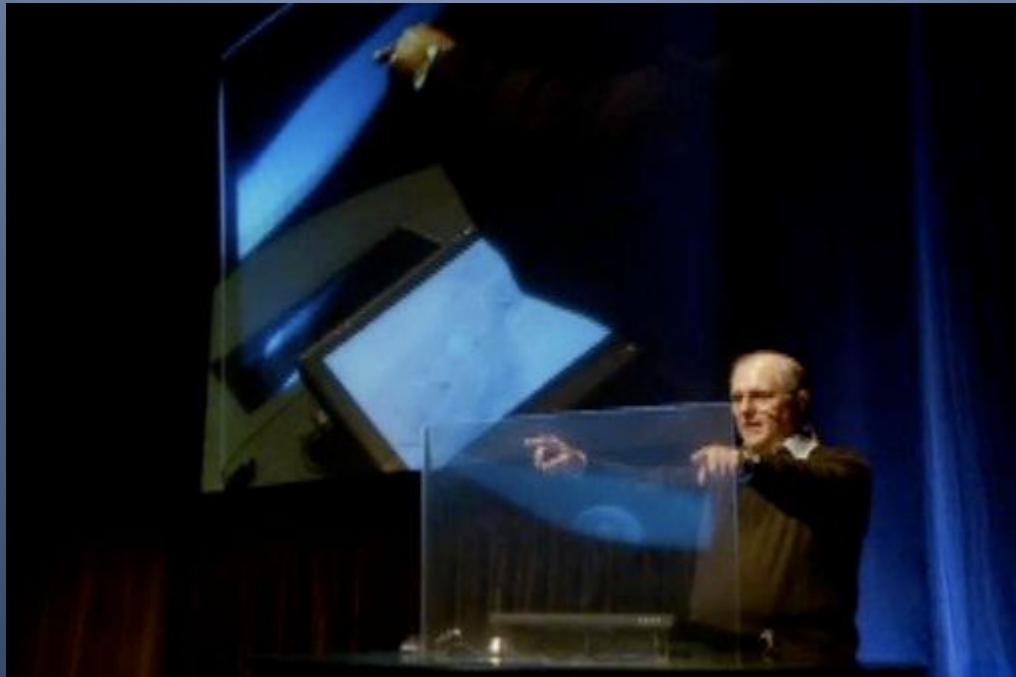


Aluno: Tiago Andrade Togores

Orientador: Professor Flávio Soares Côrrea da Silva

Vitruvius: Um Reconhecedor de Gestos para o Kinect

Introdução



Contextualização

- Interação humano-computador

Contextualização

- Interação humano-computador
- Interfaces naturais

Contextualização

- Interação humano-computador
- Interfaces naturais
- Popularização
 - Wii
 - Smartphones
 - Tablets
 - Kinect

Objetivos

- Desenvolvimento de um programa que
 - reconheça gestos
 - treine gestos
 - alcance usuários comuns

Objetivos

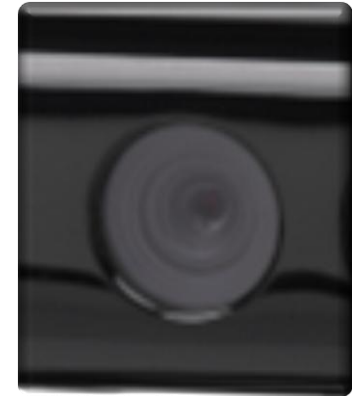
- Desenvolvimento de um programa que
 - reconheça gestos -> inteligência artificial
 - treine gestos -> aprendizado computacional
 - alcance usuários comuns -> Windows

Kinect



Hardware

- Câmera RGB
 - Resolução: 640x480
 - Taxa de atualização: 30 fps
 - Field of view (vertical): 43°
 - Field of view (horizontal): 57°
 - Stream: 32-bit

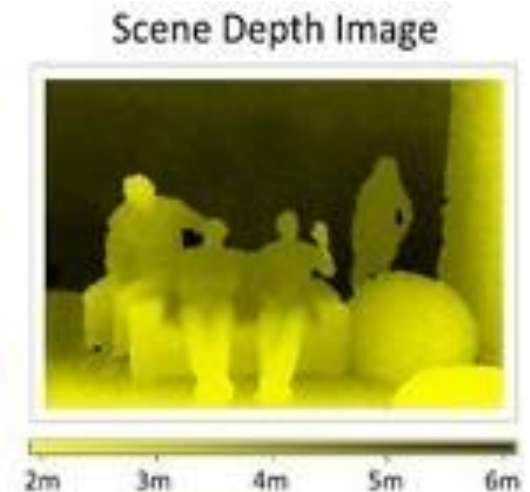


Scene



Hardware

- Sensor de profundidade
 - Resolução: 640x480
 - Taxa de atualização: 30 fps
 - Field of view (vertical): 43°
 - Field of view (horizontal): 57°
 - Stream: 16-bit



