máximo 10 e mínimo 2 para agrupar as features obtidas para o filme a ser testado. Este buffer alimenta o algoritmo de verificação dos HMMs. O HMM que chegar ao final com a maior probabilidade representa o movimento. Percebe-se aqui, uma complexidade da ordem de $O(hmm \times custo(hmm))$. A identificação utilizou apenas um filme com 6 movimentos. Eram na ordem: direita, direita, esquerda, esquerda, direita, esquerda. O resultado é visto na figura 6. O algoritmo para teste da sequencia de observações é o Viterbi, veja [7].

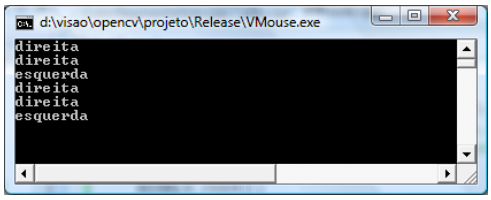

Figura 6. Resultado da Identificação.

Isto fornece segundo a formula descrita na metodologia uma taxa de acerto de $\frac{5}{6}$, que deve ser testada mais vezes, para obtermos uma taxa mais real.

5. Conclusões

O trabalho mostrou-se eficaz na detecção da gramática proposta para o arquivo de teste especificado. Deve-se evidenciar, que mais testes devem ser feitos para averiguação da real taxa de acerto. O sistema funciona em tempo real, e toda a parte de visão foi desenvolvida em cima do OpenCV. O algoritmo de tracking se mostrou robusto para o caso em que o objeto sai e retorna da cena, diferentemente do uso apenas do CamShift, que espera que o objeto volte próximo do lugar que deixou a cena, o que produzia resultados ruins para o arquivo de teste, uma vez que alguma features eram perdidas devido a não localização.

6. Trabalhos Futuros

Devido ao problema da complexidade aumentar com o número de gestos reconhecidos, para trabalhos futuros pode-se avaliar o desempenho da identificação de gestos utilizando o Condensation tanto para o tracking quanto para a identificação do gesto, como em [5]. Outra opção é sabendo a posição do blob luva identificar os gestos da mão(move dedo, faz um "joia") de cada mão, representando novas features. Estas features serão também treinadas pelos HMMs, fornecendo mais gestos.

Referências

[1] J. G. Allen, R. Y. D. Xu, and J. S. Jin. Object tracking using camshift algorithm and multiple quantized feature spaces. In *VIP '05: Proceedings of the Pan-Sydney area workshop on Visual information processing*, pages 3–7, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.

[2] D. Becker and A. Pentland. Staying alive: A virtual reality visualization tool for cancer patients, 1996.

[3] M. J. Black and A. D. Jepson. A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions. In H. Burkhardt and B. Neumann, editors, *European Conf. on Computer Vision, ECCV-98*, volume 1406 of *LNCS-Series*, pages 909–924, Freiburg, Germany, 1998. Springer-Verlag.

[4] M. Boros, W. Eckert, F. Gallwitz, G. Görz, G. Hanrieder, and H. Niemann. Towards understanding spontaneous speech: Word accuracy vs. concept accuracy. In *Proc. ICSLP '96*, volume 2, pages 1009–1012, Philadelphia, PA, 1996.

[5] A. H. Gruenstein. Using a particle filter for gesture recognition, 2002.

[6] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[7] D. Lin. A c++ implementation of hidden markov model, 2003.

[8] V. I. Pavlovic, R. Sharma, and T. S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):677–695, 1997.

[9] R. M. Resende. Desenvolvimento de uma interface Humano-Robô utilizando visão computacional e Sistemas a Eventos Discretos. Master's thesis, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Agosto 2005.

[10] G. Rigoll, A. Kosmala, and S. Eickeler. High performance real-time gesture recognition using hidden markov models. In *Proceedings of the International Gesture Workshop on Gesture and Sign Language in Human-Computer Interaction*, pages 69–80, London, UK, 1998. Springer-Verlag.

[11] B. C. Vieira. Vremote, 2008.

[12] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: real-time tracking of the human body. *fg*, 00:51, 1996.

[13] Y. Wu and T. Huang. Hand modeling, analysis and recognition. *IEEE Signal Processing Magazine*, 18(3):51 – 60, 2001.