



Universidade Federal de Pernambuco
Centro de Informática

Graduação em Engenharia da Computação

Reconhecimento de Coleta com a Mão (*Grasp*) para Aprendizagem por Demonstração

Paulo Henrique Muniz Ferreira

Trabalho de Graduação

Recife
13 de dezembro de 2011

Universidade Federal de Pernambuco
Centro de Informática

Reconhecimento de Coleta com a Mão (*Grasp*) para Aprendizagem por Demonstração

Paulo Henrique Muniz Ferreira
phmf@cin.ufpe.br

*Trabalho de Graduação apresentado no Centro
de Informática (CIn) da Universidade Federal
de Pernambuco (UFPE) como requisito parcial
para obtenção do grau de Engenheiro da
Computação.*

Orientador: Aluísio Fausto Ribeiro Araújo

Recife
13 de dezembro de 2011

Assinaturas

Aluizio Fausto Ribeiro de Araújo
Orientador

Paulo Henrique Muniz Ferreira
Aluno

Dedico este trabalho à minha família, José Henrique, Socorro e Ágata Kellen, que me forneceu todo o apoio necessário para o meu crescimento em todos os aspectos, ao meu orientador, Aluizio, por despertar em mim o interesse na pesquisa científica e na carreira acadêmica e a todos os meus amigos por me motivar todos os dias.

Agradecimentos

Gostaria de agradecer primeiramente a Deus, acima de tudo, Àquele que me concedeu saúde necessária e oportunidade de chegar nessa última fase do curso. Acreditar Nele fez com que eu tivesse estímulo para driblar os desafios, por vezes, tão recorrentes.

Agradeço imensamente a meus pais, José Henrique Gomes Ferreira e Maria do Socorro Muniz Ferreira, por terem me apoiado e me dado, dia após dia, as condições de chegar até aqui. Sem o esforço deles, de forma alguma eu teria conseguido realizar todas as minhas conquistas. Além deles, meus sinceros agradecimentos a minha irmã, Ágata Kellen Muniz Ferreira, com quem sempre pude contar em todos os momentos, tanto nos de felicidade quanto naqueles mais difíceis.

Aos meus amigos, tanto aqueles da faculdade que construíram junto a mim os conhecimentos que aprendi; quanto àqueles que, ainda que não tenhamos um contato diário, sinto com plena convicção, estão sempre dispostos a dividir comigo minhas conquistas e também os meus problemas.

Por fim, mas com a mesma importância, agradeço ao meu orientador, Aluizio Fausto Ribeiro Araújo, pela paciência e confiança que em mim depositou e, conseqüentemente, possibilitou a conclusão deste trabalho.

Obrigado

“Virtus in medio.”

Resumo

A mão humana tem a capacidade de realizar várias tarefas do cotidiano usando a combinação de sensores, controladores e atuadores biológicos. Há três funções em relação ao pulso e à palma da mão que em particular, são mais importantes: explorar, agarrar (*grasp*), e manipular objetos. Pesquisas atuais em robótica envolvendo *grasp* tenta compreender e desenvolver controles que imitar as duas últimas funções.

Mãos robóticas estão cada vez mais recebendo atenção nos últimos anos e um esforço considerável tem sido despendido para o desenvolvimento de controladores para as mãos robóticas com múltiplos dedos. Frequentemente, os trabalhos dentro dessa área empregam a abordagem de programação por demonstração cuja ideia básica consiste em desenvolver um sistema robótico que aprenda tarefas de segurar objetos (*grasp*) através das demonstrações de um operador humano.

Nesse trabalho, demonstramos o estado da arte em reconhecer gestos contínuos de *grasp* de mãos humanas. Apresentamos uma adaptação da rede neural mapas auto-organizáveis (SOM) como uma nova solução para reconhecimento de tipos de *grasp*. Em nossos experimentos, o sistema de reconhecimento obteve taxa de acerto de 80%.

Palavras Chaves: Reconhecimento de *Grasp*, Programação por Demonstração, Mapas Auto-Organizáveis.

Abstract

The human hand has the ability to perform various daily tasks using a combination of biological sensors, controllers, and actuators. There are three main functions considering the movement of the wrist and the palm of the hand: to explore, to grasp, and to manipulate objects. Current research on robot *grasping* aim to handle control strategies to reproduce the last two functions.

