Interação Humano-Robô

Humberto Henrique Campos Pinheiro
Universidade Federal de Minas Gerais
humb@dcc.ufmg.br

1. Introdução

Interação Humano-Robô é o campo de estudo dedicado ao entendimento, projeto e avaliação de sistemas robóticos para o uso por humanos. Interação, por definição, requer comunicação entre humanos e robôs. Comunicação entre um humano e um robô pode tomar várias formas de acordo com a proximidade dos dois e pode ser classificada entre interação remota e interação próxima. Interação remota é chamada de teleoperação ou controle supervisionado e se envolve um manipulador físico também é chamada de telemanipulação. Outras formas de interação incluem interação física e social, que inclui aspectos sociais, cognitivos e emotivos [1].

O objetivo deste trabalho é utilizar técnicas de Visão Computacional para interagir com um robô através de expressões gestuais. Mais precisamente será mostrada uma implementação de um esquema que consiste na utilização de um framework para reconhecimento de gestos para acionar um robô através de uma interface em software via protocolo TCP-IP.

2. Revisão Bibliográfica

Em [5] os autores sugerem uma interface gestual que controla o movimento de um robô móvel usando um conjunto de gestos manuais pré-definidos. Basicamente uma câmera a cores é utilizada para capturar as mãos e a cabeça do operador. Comandos são organizados em um padrão gestual usando as duas mãos.

As características da imagem são passadas para uma rede neural treinada. Uma rede neural é um modelo computacional que tenta simular a funcionalidade de uma rede neural biológica, consistindo de um grupo de elementos (neurônios) conectados. Uma rede neural é adaptativa, podendo mudar sua estrutura baseada no treinamento. Depois de reconhecido o padrão gestual é mapeado para um comando e repassado ao robô.

Em [4] os autores apresentam uma abordagem utilizando cadeias escondidas de Markov para reconhecer gestos de indicação, isto é, onde a pessoa usa a mão e braço para indicar direções. Eles utilizam uma câmera estéreo calibrada para obter informação sobre cor e disparidade para inferir posições das mãos e orientação da cabeça.

Uma cadeia escondida de Markov é todo processo aleatório com propriedade de Markov em que os estados não são diretamente visíveis. A propriedade de Markov corresponde ao fato de que os estados futuros não dependem de estados passados mas só dos estados presentes. As características passadas ao sistema de reconhecimento podem ainda utilizar informação visual extra adicionada por um humano para aperfeiçoar a taxa de acerto no reconhecimento.

Em [3] a autora cria uma interface visual humano-robô utilizando uma linguagem composta por gestos simples onde uma sequência de gestos significa uma mensagem completa e gera uma resposta. A intenção é a criação de uma gramática e a interação mais intuitiva com o robô. A gramática consistia em gestos simples como movimentar a mão para baixo, para cima ou para a direita. O sistema de visão foi baseado na segmentação por cor, como nas outras referências citadas.

As imagens capturadas pela câmera passam primeiro por uma fase de pré-processamento ondo os ruídos de alta frequência são minimizados e as cores uniformizadas. Em seguida os pixels da imagem são agrupados em blocos de intensidade e cor próximos, chamados *blobs*. O algoritmo de rastreamento utiliza o conceito de objeto de interesse - o *blob* de maior área. Considerando o vetor deslocamento entre os quadros constante o próximo centro de massa do objeto de interesse é calculado. O vetor deslocamento é calculado de acordo com a diferença do centro de massa do *blob* do quadro atual e o do quadro anterior. A probabilidade das transições de cada gesto na cadeia de Markov é determinada de acordo com o ângulo do vetor deslocamento. A autora ainda comparou a implementação da metodologia utilizando tanto cadeias de Markov quanto cadeias escondidas de Markov.

3. Metodologia

A metodologia consiste nos seguintes passos: capturar as imagens, identificar o gesto a partir da imagem, atribuir um comando ao gesto e passá-lo para a interface com o robô. Um esquema simplificado pode ser visto na figura 1.