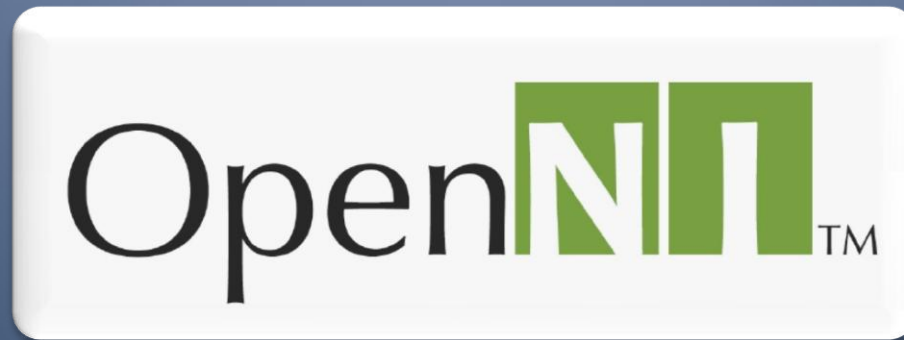
Hardware

- Microfones
- Motor



OpenNI

Open Natural Interaction



OpenNI

- Organização
- Interoperabilidade:
 - Dispositivos
 - Aplicações
 - Middleware

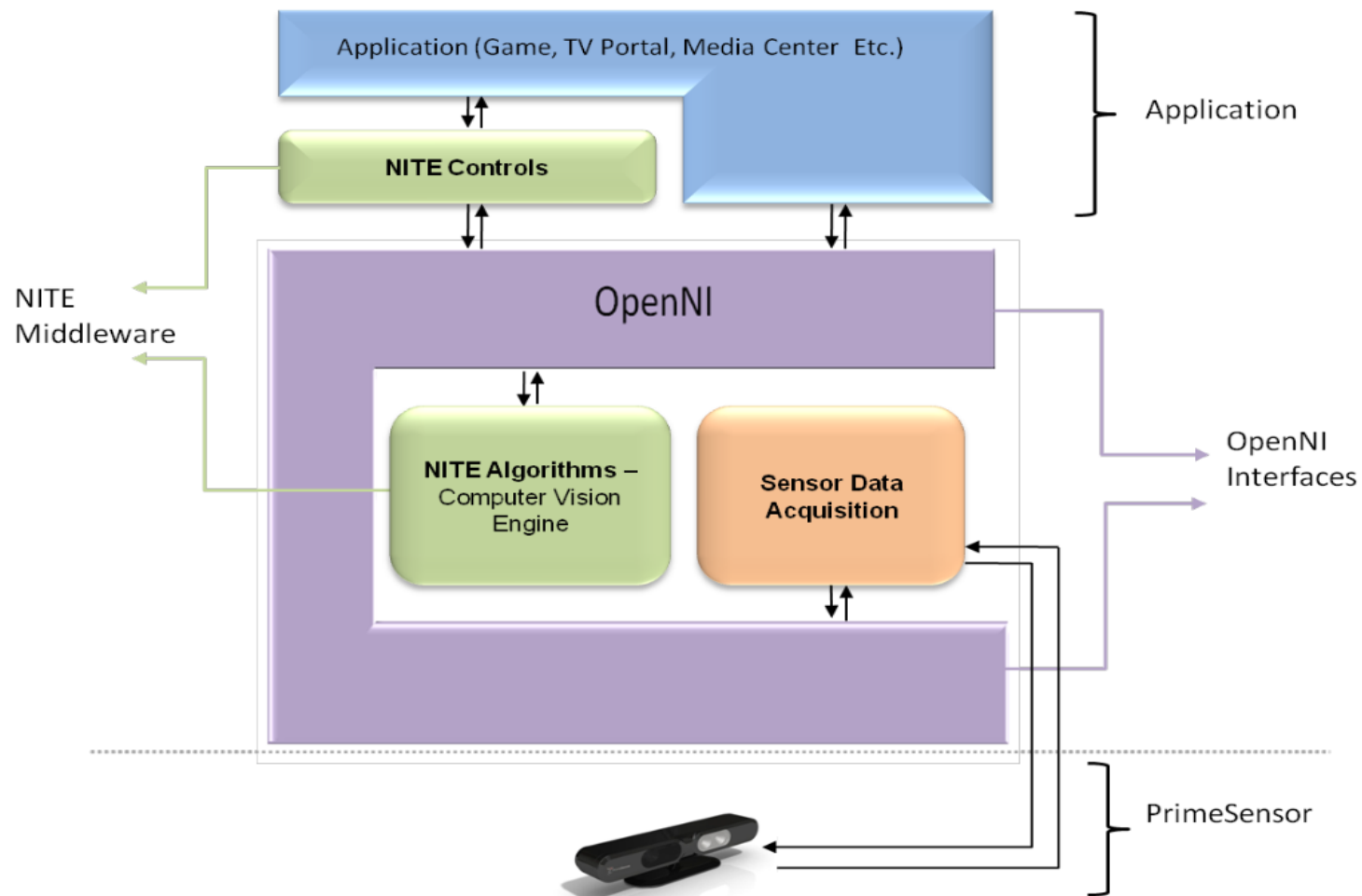
OpenNI Framework

- API - camada intermediária
- Production Chain
- Multi-plataforma
- Várias linguagens
- Gravação dados
- Uso com diferentes dispositivos