Robot hands are increasingly receiving attention in recent years and considerable effort has been expended to develop controllers for robot hands with multiple fingers. In general, the works in this area use the approach called programming by demonstration in which the basic idea is to develop a robotic system capable of learning grasp tasks through demonstrations of a human operator.

In this work, we demonstrate the state of the art in recognizing a temporal sequence of grasping movements of human hands. We present an adaptation of the neural network self-organizing maps (SOM) as a novel solution to recognize different types of *grasp*. In our experiments, the recognition system obtained accuracy rate of 80%.

Keywords: *Grasp* Recognition, Programming by Demonstration, Self-Organizing Maps.

Sumário

Introdução	1
1.1 Importância do Estudo	1
1.2 Objetivos	2
1.2.1 Objetivo Geral	2
1.2.2 Objetivo Específico	2
1.3 Estrutura do Documento	3
Estado da Arte	4
2.1 Um Breve Histórico sobre Mão Robóticas Com Dedos	4
2.2 Trabalhos Relacionados	7
Conceitos Básicos para <i>Grasps</i>	9
3.1 Definição de <i>Grasp</i>	9
3.2 Classificação de <i>Grasps</i>	10
3.2.1 Decrição da Taxonomia	12
3.3 Plataforma de Programação por Demonstração	14
Rede Neural Para <i>Grasps</i>	16
4.1 SOM – Mapas Auto-Organizáveis	16
4.2 Consideração Temporal nos mapas auto-organizáveis	18
4.3 Adaptações da Rede SOM	19
4.3.1 Alteração da Concatenação dos Padrões	20
4.3.2 Alterações no treinamento da rede SOM	20
4.3.3 Rotulação dos Nodos e Classificador	21
Experimentos	23
5.1 Human <i>Grasping</i> Database	23
5.2 Experimento - Posturas Mais Discriminantes	26
5.3 Experimentos - Reconhecimento de Tipos de <i>Grasp</i>	28
5.4 Resultados e Discussão	29
Conclusão e Trabalhos Futuros	32
Referências	34

Lista de Figuras

Figura 1 - A mão hábil de Utah/MIT (1984).....	5
Figura 2 - A Mão hábil Shadow.....	5
Figura 3 - Robô Twendy-One.	6
Figura 4 - Tipos de <i>Grasp</i> Usados.....	11
Figura 5 - Taxonomia abrangente de <i>grasps</i> , que inclui 33 tipos de <i>grasp</i>	13
Figura 6 - Taxonomia abrangente de <i>grasps</i> , que inclui 33 tipos de <i>grasp</i>	13
Figura 7 - - Programação por demonstração em um contexto de <i>grasp</i>	14
Figura 8 - Exemplo de topologia usada para SOM. As unidades de entrada são conectadas a todos os nodos da camada de saída.	17
Figura 9- O mapa auto-organizável treinado com padrões de entrada concatenados. No exemplo, três padrões de entrada são concatenados para forma o vetor de entrada da rede.....	19
Figura 10 - Mão com os 6 sensores.....	24
Figura 11 - Ilustração do tipo de <i>grasp</i> “Power Sphere.”	25
Figura 12- Ilustração da distribuição dos dados.....	26
Figura 13 Comparação visual do tipo “15 - Fixed Hook” e “4 - adducted thumb”	31

Lista de Tabelas

Tabela 1 - Tabela da diferença da média intra-classe e entre-classe.	27
Tabela 2 - Tabela de parâmetros do treinamento da rede.	29
Tabela 3 - Tipos de <i>grasp</i> com taxa de acerto $\geq 75\%$	30
Tabela 4 - Tipos de <i>grasp</i> com taxa de acerto $<75\%$ & $\geq 50\%$	31
Tabela 5 - Tipos de <i>grasp</i> com taxa de acerto $< 50\%$. Fonte: Criação própria.	31

Siglas

CRF: Conditional Random Field

DOF: Degree-of-Freedom (graus de liberdade)

GMM: Gaussian Mixture Models (Modelos de Mistura Gaussiana)

HMM: Hidden Markov Models (Modelos Escondidos de Markov)

KTH: Kungliga Tekniska Höskolan

LMU: Ludwig-Maximilian University

PbD: Programming by Demonstration (programação por demonstração)

SOFM: Self-organizing Feature Maps (mapas auto-organizáveis de característica)

SOM: self-organizing Maps (mapas auto-organizáveis)

SVD: Singular Value Decomposition (decomposição de valor singular)

SVM: Support Vector Machines (Máquina de Vetores de Suporte)

TG: time clustering (agrupamento no tempo)

TG: Trabalho de graduação

TS: Takagi-Sugeno

Capítulo 1

Introdução

Neste capítulo introdutório será discutida a justificativa do presente Trabalho de Graduação no contexto da automação inteligente. Também mostraremos o objetivo geral e as metas específicas desse trabalho e a estrutura deste documento de uma maneira geral.