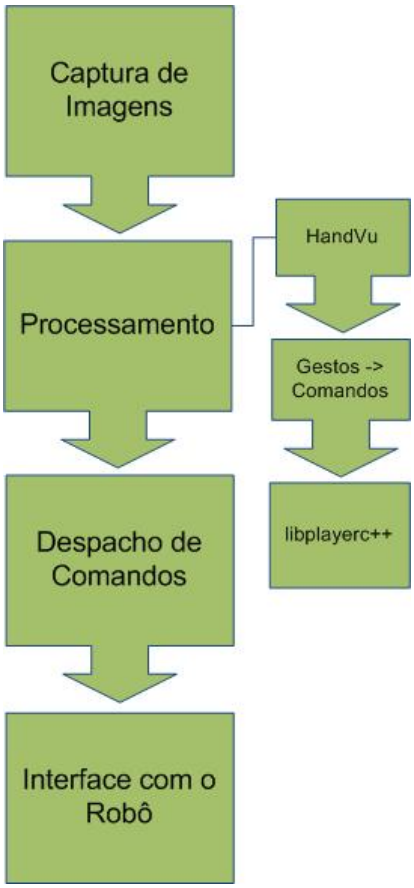

Figura 1. Metodologia

As imagems foram capturadas usando uma webcam Microsoft Lifecam vx500 com resolução de 640 por 480 pixels a 30 quadros por segundo. O setup consiste de um computador com um processador Core 2 Duo 2,6 ghz e 3 GB de memória RAM.

A parte de visão foi feita utilizando o framework HandVu [2]. Este módulo é constituído de três fases: detecção de mão, rastreamento de mão e reconhecimento gestual.

O detector de mão do HandVu é uma combinação de uma adaptação do método de Viola-Jones com um esquema de verificação de cor da pele. Para o treinamento, um grande conjunto de imagens de mão - 2300 imagens - foi coletado em várias configurações; o processamento foi feito em um cluster de máquinas Linux de forma paralelizada.

O autor implementou um esquema paralelizado de treinamento utilizando o *AdaBoost* em dois clusters de máquinas Linux, um com 16 nós e o outro com 32 nós, cada nó consistindo de dois processadores.

O *AdaBoost* é um algoritmo de aprendizagem de máquina adaptativo. Basicamente ele chama um classificador fraco repetidas vezes em série onde em cada etapa uma distribuição de pesos é atualizada de acordo com a importância dos exemplos para a classificação, de modo que classificadores de pouca importância são descartados e classificadores fortes são promovidos.

No HandVu o autor criou um novo tipo de característica baseado em retângulos como classificadores fracos. Para cada postura, um detector em cascata foi treinado para selecionar os seus classificadores fracos de acordo com quatro características - ao invés de três usadas no algoritmo original de Viola-Jones [6].

No segundo estágio o objetivo é seguir a mão detectada ao longo do vídeo. Para isso o autor criou a técnica de "Flock of Features", um método rápido de rastreamento para objetos não rígidos e altamente articulados. O método é baseado em uma conbinação de *optical flow* com uma distribuição probabilística de cores.

Essencialmente, *optical flow* corresponde ao movimento aparente dos objetos da cena que resulta da movimentação do observador em relação à cena. Esse movimento aparente pode nos dar informações sobre as distâncias dos objetos em relação ao observador uma vez que objetos mais próximos aparentam se mover mais rapidamente que objetos mais distantes.

A idéia central do rastreador (*Flock of features*) é motivado pelo aparentemente caótico movimento de voo de um bando de pássaros, em que um pássaro sozinho não tem nenhum controle global mas o bando inteiro ainda permanece unido. Essa organização descentralizada resulta de duas restrições que podem ser calculadas localmente: um pássaro mantém uma distância mínima e uma distância máxima dos outros. O rastreador de mão consiste em um conjunto de pequenas áreas da imagem movendo de um quadro para o outro de maneira similar ao bando de pássaros. Os caminhos que cada característica "segue"são determinados por *optical flow* e restringidos de acordo com uma distância mínima e máxima. Se essas condições não são satisfeitas a característica é reposicionada em uma localização que tem uma cor parecida.

