

# Redes Neurais Artificiais aplicadas no Reconhecimento de Gestos usando o Kinect

Matheus Lin Truglio Alvarenga, Diogo S. Ortiz Correa, Fernando Santos Osório

USP – Univ. de São Paulo - ICMC – Instit. de Ciências Matemáticas e de Computação  
SSC – Depto. de Sistemas de Computação - Av. do Trabalhador são-carlense, 400  
Cx. Postal 668 - São Carlos-SP - CEP 13560-970 - Brasil

matheus.lin@gmail.com, diogosoc@icmc.usp.br, fosorio@icmc.usp.br

***Resumo.** Este artigo tem por objetivo descrever o projeto, desenvolvimento e os resultados obtidos junto a um sistema para o aprendizado e classificação de gestos através do uso do sensor Kinect (sensor usado no videogame XBox da Microsoft). Foi desenvolvida uma aplicação para testar o reconhecimento de gestos simples, realizados livremente com a mão do usuário, permitindo assim o treinamento, e o posterior reconhecimento e diferenciação de um conjunto de gestos diferentes. O sensor Kinect permite a captura de uma imagem juntamente com o mapa de profundidade, ou seja, obtemos para cada pixel a informação RGB-D (cor RGB e "Depth"). A partir de uma seqüência de quadros com esta informação é possível caracterizar um determinado movimento, que define um gesto em particular. Este trabalho permitiu demonstrar através de uma aplicação prática baseada em uma Rede Neural, que executa em tempo real, que é possível usar um Kinect para tarefas que vão além do simples acompanhamento (tracking) de elementos de uma cena, como é usual em jogos. Através do sistema proposto é possível treinar a Rede Neural para aprender a reconhecer novos e diferentes gestos específicos. Os resultados obtidos foram muito promissores, com uma taxa de reconhecimento bastante alta, o que permite a aplicação deste tipo de técnica em jogos e em interfaces humano-computador (IHC) que possam requerer uma interação mais natural e comandos à distância.*

## 1. Introdução

Os seres humanos se comunicam através de diferentes formas e com o uso de diferentes instrumentos, como por exemplo, através da fala, da escrita, e por meio de gestos. Atualmente as interfaces modernas de comunicação baseadas na Tecnologia da Informação (TI) privilegiam o uso da escrita (teclado) e da entrada de dados através de botões e do uso de dispositivos de apontamento (mouse). Por outro lado, interfaces gestuais vem ganhando cada vez mais importância no contexto das aplicações de TI, onde novos dispositivos como *Tablets*, com interfaces do tipo "touch", vem redescobrimo a importância do reconhecimento de comandos através de gestos específicos. A área de Jogos Digitais também sofreu uma grande revolução recentemente com a introdução de jogos com interfaces gestuais, a exemplo do Wii, Xbox 360 e PS3 [De Paula, 2011; Pfutzenreuter & Tavares, 2008].

As “interfaces naturais com o usuário” (NUI – *Natural User Interfaces*) e baseadas em gestos são uma proposta mais atual e visam uma melhor interação com os sistemas computacionais [Buxton 2010; Buxton et al., 2011]. Existem diversas aplicações para este tipo de interfaces, sendo as mais usadas atualmente, junto às aplicações de entretenimento digital (jogos). Outra importante aplicação deste tipo de interfaces de reconhecimento de gestos é o seu uso junto a pessoas deficientes para o reconhecimento da linguagem de sinais, como a ASL (*American Sign Language*) ou LIBRAS (Linguagem Brasileira de Sinais) [Starner, 1998; Rung-Huei & Ming, 1998; Peres et al., 2006].

Com o surgimento de tecnologias como a do sensor Kinect [Microsoft 2011; Prime Sense 2011], utilizado no videogame Xbox 360 da Microsoft, o desenvolvimento de aplicações para a captura de dados 3D tornou-se mais acessível, devido ao relativo baixo custo deste dispositivo. O Kinect permite o rastreamento da posição de uma pessoa e da movimentação de seu corpo, braços, pernas, mãos e cabeça. Este dispositivo torna possível o desenvolvimento de uma nova geração de “interfaces naturais”, permitindo assim uma melhor interação com as pessoas, bem como permite o reconhecimento de gestos de forma mais precisa do que através do uso de sistemas baseados unicamente em uma câmera (PS3) ou mesmo através do uso de acelerômetros (Wiimote).