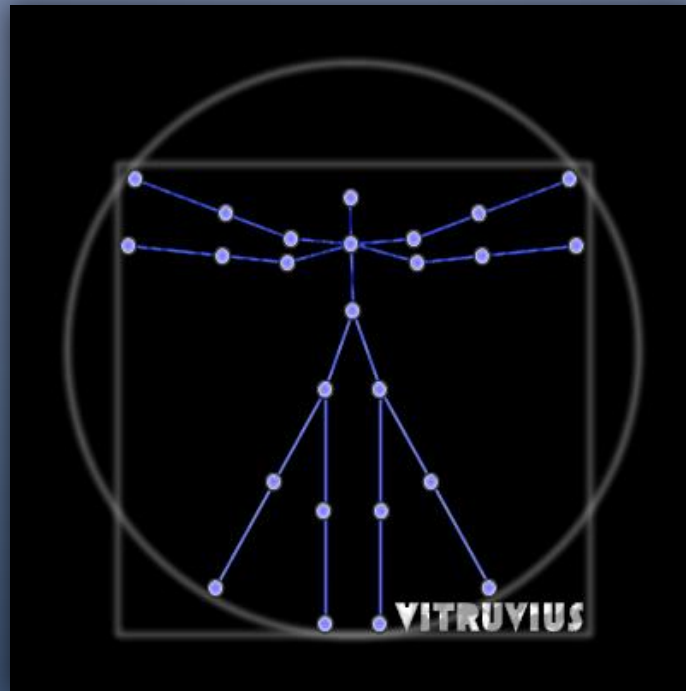
NITE

- Middleware
- Visão computacional
- Detecção de usuários
- Rastreamento
- 8 ativos