1.1 Importância do Estudo

A função motora dos seres humanos pode ser considerada como sendo composta de duas habilidades principais: a capacidade de andar em pé e pegar (*grasp*) e manipular objetos pelas mãos. Há várias pesquisas para modelar, propor algoritmos, e realizar implementações (computacionais e/ou em hardware) considerando tais habilidades.

As mãos robóticas foram inicialmente estudadas de forma mais ativa do que os pés do robô devido ao seu emprego maior em tarefas diversas. No entanto, após êxito do desenvolvimento de robôs humanoides caminhantes pela Honda em 1996 (HIRAI et al., 1998) e por alguns outros grupos de pesquisa mostrou que a função motora de andar está mais avançada do que a de agarrar e a de manipular objetos com a mão. Hoje, para realização de robôs humanoides que possam realmente ajudar as pessoas, há um desafio maior no desenvolvimento das mãos robóticas para a realização de vários tipos de coleta com a mão (*grasp*) e tarefas de manipulação do que robôs humanoides caminhantes já que a capacidade de caminhar em robôs já avançou bastante.

Mãos robóticas estão cada vez mais recebendo atenção nos últimos anos

e um esforço considerável tem sido despendido para o desenvolvimento de controladores para as mãos robóticas com múltiplos dedos. A necessidade de tais controladores é explicada por agarrar e manipular uma variedade de objetos por mãos robóticas são funcionalidades fundamentais dos vários sistemas robóticos tais como robôs humanoides, robôs industriais, e vários tipos de robôs móveis, que são úteis em diferentes ambientes, como casa, edifícios públicos, área da cidade, fábrica, no espaço e no mar (YOSHIKAWA, 2010).

Uma das dificuldades desse contexto é a sua programação, devido à alta dimensionalidade inerente do *grasp* e tarefas de manipulação. Uma abordagem para resolver o problema é programar a tarefa da mão robótica diretamente por demonstração de execução de *grasp* e manipulação por humanos. Ou seja, o operador executa uma tarefa, enquanto o robô observa as ações demonstradas, com a ajuda de dispositivos de captura de movimento e câmeras de vídeo. Nesse caso, o robô tem que reconhecer as ações executadas pelo demonstrador e replicá-las por conta própria. Esta abordagem é chamada de programação por demonstração (*Programming by Demonstration* - PbD) e é usada em aplicações robótica complexas, tais como *grasp* e manipulação (PALM, ILIEV e KADMIRY, 2009).

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo desse trabalho de graduação é produzir um sistema que, a partir de dados adquiridos de coletas de objetos por mão humana, categorize e reconheça os diferentes tipos de *grasp*.

1.2.2 Objetivos Específicos

Este TG realizou quatro objetivos específicos relacionados ao contexto de reconhecimento de tipos de *grasp*. Um deles foi elencar uma pesquisa extensa relacionada com o *grasp* humano e através dela obter uma base de dados de *grasp* humano. Outro objetivo foi um estudo direcionado ao contexto de classificação (taxonomia) dos diversos tipos de *grasps*. Também foi feito um estudo de técnicas de reconhecimento de gestos de *grasps*. A última é uma adaptação da rede neural chamada mapas auto-organizáveis de Kohonen (KOHONEN, 1990) utilizada para reconhecer um *grasp*.

1.3 Estrutura do Documento

O restante do trabalho se organiza da seguinte forma: No capítulo 2, temos um breve discussão do estado da arte do desenvolvimento de mãos robóticas, como também, o estudo sobre o reconhecimento de tipos de *grasp*. No capítulo 3, temos os conceitos básicos desse trabalho. Neste capítulo, definimos formalmente *grasp*, dissertamos sobre as classificações de *grasp* utilizadas nesse trabalho e, por último, enunciamos o problema alvo. No capítulo 4, descrevemos as técnicas utilizadas no desenvolvimento do nosso sistema de reconhecimento e, por fim, detalhamos a nossa proposta. No capítulo 5, relatamos a estrutura experimental usada para avaliar nossa proposta, e também, mostramos os resultados obtidos. Será realizada também uma discussão a respeito desses resultados. No capítulo 6, temos uma breve conclusão desse trabalho, no qual relatamos as considerações finais e as propostas de trabalhos futuros.