Este trabalho busca desenvolver um sistema de aprendizado e reconhecimento de gestos através do uso de Redes Neurais Artificiais. Esta proposta é original e inovadora, pois ao contrário da maioria dos sistemas atuais que fazem apenas o reconhecimento de uma pose (posição estática), ou o rastreamento da posição de determinados elementos da cena, realiza também o reconhecimento do movimento como um todo (seqüência de posições), em tempo real. Neste trabalho foi desenvolvido um sistema inteligente capaz de aprender novos gestos, e assim, é possível treinar o sistema para reconhecer diferentes tipos de gestos. O trabalho está dividido da seguinte maneira: a seção 2 apresenta conceitos sobre reconhecimento de gestos e sobre as tecnologias adotadas neste trabalho (Kinect e Redes Neurais), a seção 3 apresenta o sistema desenvolvido e seus módulos, e a seção 4 apresenta uma análise dos principais resultados, seguida das conclusões finais e perspectivas deste trabalho.

## **2. Reconhecimento de Gestos**

O reconhecimento de poses 2D [Souza Jr. et al., 2007] e 3D [Zinbin & Chung, 2010], assim como o reconhecimento de gestos em 2D [Buckland, 2002; Heinen & Osório, 2006] e 3D também, tem sido buscado por diversos pesquisadores de áreas como IHC (Interface Homem-Máquina), Realidade Virtual, Realidade Aumentada, e Jogos Digitais. Entre as técnicas mais usadas destacam-se as técnicas de Inteligência Artificial, baseadas em aprendizado de máquina (como no caso das Redes Neurais Artificiais - RNAs) e no uso de HMMs (*Hidden Markov Models*).

Neste trabalho, considerando a experiência prévia dos autores no uso de Redes Neurais Artificiais, optou-se por desenvolver um sistema baseado no uso deste tipo de técnica para o reconhecimento da seqüência de movimentos de um gesto. Além disto, foi adotado o dispositivo sensor Kinect, um dispositivo de baixo custo e alta capacidade de aquisição de dados 3D, para a captura dos dados que representam a ação gestual.

Este trabalho teve como inspiração trabalhos prévios de reconhecimento de gestos em 2D usando Redes Neurais Artificiais (RNA). Um primeiro trabalho, bastante didático, foi o trabalho desenvolvido por Buckland (2002), onde foi feito o uso de um dispositivo do tipo mouse para o reconhecimento de gestos 2D (“desenhos” e movimentos feitos com o auxílio deste dispositivo). Neste trabalho foi usada uma RNA para o aprendizado de formas (diagonal, quadrado, marca ‘Z’, etc) que eram desenhadas com o auxílio do mouse, definindo assim uma seqüência de pontos 2D a ser reconhecida. A seqüência de pontos pode ser aprendida, e posteriormente reconhecida, sendo que é reconhecido o padrão temporal do gesto realizado, ou seja, o gesto deve ser repetido de uma forma aproximada ao padrão que foi usado para treinar a RNA (na mesma ordem e sentido dos movimentos originais). Esta aplicação é bastante simples, mas demonstra bem a capacidade de generalização de uma Rede Neural, capaz de reconhecer gestos simples e até mesmo uma ampla gama de caracteres manuscritos fornecidos através de uma superfície sensível ao toque de um *Tablet Computer* ou PDA.

Um segundo trabalho, também importante no contexto dos trabalhos de base para a pesquisa que foi desenvolvida a seguir, foi o desenvolvimento de um reconhecedor *on-line* de assinaturas [Heinen & Osório, 2006]. Este trabalho permitiu demonstrar a capacidade de uma RNA em aprender padrões complexos de movimentos, que caracterizam uma assinatura capturada em tempo real a partir de uma pequena mesa digitalizadora. Neste trabalho foi possível alcançar altas taxas de reconhecimento da ordem de 96% a 100% de acertos na base de validação do aprendizado (*cross-validation test*) [Heinen & Osório, 2006], onde percebe-se todo o potencial do uso de RNAs no aprendizado de seqüências de dados que representam padrões 2D (padrão representando o gesto de realizar uma assinatura).