Arquitetura

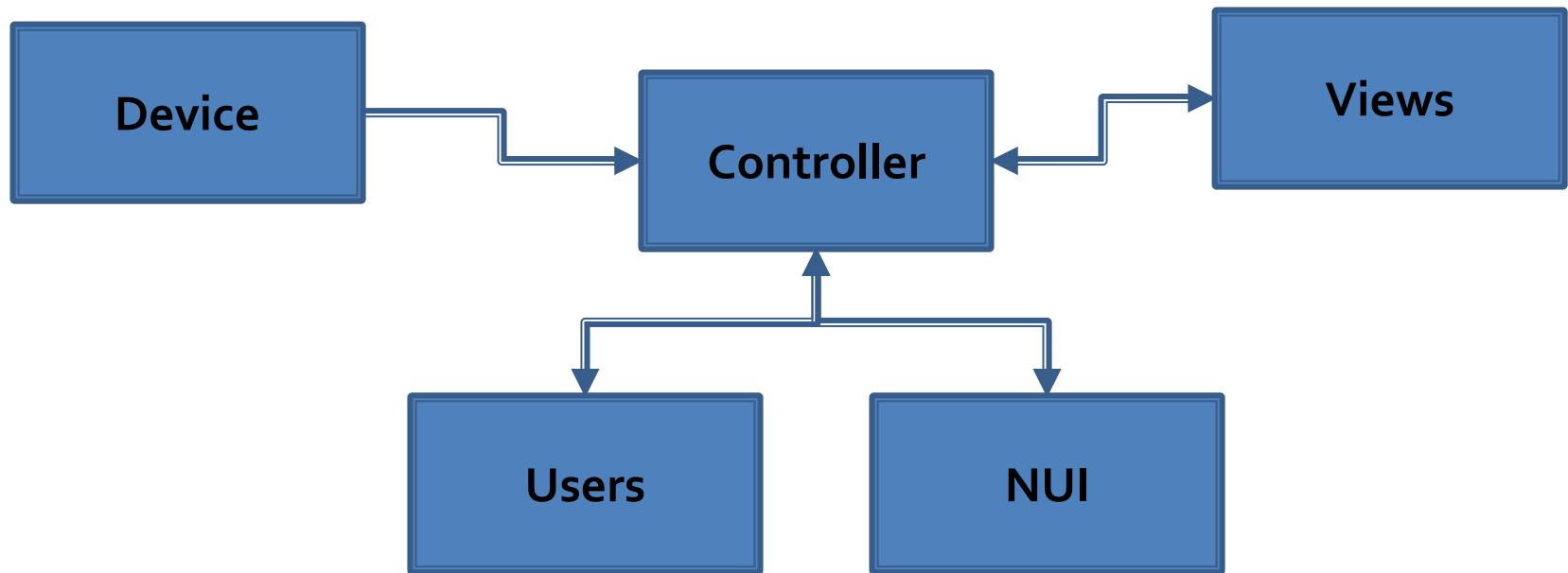


Vitruvius

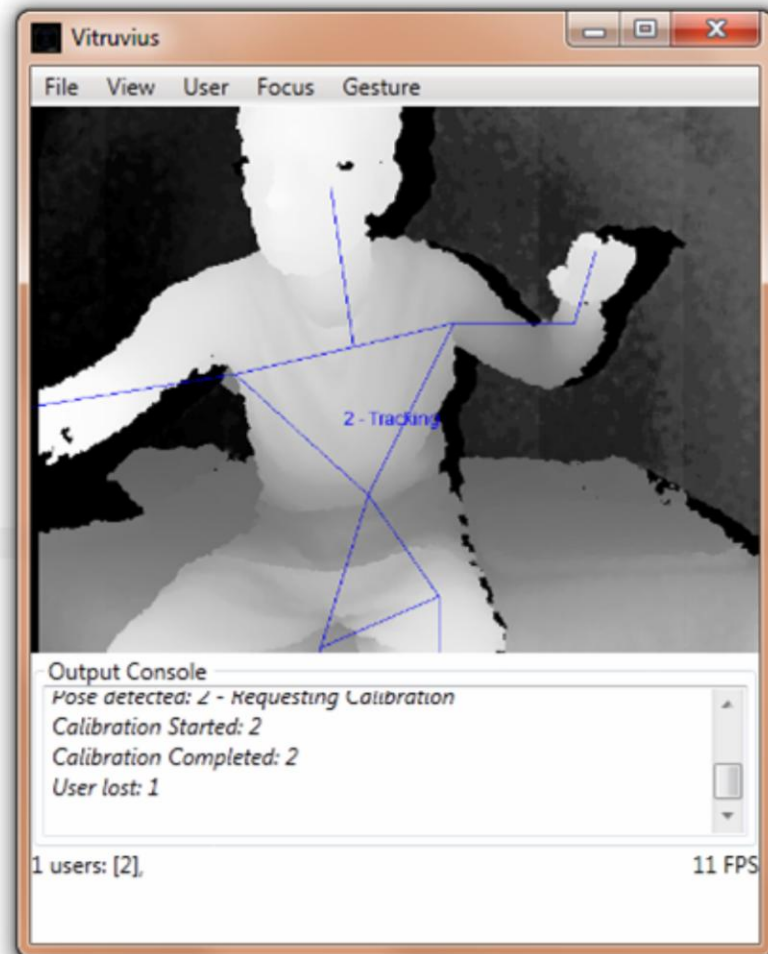


Arquitetura

- Model-View-Presenter (Passive View)



Interface



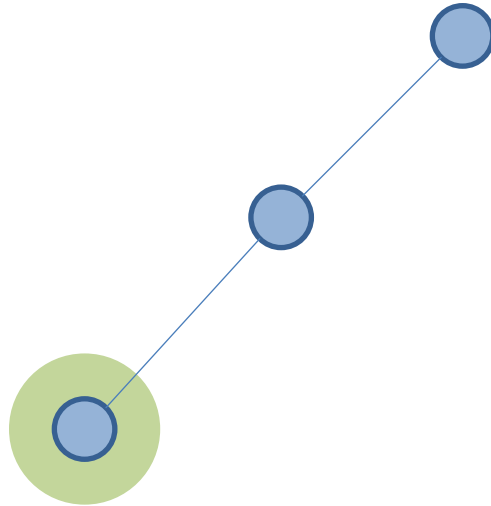
Pipeline



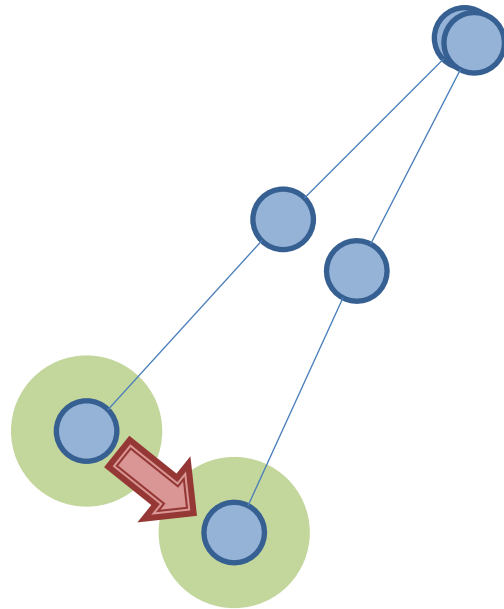
Extração de Características

- Características:
 - Posição
 - Orientação
 - Velocidade
 - Aceleração
- Escolha delas é MUITO IMPORTANTE

