

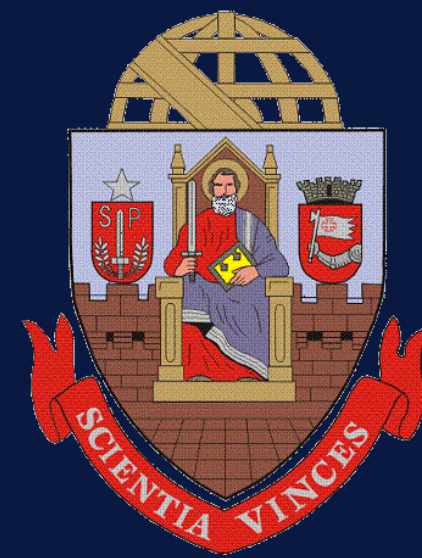


# Vitruvius

## Um Reconhecedor de Gestos para o Kinect™

Aluno: Tiago Andrade Togores

Orientador: Professor Flávio Soares Corrêa da Silva



### Introdução

Interfaces naturais permitem que usuários se comuniquem com dispositivos de forma mais intuitiva, com uma curva de aprendizado baixa. Em 2010, o Kinect popularizou esse tipo de interação em computadores pessoais. O objetivo deste trabalho é desenvolver um sistema que permita treinamento e reconhecimento de gestos pelo usuário final, pois o que foi criado até hoje se baseia em reconhecer um conjunto fixo de gestos ou permite a adição de gestos apenas simples. Para tanto, será necessário o emprego de uma técnica de inteligência artificial adequada.

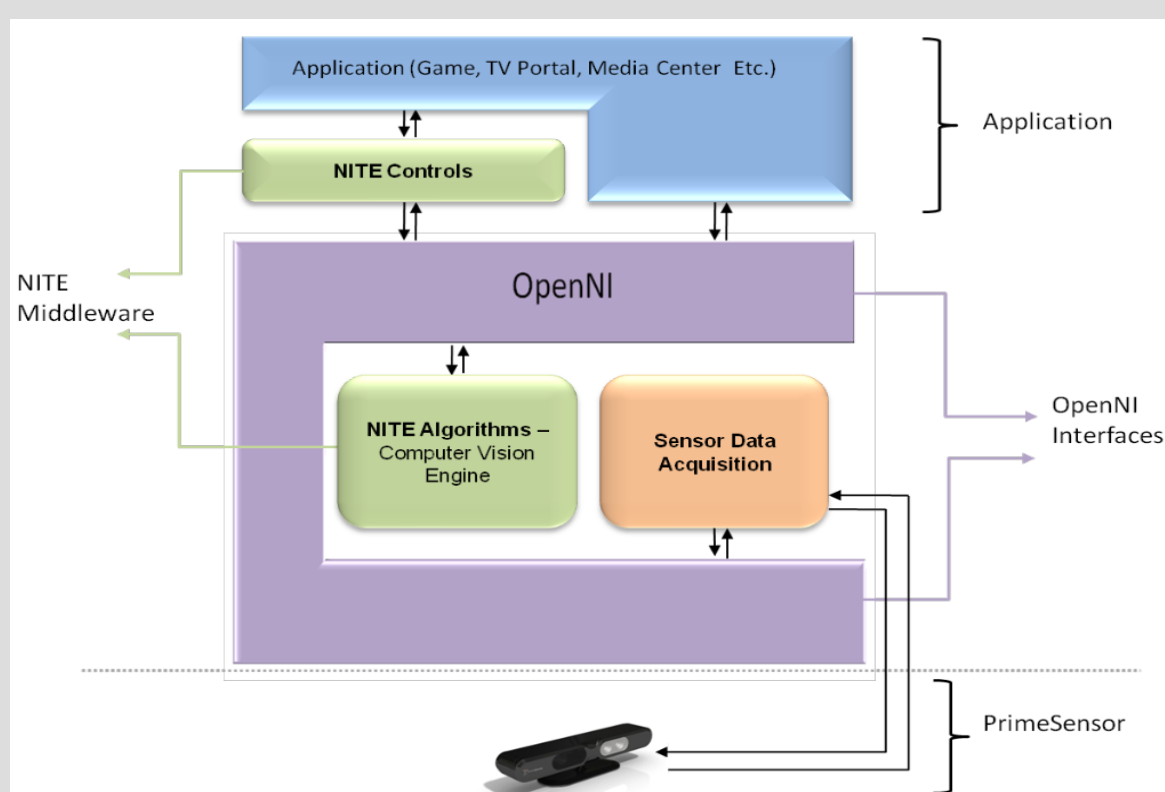
### Kinect

É um periférico desenvolvido pela Microsoft para uso no XBOX 360. Possui uma câmera RGB, um sensor de profundidade, que gera frames em que cada pixel representa uma distância, quatro microfones e um motor na base para inclinação.