A partir de trabalhos como os citados acima, iniciou-se um novo projeto que visa aplicar Redes Neurais Artificiais no reconhecimento de padrões de gestos, porém passando do reconhecimento de gestos 2D para o reconhecimento de gestos 3D. Para que fosse possível realizar o reconhecimento de gestos 3D era necessário o uso de um dispositivo de aquisição de posturas 3D, ou seja, era necessário poder identificar a posição e movimentação do braço e da mão do usuário no espaço tridimensional. O dispositivo adotado para o desenvolvimento deste projeto foi o Kinect [De Paula 2011].

## 2.1. Kinect

O Kinect (Figura 1) é um dispositivo desenvolvido principalmente para uso em jogos eletrônicos do console Xbox 360, da Microsoft, lançado em Novembro de 2010. Atualmente, mantém o recorde mundial de dispositivo eletrônico mais rapidamente vendido, alcançando a marca de 8 milhões de vendas em 60 dias (de acordo com o *Guinness Book* 2011). O Kinect é dotado de uma câmera RGB, um sensor de profundidade que usa infra-vermelho e é capaz de mapear o ambiente em 3 dimensões, microfones, um motor para alterar seu ângulo de visão, e interface USB 2.0 para conexão com o videogame (Xbox) ou mesmo com um PC comum (Windows ou Linux). Outras áreas, como a Robótica, vêm também fazendo um extensivo uso deste dispositivo, devido a sua capacidade de percepção espacial. Em Junho de 2011, a Microsoft lançou um SDK não comercial para Windows, e outra versão comercial mais tarde [Microsoft, 2011]. Entretanto, muitas das aplicações atuais usam *drivers* e SDKs fornecidos pela empresa Prime Sense [Prime Sense 2011] que desenvolveu originalmente este dispositivo, e também *drivers* de código abertos (*opensource*).



(a) [De Paula 2011]



(b) Lab. LRM/ICMC-USP

**Figura 1. Kinect: (a) Sensor e seus Componentes – (b) Aplicação em Robótica**

O Kinect é um dispositivo RGB-D, ou seja, permite a captura de imagens representando as cores (RGB) e a profundidade de uma cena ( $D - Depth$ ). Portanto, uma cena capturada pelo Kinect é usualmente representada por um par de imagens de resolução  $640 \times 480$ , com uma imagem em formato colorido RGB (24 bits/pixel) e a outra imagem representando a profundidade de cada pixel ( $Depth$ ). A profundidade representa a distância dos pixels em relação ao sensor, formando uma mapa  $L \times C \times P$  (Linha x Coluna x Profundidade). A Figura 2 apresenta um exemplo de imagem capturada pelo Kinect, com a imagem colorida Fig.1(b) e a imagem do mapa de profundidade Fig.1(c,d). Na Figura 1(c) são aplicadas falsas cores, com as cores frias (tendendo ao azul) representando os pixels mais distantes, e as cores quentes (tendendo ao vermelho) representando os pixels mais próximos do sensor. Do mesmo modo, na Figura 1(d) as cores claras são pixels mais distantes e as cores mais escuras são pixels mais próximos. As imagens RGB-D também podem ser representadas por uma nuvem de pontos com coordenadas espaciais  $X, Y, Z$ , onde cada ponto possui a cor RGB.



**Figura 2. Sensor RGB-D do Kinect: (a) Imagem em Escala de Cinza; (b) Imagem RGB colorida; (c) Mapa de Profundidade com Falsas Cores; (d) Mapa de profundidade em Tons de Cinza (Lab. LRM/ICMC-USP)**

O sensor de profundidade 3D, foi desenvolvido pela empresa Prime Sense (2011), que juntou-se a outras duas empresas para desenvolver o projeto *OpenNI Kinect* [Rusu et al., 2011], como um pacote para o ROS [Willow Garage, 2011], disponibilizando *drivers* e ferramentas *open source* para seus sensores, compatível com os sensores presentes no *Kinect*. A primeira versão do *OpenNI Kinect* foi publicada em Março de 2011, para a versão *DiamondBack* do ROS (*Robotic Operating System*<sup>1</sup>). Em Julho de 2011, foi lançado o *OpenNI Tracker*, capaz de reconhecer um humano e acompanhar seus movimentos, estimando-se a posição de algumas juntas do corpo (como pescoço, ombros, cotovelos e mãos). Esse módulo é amplamente utilizado em inúmeras outras aplicações, pois é uma forma rápida e eficiente de rastrear o usuário. A Figura 3 mostra uma captura de tela da aplicação de rastrear humanos. O Kinect permite capturar 30 fps (quadros por segundo) de imagens RGB-D como as apresentadas nas Figuras 2 e 3.