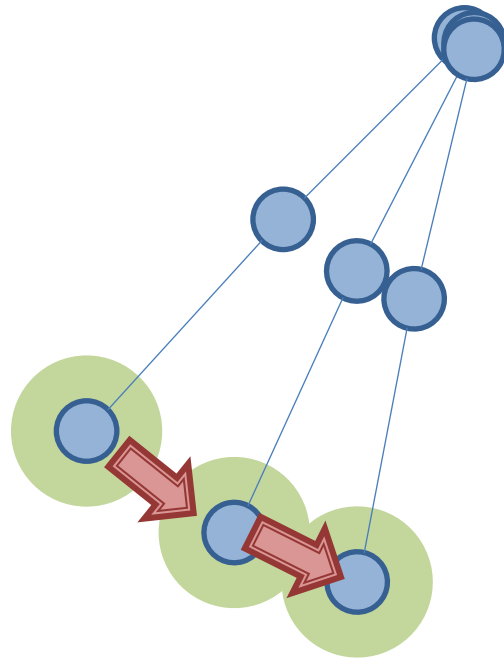
Extração de Características



Extração de Características



Extração de Características



Filtragem

- Informação Redundante
- Representação mínima
- Filtro de semelhança

Quantização

- Mapeamento de vetores para números inteiros

$$Q : \mathbb{R}^m \mapsto \mathbb{Z}_n$$

Quantização

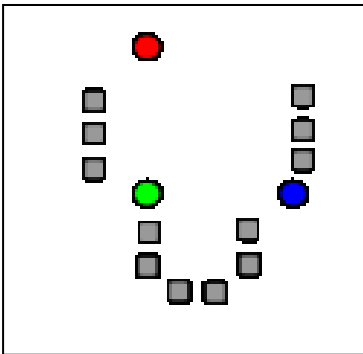
- Mapeamento de vetores para números inteiros

$$Q : \mathbb{R}^m \mapsto \mathbb{Z}_n$$

- Particionar o espaço

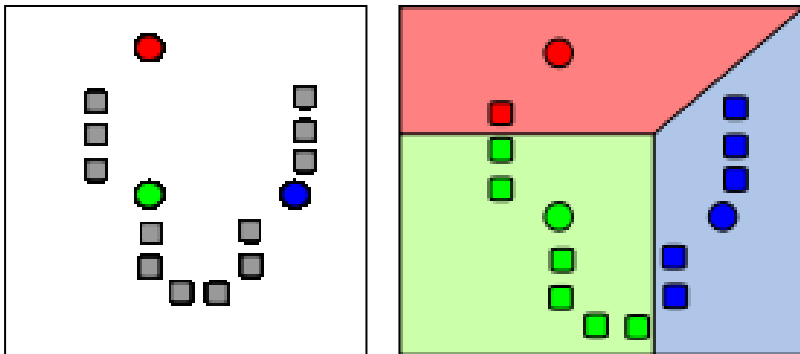
Quantização

- Algoritmo K-Médias



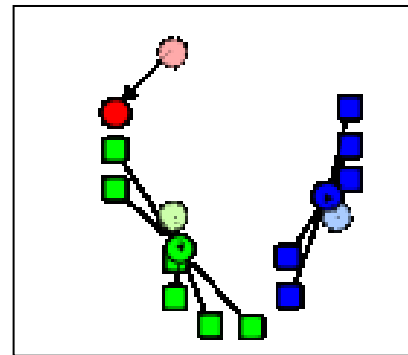
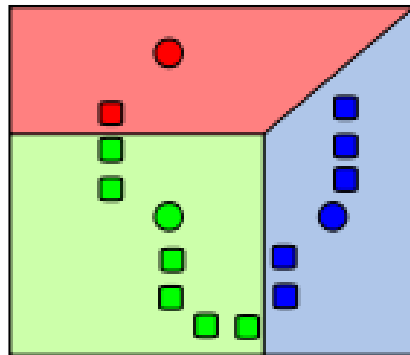
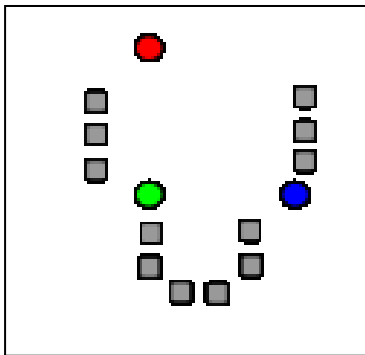
Quantização

- Algoritmo K-Médias



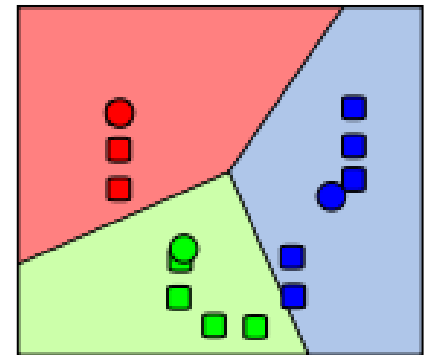
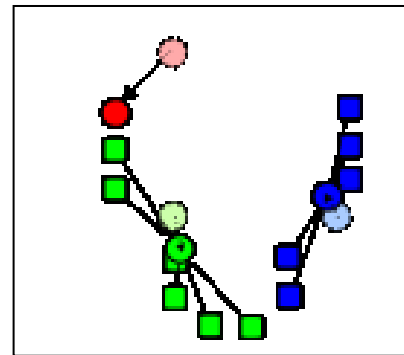
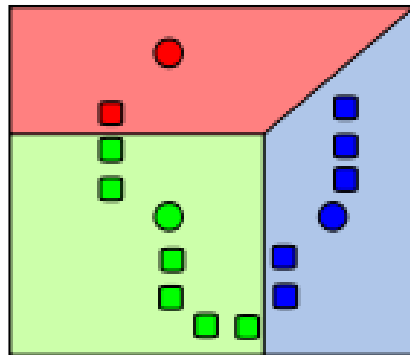
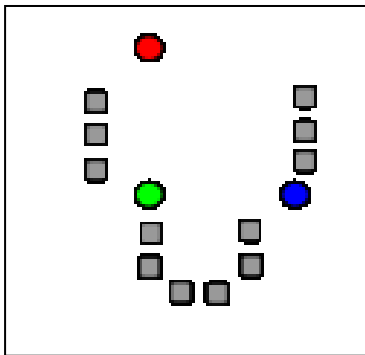
Quantização