O terceiro estágio consiste no reconhecimento do gesto. O método foca mais na confiabilidade do que na expressividade, de modo que reconhece apenas seis posturas, a saber: mão fechada, mão aberta, sinal de

"V", sinal de indicação com o dedo indicador, sinal de "L"formado pelos dedos indicador e polegar (palma e costas da mão).

O método utiliza uma abordagem baseada em textura para classificar as áreas da imagem sem sete classes, seis postura e uma "postura desconhecida". Uma modificação do algoritmo de Viola-Jones foi implementada de modo que no primeiro passo o detector tenta encontrar um dos seis gestos sem distinguir entre eles. Com isso elimina-se muitos candidatos a gestos incorretos de forma rápida e eficiente. No segundo passo apenas as áreas que passaram pelo primeiro são verificadas. A verificação consiste em utilizar os exemplos positivos para um classificador como exemplos negativos para todos os outros (*cross-training*), de modo que se confirme que as classes são suficientemente distintas. O treinamento então utiliza o mesmo algoritmo *AdaBoost* explicitado anteriormente.

Um programa foi implementado em C++ utilizando as bibliotecas do OpenCV. Basicamente o programa consiste de três módulos: um para a integração com o HandVu, um módulo para comunicação via TCP/IP e o outro para integração com a interface do robô. A saída do processamento do HandVu é mapeada em comandos e passada para a interface com o robô via TCP/IP. Os gestos eram mapeados nos seguintes comandos:

- Mão fechada - mover o robô para frente
- Mão aberta - mover o robô para trás
- Sinal de L (palma) - rotacionar o robô para a esquerda
- Sinal de L (costas) - rotacionar o robô para a direita

Para a interface com o robô foi utilizado o Projeto Player, um software livre para pesquisa e ensino na área de robótica. O *player* é um servidor de controle de robôs e pode usar simuladores. Neste trabalho foi usado o *Stage* para simular um mundo 2D com um robô.

4. Resultados e Conclusão

Em um computador com processador Core 2 Duo de 2,6 Ghz e 3 GB de memória RAM o programa era capaz de processar cinco quadros por segundo, permitindo uma interação razoável em tempo real. Notou-se que o HandVu é também robusto, conseguindo identificar corretamente os gestos mesmo variando as condições de luminosidade e o ângulo da mão. A quantidade de ações que o robô é capaz de fazer é pequena devido às limitações do simulador, em uma configuração com robôs reais poderia se especificar um conjunto de comandos mais úteis, de acordo com a necessidade. Entretanto o número de ações ainda estaria limitado pelo número de gestos que o HandVu é capaz de reconhecer. Acrescentar mais gestos é uma tarefa dispendiosa visto que o treinamento requer uma grande quantidade de amostras e poder computacional, como foi explicitado anteriormente. Uma abordagem para contornar isso seria criar uma gramática gestual utilizando essas seis posturas, conforme mostrado em [3], permitindo assim que o robô execute uma quantidade arbitrária de ações diferentes independente do número de posturas distintas identificadas pelo HandVu.

Referências

[1] M. A. Goodrich and A. C. Schultz. Human-robot interaction: A survey. *Foundations and Trends in Human-Computer Interaction*, 1(3):203–275, 2007.

[2] M. Kölsch. Vision based hand gesture interfaces for wearable computing and virtual environments, 2004.

[3] R. Maíra Resende. Desenvolvimento de uma interface humano-robô utilizando visão computacional e sistemas a eventos discretos, 2006.

[4] K. Nickel and R. Stiefelhagen. Real-time person tracking and pointing gesture recognition for human-robot interaction. In *Computer Vision in Human-Computer Interaction*, pages 28–38, 2004.

[5] V. Paquin and P. Cohen. A vision-based gestural guidance interface for mobile robotic platforms. In *Computer Vision in Human-Computer Interaction*, pages 39–47, 2004.

[6] P. Viola and M. Jones. Robust real-time object detection. *Workshop on Statistical and Computational Theories of Vision*, July 2001.