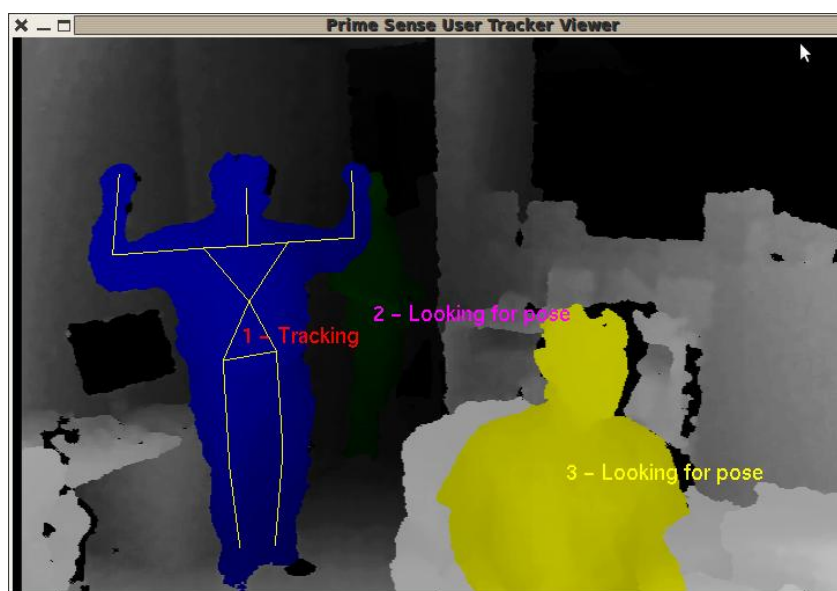


Figura 3. OpenNI Tracker: Detector de Humanos (Fonte: Rusu et al., 2011)

## 2.2. Redes Neurais Artificiais

Redes Neurais Artificiais são modelos matemáticos que se assemelham e/ou se inspiram nas estruturas neurais biológicas e que têm capacidade computacional adquirida por meio de adaptação, apresentando propriedades de aprendizado e generalização [Haykin, 1998]. Uma Rede Neural Artificial (RNA) é um sistema inspirado no conceito do neurônio biológico, onde uma rede é formada por células desse tipo interconectadas entre si. Os neurônios biológicos são células do sistema nervoso, capazes de receber, processar e enviar sinais elétricos, e, por outro lado, os neurônios artificiais são representações numéricas destas células que computam funções matemáticas, buscando de certo modo reproduzir os potenciais elétricos das células e a sua propagação, através do uso de valores numéricos, de funções de soma, de ponderação e de transferência de valores destes potenciais simulados.

---

<sup>1</sup> ROS – *Robotic Operating System* (Willow Garage): É uma coleção de *softwares* com *drivers*, *interfaces*, pacotes e aplicações para interação e operação de dispositivos robóticos.

Existem diferentes modelos de RNAs propostas na literatura, no entanto, um dos modelos mais conhecidos e adotados é o das redes de neurônios do tipo *Perceptron*, agrupados formando redes de múltiplas camadas denominadas redes do tipo MLP (*Multi-Layer Perceptron Networks*) [Haykin, 1998]. As Redes Neurais do tipo MLP foram adotadas neste trabalho, sendo adotadas redes de 3 camadas com propagação do tipo *feed-forward* e aprendizado através do uso do Algoritmo Supervisionado do tipo *Back-Propagation* [Haykin, 1998]. Este tipo de rede é basicamente o mesmo modelo adotado nos trabalhos relacionados apresentados como referência e inspiração para este trabalho.