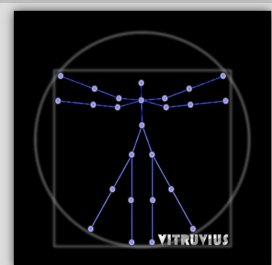


### OpenNI

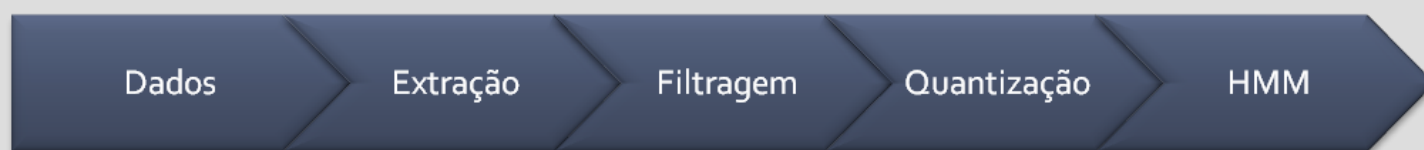
É um framework que provê uma API que serve como uma camada intermediária entre dispositivos de baixo nível e soluções middleware de alto nível. Para poder utilizar o Kinect, é utilizado um driver da PrimeSense e o middleware NITE, responsável pelo trabalho de visão computacional (interpretação de dados de profundidade do sensor, detecção de usuários e rastreamento das articulações deles).



### Vitruvius



Programa que faz uso das ferramentas acima para obter os gestos. Utiliza o seguinte pipeline para processamento:



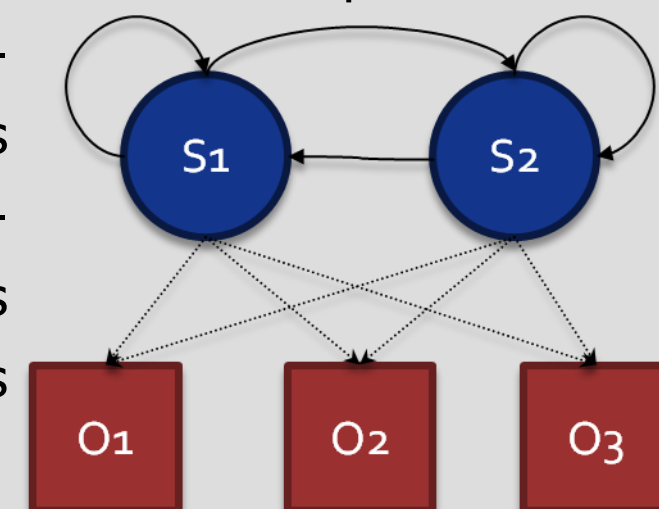
**Dados:** são os dados gerados pelo NITE, ou seja, posição e orientação das articulações de cada usuário sendo rastreado no momento.

**Extração:** extrai vetores de características. Como os gestos considerados aqui são os realizados com a mão direita, consiste em combinar e normalizar apenas a posição, orientação e velocidade dela.

**Filtragem:** informações no conjunto de vetores podem ser irrelevantes às vezes. Aplica-se, então, filtros para detectar se houve alteração significativa entre valores consecutivos, por exemplo.

**Quantização:** realiza o mapeamento de vetores para números inteiros no intervalo  $0 \dots N - 1$ . É utilizado o algoritmo de clusterização k-means, que divide o espaço vetorial em N partições, associando um representante a cada uma delas.

**HMM:** O modelo oculto de Markov representa um processo Markoviano duplamente estocástico, onde os estados são ocultos e só podem ser inferidos através das observações, emitidas com certa probabilidade.



Supondo que cada modelo  $M_i$  representa um gesto e foi executado por um usuário o gesto  $O$ , temos:

- Reconhecimento: com algoritmo forward, determinamos o gesto executado por  $\arg \max(i) P(O|M_i)$ .
- Treinamento: com o algoritmo de Baum-Welch, reestimamos as matrizes de transição e emissão maximizando localmente  $P(O|M_i)$ .

### Referências Bibliográficas

- L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257-286, 1989.
- VM. Mantyla. Discrete hidden markov models with application to isolated user-dependent hand gesture recognition. *VTT publications*, 2001.