- Algoritmo K-Médias



Quantização

- Algoritmo K-Médias



Hidden Markov Model (HMM)

- Cada modelo representa um gesto

Hidden Markov Model (HMM)

- Cada modelo representa um gesto
- Processo duplamente estocástico
- Estados ocultos variam com o tempo
- Observações emitidas pelo estados
- Parâmetros conhecidos
- Propriedade de Markov

HMM - Definição

$$\lambda = (N, M, A, B, \pi)$$

$$S = \{1, \dots, N\}$$

$$V = \{1, \dots, M\}$$

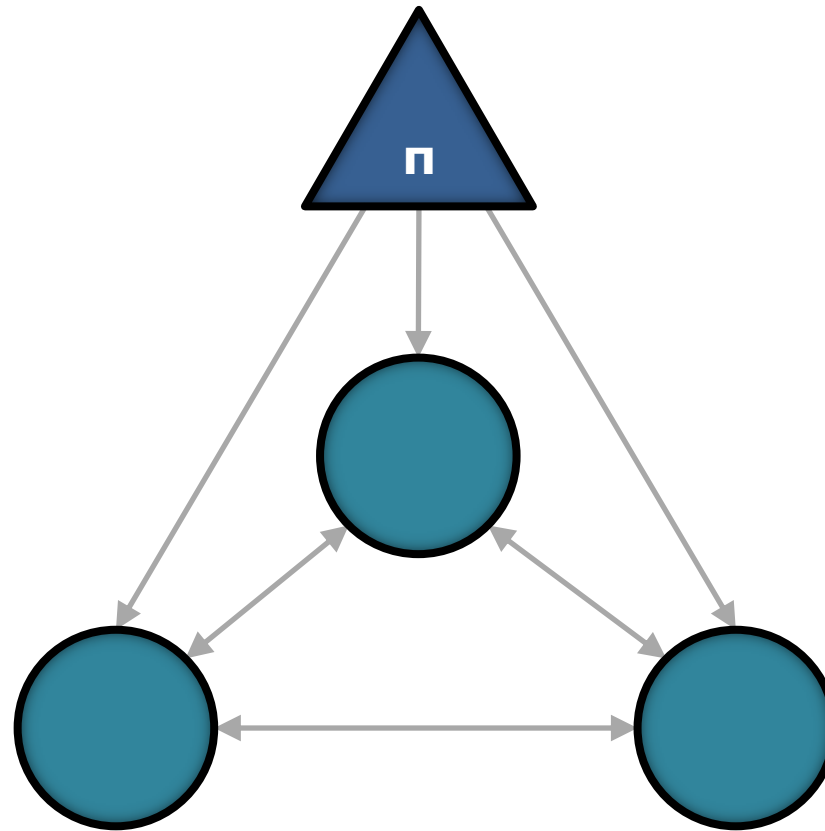
$$a_{ij} = P(X_{t+1} = j | X_t = i)$$

$$b_j(k) = P(Y_t = k | X_t = j)$$

$$\pi_i = P(X_1 = i)$$

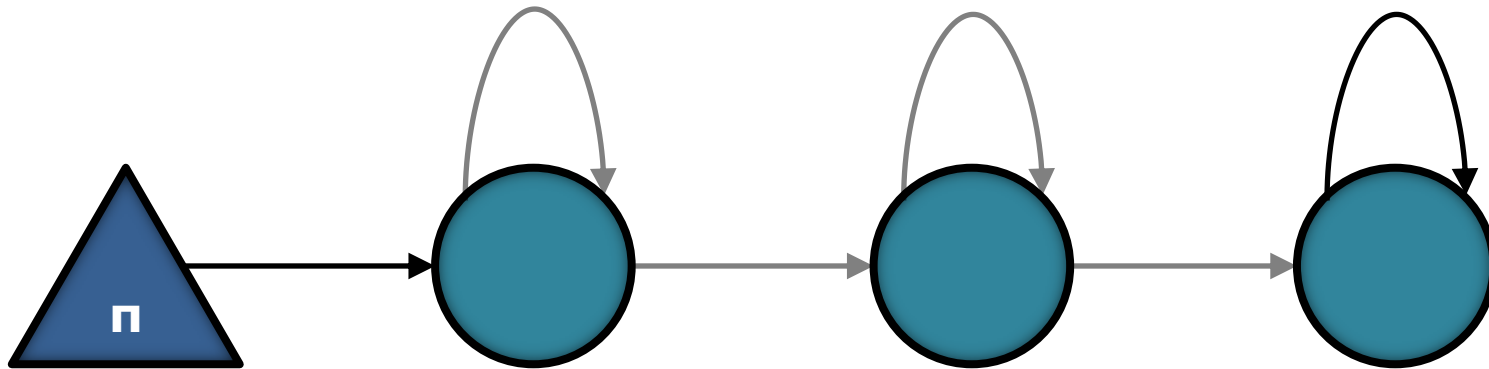
HMM - Topologia

- Ergódica

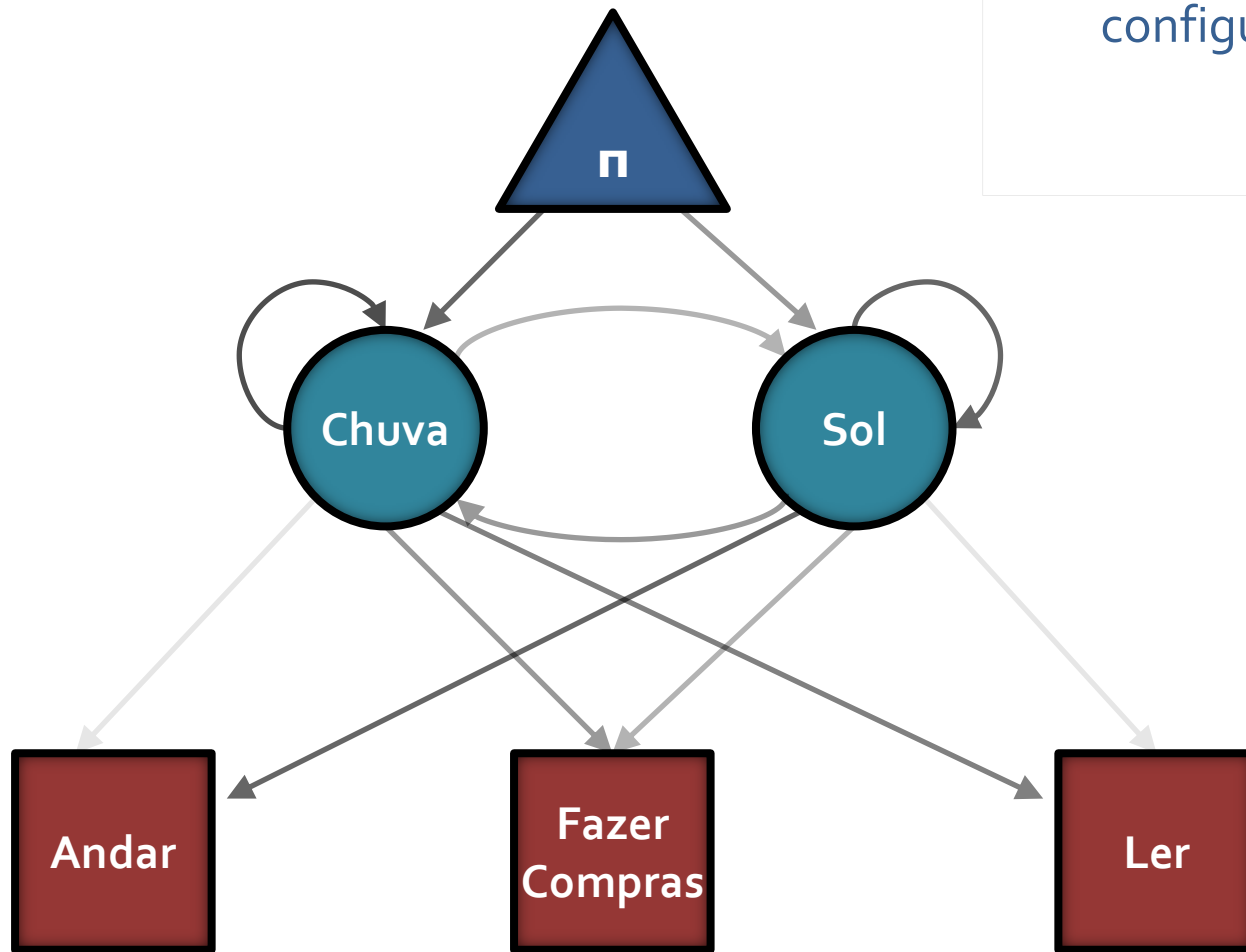


HMM - Topologia

- Bakis



HMM - Exemplo



configuração inicial
estados
observações

HMM - Problemas Canônicos

- Estimação
- Decodificação
- Aprendizagem

HMM - Problemas Canônicos

- Estimação -> **Reconhecimento**
- ~~Decodificação~~
- Aprendizagem -> **Treinamento**

HMM - Estimação

- Problema: computar eficientemente a probabilidade de uma sequência de observações dado um HMM

$$P(O|\lambda) = P(O_1, \dots, O_T|\lambda)$$

HMM - Estimação

- Variável Forward:

$$\alpha_t(i) = P(O_1, \dots, O_t, X_t = i | \lambda)$$

HMM - Estimação

- Variável Forward:

$$\alpha_t(i) = P(O_1, \dots, O_t, X_t = i | \lambda)$$

- Por Indução:

$$\alpha_1(i) = \pi_i b_i(O_1), 1 \leq i \leq N$$

$$\alpha_{t+1}(j) = [\sum_{i=1}^N \alpha_t(i) a_{ij}] b_j(O_{t+1}),$$
$$1 \leq t \leq T - 1, 1 \leq j \leq N$$