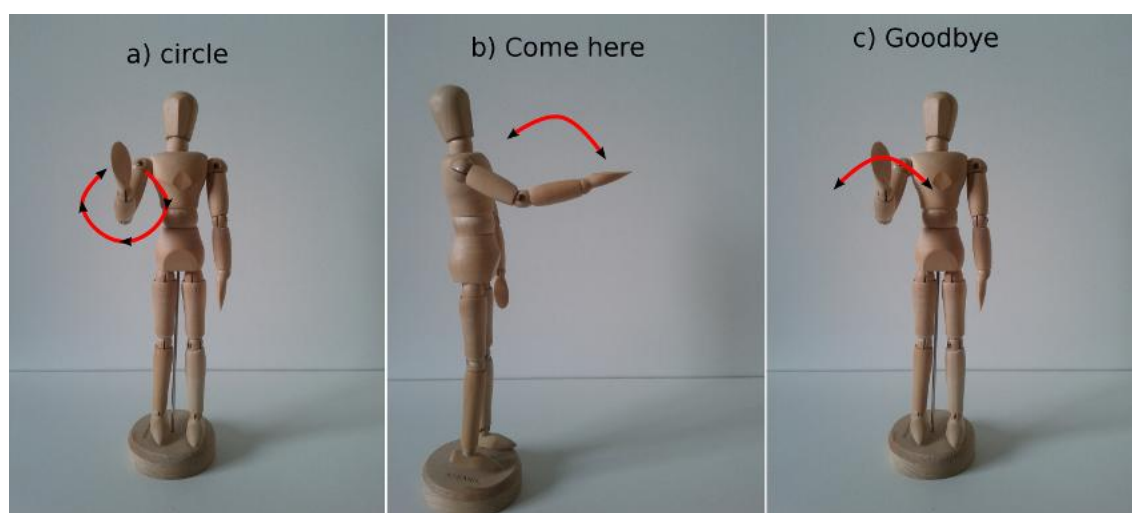
O aprendizado supervisionado nas redes MLP com o uso do Algoritmo *Back-Propagation*, consiste basicamente em se criar uma base de dados de aprendizado, com exemplos das entradas apresentadas para a rede e das saídas desejadas a serem obtidas. A rede propaga os sinais de entrada através dos neurônios e camadas, realizando uma soma ponderada das entradas em função de pesos que definem a “importância” de cada entrada. Depois, são aplicadas funções de transferências e os dados de entrada são propagados até a saída da rede neural, obtendo assim sua resposta. O algoritmo de aprendizado irá otimizar e adaptar estes pesos que são denominados de “pesos sinápticos”, e através da adaptação do valor destes, é que fazemos com que a rede vá ajustando suas respostas de modo a responder da melhor maneira possível as entradas fornecidas (aproximando de modo iterativo as respostas obtidas na saída da rede dos valores da saída desejada indicada no arquivo de treinamento). O processo de aprendizado resulta usualmente em uma melhoria das respostas da rede, até que se alcance um determinado critério de parada. O critério mais simples é a parada quando a rede “acerta” todas as respostas para toda a base de aprendizado.

Apesar de parecer simples, o algoritmo de Aprendizado Neural envolve outras questões como [Da Silva et al., 2010]: o bloqueio do aprendizado em mínimos locais (e o uso do *Momentum* para tentar evitar alguns mínimos locais); a dependência da inicialização dos pesos da rede; o uso da validação cruzada (*Cross-validation*) buscando não apenas decorar os dados da base de aprendizado, mas também obter a resposta correta para exemplos diferentes dos aprendidos (base de teste/validação); a definição do critério de parada baseado no erro e/ou acerto na base de aprendizado ou de validação (p.ex. adoção do critério de *Best-epoch* ou *Early Stopping*, evitando o *OverTraining*); o tipo de limiar usado na binarização da saída da rede (no caso de redes usadas para a classificação de dados) e o critério adotado para a contagem de acertos e erros (p.ex. *winner-take-all*). Neste trabalho foi adotado o treino com uso de *Momentum*, *Multiple Runs* (inicialização), *Cross-Validation* (Base de dados de treinamento e de validação), *Early Stopping* (Critério de parada), e *Winner-Takes-All* (Contagem de Acertos/Erros).

Para poder implementar o sistema de reconhecimento de gestos foi necessário definir os parâmetros que seriam usados como entrada da Rede Neural (entradas da rede), os tipos de gestos a serem reconhecidos (saídas da rede), assim como a estrutura interna da rede (camada intermediária). A seção 3 irá descrever a seguir os detalhes de funcionamento do sistema de reconhecimento de gestos que foi desenvolvido.