Capítulo 2

Estado da Arte

Este capítulo busca oferecer uma visão geral do estado da arte das pesquisas relacionadas ao tema desse trabalho de graduação. Em um primeiro momento, trazemos um breve histórico do desenvolvimento de mão robóticas que visa contextualizar a importância do estudo neste TG. Logo após, é feito um apanhado sobre propostas de soluções para o problema de identificação, em especial, reconhecimento de tipos de *grasp*.

2.1 Um Breve Histórico sobre Mão Robóticas Com Dedos

Nesses últimos 45 anos foram desenvolvidos muitos experimentos envolvendo mãos mecânicas com múltiplos dedos. As primeiras mãos robóticas com dedos com várias articulações foram desenvolvidas com controle de computador. Um exemplo é a mão hábil de Utah/MIT (1984) (JACOBSEN et al., 1986) mostrada na Figura 1.

Por volta do ano 2000, várias outras novas mãos foram desenvolvidas. Todas estas novas mãos já não usaram mais cabos de tendões, diferentemente das primeiras mãos desenvolvidas, mas empregavam algumas ligações mecânicas, tais como parafusos de chumbo e engrenagens diferenciais. Os mecanismos de acionamento são integrados na própria mão ou contidos no antebraço. Elas são equipadas com vários sensores, mas seus tamanhos são um pouco maior do que as mãos humanas.



Figura 1 - A mão hábil de Utah/MIT (1984).

Fonte: <<http://www.computerhistory.org/collections/accession/102693567>>

Mais recentemente, mãos menores, que poderiam ser usados para robôs humanoides, foram desenvolvidas, tais como Universal-Mão (HOSHINO e KAWABUCHI, 2005) e Mão hábil Shadow (2007) (SHADOW , 2010) mostrada na Figura 2. Estas mãos têm cinco dedos e tem tamanho de uma mão humana. Elas foram desenvolvidas por empresas privadas. Outra mão, de tamanho semelhante ao humano, é a mão de cinco dedos chamada DLR / HIT Hand II e foi desenvolvida por (LIU, H. et al., 2008).

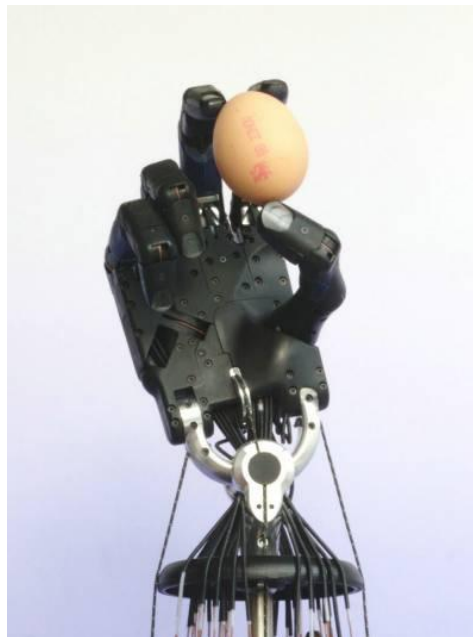


Figura 2 - A Mão hábil Shadow.

Fonte: <<http://www.davidbuckley.net/RS/HandResearch.htm>>.

As mais recentes mãos robóticas desenvolvidas foram projetadas com um propósito de realmente montá-las em robôs humanoides. Uma delas é a mão para HRP3P (KANEKO, HARADA e KANEHIRO, 2007) e a outra é a mão do robô TWENDY-ONE (2009) (IWATA e SUGANO, 2009), Figura 3. Ambas as mãos têm quatro dedos. Uma característica interessante da mão, nestes casos, é que toda a superfície é coberta por pele macia.



Figura 3 – Robô Twendy-One.

Fonte: <<http://www.ubergizmo.com/2010/03/twendy-one-robot/>>.