HMM - Estimação

- Variável Backward:

$$\beta_t(i) = P(O_{t+1}, \dots, O_T | X_t = i, \lambda)$$

HMM - Estimação

- Variável Backward:

$$\beta_t(i) = P(O_{t+1}, \dots, O_T | X_t = i, \lambda)$$

- Por Indução:

$$\beta_T(i) = 1, 1 \leq i \leq N$$

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(O_{t+1}) \beta_{t+1}(j),$$
$$1 \leq t \leq T - 1, 1 \leq j \leq N$$

HMM - Estimação

- Solução:

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i)$$

$$P(O|\lambda) = \sum_{i=1}^N \pi_i \beta_1(i) b_i(O_1)$$

HMM - Estimação

- Solução:

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i)$$

$$P(O|\lambda) = \sum_{i=1}^N \pi_i \beta_1(i) b_i(O_1)$$

- Complexidade: $O(N^2T)$

Classificador Bayesiano

- Temos $P(O|\lambda_i)$

Classificador Bayesiano

- Temos $P(O|\lambda_i)$
- Queremos $\lambda^* = \operatorname{argmax}_{\lambda_i} P(\lambda_i|O)$

Classificador Bayesiano

- Temos $P(O|\lambda_i)$
- Queremos $\lambda^* = \operatorname{argmax}_{\lambda_i} P(\lambda_i|O)$
- Bayes:
$$P(\lambda_i|O) = \frac{P(\lambda_i)P(O|\lambda_i)}{P(O)}$$

Classificador Bayesiano

- Temos $P(O|\lambda_i)$
- Queremos $\lambda^* = \operatorname{argmax}_{\lambda_i} P(\lambda_i|O)$
- Bayes:
$$P(\lambda_i|O) = \frac{P(\lambda_i)P(O|\lambda_i)}{P(O)}$$
$$\lambda^* = \operatorname{argmax}_{\lambda_i} P(O|\lambda_i)$$

HMM - Aprendizagem

- Problema: Achar um novo modelo a partir do existente que maximize (localmente) a probabilidade de se obter uma sequência observada

$$P(O|\bar{\lambda}) > P(O|\lambda)$$

HMM - Aprendizagem

- Mais uma variável:

$$\gamma_t(i) = P(X_t = i | O, \lambda) = \alpha_t(i)\beta_t(i) / P(O | \lambda)$$

HMM - Aprendizagem

- Mais uma variável:

$$\gamma_t(i) = P(X_t = i | O, \lambda) = \alpha_t(i)\beta_t(i) / P(O | \lambda)$$

- Mas o que ela representa?

$$\sum_{t=1}^{T-1} \gamma_t(i)$$

HMM - Aprendizagem

- Mais outra variável:

$$\begin{aligned}\xi_t(i, j) &= P(X_t = i, X_{t+1} = j | O, \lambda) \\ &= \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) / P(O | \lambda)\end{aligned}$$

HMM - Aprendizagem

- Mais outra variável:

$$\begin{aligned}\xi_t(i, j) &= P(X_t = i, X_{t+1} = j | O, \lambda) \\ &= \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) / P(O | \lambda)\end{aligned}$$

- Mas o que ela representa?

$$\sum_{t=1}^{T-1} \xi_t(i, j)$$

HMM - Aprendizagem

- Solução:

$$\overline{\pi}_i = \gamma_1(i)$$

$$\overline{a}_{ij} = \sum_{t=1}^{T-1} \xi_t(i, j) / \sum_{t=1}^{T-1} \gamma_t(i)$$

$$\overline{b}_j(k) = \sum_{t=1, O_t=k}^T \gamma_t(j) / \sum_{t=1}^T \gamma_t(j)$$

HMM - Aprendizagem

- Solução:

$$\overline{\pi}_i = \gamma_1(i)$$

$$\overline{a}_{ij} = \sum_{t=1}^{T-1} \xi_t(i, j) / \sum_{t=1}^{T-1} \gamma_t(i)$$

$$\overline{b}_j(k) = \sum_{t=1, O_t=k}^T \gamma_t(j) / \sum_{t=1}^T \gamma_t(j)$$

- Complexidade por iteração: $O(N^2T)$

HMM

- Problemas resuolvidos

HMM

- ~~Problemas resuolvidos~~
- Underflow!

HMM

- ~~Problemas resuolvidos~~
- Underflow!
 - Escalar probabilidades
 - Usar valor do logaritmo

HMM

- ~~Problemas resolvidos~~
- Underflow!
 - Escalar probabilidades
 - Usar valor do logaritmo
- Treinamento adequado exige múltiplas observações!

Considerações Finais

Testes Realizados

- 4 pessoas
- 4 gestos
- 160 execuções de gesto
 - 80 para treinamento
 - 80 para reconhecimento

Testes Realizados

- Diferentes módulos
 - Inicialização do quantizador
 - Extratores
 - Níveis de filtragem
 - HMM
 - Topologia
 - Número de estados
 - Número de símbolos

Conclusão

- Variação da porcentagem de acerto
- Necessidade de grande conjunto de teste
- Escolha dos melhores parâmetros

Referências

- 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.

Perguntas