### 3. Sistema de Reconhecimento de Gestos

O objetivo deste trabalho foi o de desenvolver um sistema capaz de reconhecer gestos, através da descrição de uma seqüência de movimentos das posições do braço e da mão no espaço tridimensional. O sistema foi desenvolvido para realizar o acompanhamento da mão direita do usuário, para que fosse possível assim fornecer comandos a computadores, aplicações, jogos, ou até mesmo comandar robôs à distância através de uma interface natural (NUI). A fim de validar a proposta do sistema de reconhecimento de gestos, foram definidos inicialmente 3 gestos para serem identificados, conforme mostra a Figura 4.



**Figura 4. Movimentos a serem aprendidos pela RNA e classificados automaticamente: a) circle, b) come here, e c) goodbye**

Inicialmente foi instalado o ROS, o pacote *OpenNI Tracker* e o aplicativo *MIT hand\_interaction*, sendo estudadas as informações providas pelo nó *hand\_interaction*. Os estudos realizados permitiram obter as informações descrevendo a posição 3D da mão direita do esqueleto, onde foram separadas as mensagens relativas a este elemento. A pose da mão direita do esqueleto extraído através do *Tracker* permitiu analisar deslocamentos da mão e assim criar a rede neural com as entradas e saídas adequadas para classificar os movimentos. Foi possível assim gravar as amostras para criar conjuntos de treinamento para a rede, e depois obter os dados que são passados para o teste em tempo real da rede neural.

O aplicativo de demonstração do *MIT hand\_interaction* disponibiliza um conjunto de programas que são lançados em conjunto, capazes de identificar e rastrear um usuário (humano), estimando a pose de seu esqueleto (utilizando posições estimadas das juntas do usuário, como ombros, cotovelos e mãos). Opcionalmente, é possível utilizar o visualizador *RViz* (outro pacote padrão do ROS) para visualizar o esqueleto detectado e destacar as mãos do usuário.

Um dos nós criados dessa forma é o *skeletons*, que envia mensagens contendo um vetor com todos os esqueletos identificados em cada leitura de dados do Kinect. Desses esqueletos, seleciona-se somente o primeiro (posição 0), por simplificação, pois supõe-se que apenas um usuário por vez utilizará o software. Outros usuários podem entrar em cena, também, mas suas mensagens não serão consideradas. Cada esqueleto

contém um conjunto posições tridimensionais, indicando a posição referente a cada junta do usuário. Uma dessas juntas é a mão direita (*right hand*), que é o alvo da identificação deste projeto.

Analisando-se o conteúdo da mensagem de um esqueleto (comando `rosmmsg show [pacote/tipo de mensagem]`), obtém-se a seguinte árvore do código abaixo (as reticências referem-se a segmentos da mensagem sem relevância para este projeto):

```
rosmmsg show body_msgs / Skeletons
...
body_msgs / Skeleton[] skeletons
...
body_msgs / SkeletonJoint right_hand
geometry_msgs / Point position
float64 x
float64 y
float64 z
float32 confidence
...
```

Dessa forma, para extrair a posição (x, y, z) da mão direita do primeiro esqueleto da mensagem (do tipo *body\_msgs/Skeletons*) deve-se requisitar `data.skeletons[0].right_hand.position` e a essa linha, adicionar `.x`, `.y` ou `.z`, sendo `data` a mensagem recebida pela função de callback. Para analisar os deslocamentos da mão direita do usuário, um programa supervisor foi desenvolvido, em linguagem python. Este programa analisa a posição da mão direita, armazenando pontos e deslocamentos em listas separadas.