A partir destes acontecimentos, pode-se dizer que o tamanho e a funcionalidade mecânica das mãos robóticas estão agora muito mais próximos da mão humana. Realizações experimentais de várias tarefas específicas de manipulação, usando estas mãos também têm sido relatadas (YOSHIKAWA, 2010). No entanto, os algoritmos de controle parecem ser bastante orientados à tarefa. Uma organização mais geral do planejamento e controle será necessária para a realização de tipos mais complexos de *grasp* e tarefas de manipulação mais elaboradas. Um passo indispensável para que mãos robóticas atuem como

mãos humanas é que as primeiras repliquem a forma e os movimentos das segundas. Mais ainda, o agente robótico que possui as mãos deve ser capaz, como o ser humano, de reconhecer o tipo de *grasp* para poder realizar tarefas como os humanos. O reconhecimento de diferentes tipos de *grasps*, baseado na movimentação da mão humana é o tema principal deste trabalho de graduação.

2.2 Trabalhos Relacionados

O reconhecimento de *grasp* é ainda um problema em aberto devido à ambiguidade da dinâmica do *grasp* embora venha sendo investigado nessas últimas duas décadas. Diferentes técnicas e configurações de captura de movimento têm sido aplicadas para a abordagem de programação por demonstração (PbD). O estado da arte em reconhecimento de *grasp* pode ser representado por técnicas baseadas nos Modelos de Mistura Gaussiana (*Gaussian Mixture Models* - GMM), Modelos Escondidos de Markov (*Hidden Markov Models* - HMM), Máquina de Vetores de Suporte (*Support Vector Machines* - SVM) e também em lógica *fuzzy* (JU et al., 2008; PALM, 2008; PALM, MEMBER e ILIEV, 2007). Por exemplo, Kang e Ikeuchi (1997) descrevem um sistema que observa, reconhece e mapeia *grasps* humano para um robô manipulador. Neste trabalho, as demonstrações humanas são capturadas por um sistema de visão estéreo e uma luva de dados.

Em outro caso, palm, Iliev e Kadmiry (2009) apresentaram um método baseado em lógica *fuzzy* para modelagem e classificação de *grasp* humanos. Neste trabalho, as trajetórias dos dedos dos *grasps* primitivos são modeladas no tempo e no espaço por agrupamento *fuzzy* e modelagem Takagi-Sugeno (TS). Nesse caso, a captura dos dados é feita a partir de luvas de dados. Calinon e Billard (2007) utilizam modelos de mistura gaussiana (GMM) para o aprendizado do robô de gestos humanos. Liu, H. et al. (2008) propuseram um novo método de agrupamento no tempo (*time clustering* - TC) e métodos adaptados a partir dos modelos de mistura gaussiana (GMM) e modelos escondidos de Markov (*Hidden Markov Models* - HMM). Mostraram, através de seus experimentos, que a técnicas de agrupamento no tempo foi superior as outras duas técnicas em relação à taxa de reconhecimento.

Por outro lado, Bernardin et al. (2005) também aplicam HMM para a segmentação e reconhecimento das seqüências de *grasp* em configuração de captura de dados usando luva de dados integrada com sensores táteis. Ekvall e Kragic (2005) também abordam o problema com o PbD. Eles apresentam três

modelos para reconhecer o tipo de *grasp*. O primeiro utiliza HMM para modelagem da sequência de posturas da mão, outra solução se baseia na trajetória da mão e na rotação da mão e a terceira é híbrida da primeira com a segunda. Martinez e Kragic (2008) usam uma combinação de SVM com *Conditional Random Field* (CRF) e HMM para modelagem e reconhecimento de ações objeto de manipulação no framework de aprendizado para robô.

Zollner, R. et al. (2001) usaram uma luva de dados integrada com sensores táteis, onde o método de classificação é baseado em SVMs. Chen, Y.-T. e Tseng (2007) também propuseram uma solução que utiliza a SVM para reconhecer gestos de mão. Li, Khan e Prabhakaran (2005) usam de decomposição de valor singular (*Singular Value Decomposition* - SVD) para gerar vetores de características para *grasps* humanos e SVMs, que são aplicados ao problema de classificação. Uma combinação de HMM e SVM é apresentada em (VICENTE et al., 2007) para a aprendizagem das ações de manipulação.