A lista de pontos armazena a posição atual, caso a lista esteja vazia, ou, caso a distância do ponto atual e do último ponto adicionado seja maior que 5 cm. Os pontos são adicionados ao início da lista. Se já houver pelo menos um ponto na lista quando outro ponto for adicionado, calcula-se os "deslocamentos significativos" para cada eixo: se houve um deslocamento maior que 5 cm no sentido positivo de X, o "deslocamento X" é codificado como sendo "+1"; se for no sentido negativo, é representado por "-1"; se o deslocamento no eixo X for menor que 5cm, o "deslocamento X" é "0". Realiza-se essas atribuições também para os eixos Y e Z. Esses três deslocamentos são adicionados ao início da lista de deslocamentos. Quando a lista de deslocamentos atinge um tamanho de 20 (selecionado arbitrariamente), uma mensagem é enviada ao tipo Trace no tópico "*right hand trace*", contendo esses 20 deslocamentos. Se a lista ultrapassar a quantidade de 20, descarta-se os pontos mais antigos, até retornar ao máximo de 20 permitidos. Se forem realizadas 50 leituras de posição sem que nenhum deslocamento significativo ocorra, as listas são então esvaziadas. Esse recurso foi desenvolvido para "cancelar" um movimento sem precisar interagir com o computador.

Note que o método acima permite detectar o início de movimento (deslocamento significativo), gerando sempre uma coleção de 20 descritores de movimento (em X, Y e Z) a serem usado no reconhecimento dos gestos. Além disto, os descritores de movimento descrevem movimentos relativos da mão, o que é importante em termos do aprendizado para que se obtenha uma independência de posição inicial e translação da origem da mão. Com isto, obtém-se um conjunto de 60 dados de entrada a serem fornecidos para a rede (20 deslocamentos x 3 eixos), definindo assim uma rede com 60 neurônios de entrada.



Uma vez que inicialmente serão reconhecidos apenas 3 gestos, a rede neural deverá possuir apenas 3 neurônios na camada de saída. As saídas foram codificadas da seguinte forma: [1, 0, 0] - *Circle*: o usuário descreve um círculo no sentido anti-horário para si mesmo (ou horário para o Kinect); [0, 1, 0] - *Come here*: com o braço esticado para frente e palma da mão para cima, o usuário dobra o cotovelo, trazendo sua mão para perto do corpo, e esticando o braço novamente; [0, 0, 1] - *Goodbye*: o usuário acena, deslocando a mão alternadamente para a esquerda e para a direita.

Após a gravação de aproximadamente 300 amostras de cada classe de movimento (900 exemplos no total), foram criados 3 conjuntos de amostras: o de treinamento (80% das amostras – 720 exemplos), o de validação cruzada (10% - 90 exemplos) e o de teste real final (10% - 90 exemplos).

#### **4. Resultados**

Inicialmente foram testadas diferentes configurações de Redes Neurais, adotando a topologia com 60 entradas, 3 saídas binárias, e testando com 5, 10, 20 e 30 neurônios na camada intermediária. Também foram feitos diversos teste variando a taxa de aprendizado e o momentum. A partir destes testes preliminares, definiu-se como melhor modelo de RNA a seguinte configuração: 60 neurônios na camada de entrada; 20 neurônios na camada intermediária; 3 neurônios na camada de saída; taxa de aprendizado de 0,1 (*alfa* ou *learning rate*); momentum de 0,5; aprendizado com validação cruzada.

Considerando a base de dados de validação, a rede obteve um acerto de 83 exemplos dos 90 exemplos da base real apresentada, errando apenas 7 dos exemplos. É importante destacar que o erro atual implica apenas em que o usuário execute novamente o gesto, ou repita por um pouco mais de tempo, pois a RNA irá acabar acertando a classificação. O principal resultado deste trabalho pode ser observado no vídeo que foi disponibilizado na Internet e que apresenta o sistema sendo utilizado em tempo real. O vídeo está disponível em: <http://osorio.wait4.org/Kinect-Gestos/>

#### **5. Conclusões**

Este trabalho demonstrou a viabilidade de realizar o aprendizado e reconhecimento de gestos em tempo real através do uso de um Kinect e de Redes Neurais Artificiais. Os resultados foram bastante promissores, porém ainda é possível aprimorar o sistema a fim de obter melhores resultados de acerto no reconhecimento dos gestos.

Como trabalhos futuros espera-se poder incluir novos parâmetros de análise do movimento da mão, a exemplo do que foi feito em um trabalho anterior para o reconhecimento de assinaturas 2D [Heinen & Osório, 2006], porém neste caso, adaptado para o problema de reconhecimento de movimentos em 3D. Além disto, serão realizados novos testes com a captura de um conjunto maior e mais completo de gestos, a fim de melhor validar o sistema proposto.

#### **Referências Bibliográficas**

Buckland, M. *AI Techniques for Game Programming*. The Premier press game development series - Premier Press: Portland, USA. 2002. ISBN 9781931841085.

- Buxton, Bill. *NUI - What's in a Name?* TechTalk: Microsoft Development Center Copenhagen, Denmark. Oct. 4, 2010. URL: <http://channel9.msdn.com/posts/TechTalk-NUI-Whats-in-a-Name> (ultimo acesso: 10/12/2011).
- Buxton, Bill; Billinghamurst, Mark; Guiard, Yves; Sellen, Abigail; Zhai, Shumin. *Human Input to Computer Systems: Theories, Techniques and Technology*. Cambridge: Cambridge University Press (2011, in preparation). URL: <http://billbuxton.com/inputManuscript.html> (ultimo acesso: 10/12/2011).
- Da Silva, Ivan Nunes; Spatti, Danilo Hernan; Flauzino, Rogério Andrade. *Redes Neurais Artificiais para Engenharia e Ciências Aplicadas*. 399 páginas - 1ª edição – ArtLiber Editora 2010. ISBN: 8588098534
- De Paula, Bruno Campagnolo. *Adaptando e desenvolvendo jogos para uso com o Microsoft Kinect*. Proceedings of SBGames 2011 - Tutorials Track: Computing. SBC, Salvador. 2011.
- Haykin, S. *Neural Networks: A Comprehensive Foundation*. 2nd. ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998. ISBN 0132733501.
- Heinen, Milton Roberto; Osório, Fernando Santos. *Handwritten Signature Authentication using Artificial Neural Networks*. International Joint Conference on Neural Networks - IJCNN'06, 2006. pp. 5012-5019.
- Microsoft Corporation. *Kinect for Windows*. 2011. URL: <http://kinectforwindows.org/> (ultimo acesso: 10/12/2011)
- Peres, Sarajane Marques; Flores, Franklin Cesar; Veronez, Denise; Olguin, Carlos Jose Maria. *LIBRAS Signals Recognition: a study with Learning Vector Quantization and Bit Signature*. Ninth Brazilian Symposium on Neural Networks, 2006. SBRN '06. pp. 119 – 124. DOI: 10.1109/SBRN.2006.26
- Pfutzenreuter, Edson P.; Tavares, Roger. *Dedões - Corpo – Gestos: A inteligência corporal no design dos jogos digitais*. Proceedings of SBGames 2008 – Game and Culture Track. SBC, Belo Horizonte, 2008. pp.19-23.
- Prime Sense. *Prime Sense Natural Interaction*. 2011. URL: <http://www.primesense.com/> (ultimo acesso: 10/12/2011).
- Rung-Huei Liang; Ming Ouhyoung. *A real-time continuous gesture recognition system for sign language*. Proceedings. Third IEEE International Conf. on Automatic Face and Gesture Recognition, 1998. pp. 558–567. DOI: 10.1109/AFGR.1998.671007
- Rusu, R. B.; Conley, K.; Foote, T. *OpenNI Kinect*. 2011. URL: [http://www.ros.org/wiki/openni\\_kinect](http://www.ros.org/wiki/openni_kinect) (ultimo acesso: 10/12/2011).
- Souza Jr., Nelson; Kehl, Thiago; Jung, Cláudio; Osório, Fernando S.; Musse, Soraia R. *Animando Humanos Virtuais em Tempo-Real usando Visão Computacional e Redes Neurais*. In: Proceedings of SVR - Symposium on Virtual and Augmented Reality, 2007. Petrópolis, RJ : SBC – LNCC. pp. 180-186.
- Starnes, T.; Weaver, J.; Pentland, A. *Real-time American sign language recognition using desk and wearable computer based video*. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Dec. 1998 Vol.20,N.12. pp. 1371 – 1375. DOI: 10.1109/34.735811
- Willow Garage. ROS - Robot Operating System. 2011. URL: <http://www.ros.org/> (ultimo acesso: 10/12/2011).
- Zibin Wang; Chung, R. *Articulated Human Body: 3D Pose Estimation Using a Single Camera*. 20th International Conference on Pattern Recognition (ICPR), 2010. pp. 3768-3771.