Uma abordagem lógica *fuzzy* para o reconhecimento de gestos foi publicado em (BIMBER, 1999). O método é aplicado a 6 graus de liberdade (6DOF - *Degree-of-Freedom*) nas trajetórias de um braço humano, mas não consegue lidar com nosso problema multi-dimensional. Bedregal e Costa (2005) também desenvolveram métodos baseados em regras *fuzzy* para o reconhecimento de gestos com as mãos, porém os métodos são altamente dependentes de uma análise prévia detalhada, feita de maneira manual, das características dos gestos a ser reconhecidos.

Além disso, as abordagens de rede neural demonstraram o seu potencial em reconhecimento de *grasp* no projeto *Glove-Talk* (FELS e HINTON, 1993, 1998). Máquina de Estados Finitos (*Finite State Machine* - FSM) também foi proposto para rastrear e reconhecer gestos na interação com o vídeo games em (STERN, WACHS e EDAN, 2006). Aleotti e Caselli (2006) descrevem uma realidade virtual baseada em um sistema PbD de reconhecimento de *grasp*, no qual as posturas dos *grasps* finais são modelados com base nos ângulos das articulações dos dedos.

Já o trabalho de Heumer et al. (2007) tentou fornecer um critério de desempenho para a classificação de *grasp* incluindo 28 algoritmos fornecidos pelo pacote de software Weka de mineração de dados. No entanto, estes algoritmos geralmente exigem mais parâmetros como a velocidade angular dos dedos ao invés usar somente as posições das trajetórias. Eles também recebem menos atenção no contexto da robótica inteligente, devido à exigência de reconhecimento em tempo real.

Capítulo 3

Conceitos Básicos para *Grasps*

A mão humana é usada em uma variedade de maneiras. Há três funções em relação ao pulso e à palma da mão que em particular, são mais importantes: explorar, agarrar (*grasp*), e manipular objetos. A primeira função se enquadra no âmbito da *haptics*, uma área ativa de pesquisa em seus próprios méritos (KLATZKY e LEDERMAN, 1990). Não vamos tentar fazer uma cobertura desta área. Os trabalhos atuais na robótica envolvendo *grasp* tenta compreender e desenvolver controles que imitar as duas últimas funções. Iremos fazer uma cobertura sobre a definição de *grasp*. Logo após, dissertaremos sobre a classificação dos *grasp* já catalogados pela literatura. Por fim, discutiremos uma plataforma de programação por demonstração na qual enunciaremos o problema alvo desse trabalho.

3.1 Definição de *Grasp*

Como o *grasp* o objeto central desse trabalho, foi necessário encontrar uma definição de uma ideia relevante para nosso trabalho. Algumas definições foram estudadas.

Segundo “*The American Heritage Dictionary of the English Language*” (GRAY, 1993) um *grasp* é definido por como “Segurar firme, ato de agarrar ou pegada”¹. Neste trabalho, usamos a proposta de definição da referência (FEIX et al., 2009) que é a seguinte:

O *grasp* é cada postura estática da mão com a qual um objeto pode ser

¹ tradução livre de “firm hold or grip”

preso por uma mão de forma segura².

Vale ressaltar que esta definição também implica que a estabilidade deve ser garantida, independentemente da direção da força relativa entre a mão e o objeto. Sendo assim, os movimentos intrínsecos são excluídos porque o objeto não está em um relacionamento constante com a mão. Tarefas que utilizam as duas mãos não são relevantes nesse trabalho. *Grasps* que dependem da gravidade também são descartados, pois a orientação da mão é vital para a estabilidade do *grasp*. No nosso caso, se alguém vira a mão, o objeto pode cair, o que mostra que não é independente da direção da força. Dessa forma alguns tipos de *grasps* foram excluídos.

Grasp estável pode ser dividido em dois tipos diferentes (PONCE et al., 1997):

Força de fechamento (*Force closure*) - segundo a qual o objeto é mantido na oposição dos dedos exercendo pressão contra a superfície do objeto.

Forma de fechamento (*Form closure*) - pela qual o objeto é envolvido pela mão de modo que o objeto não pode se mover, mesmo que os dedos da mão não exerça nenhuma força.

Muitas vezes os termos *grasp* e apreensão (*aprehension*) são usados indistintamente (MACKENZIE e IBERALL, 1994). Para outros, a apreensão é o ato coordenado de chegar a um objeto e agarrá-lo (DE KAMP, VAN e ZAAL, 2007). Kang e Ikeuchi (1992) afirmam que uma tarefa de *grasp* é composta de três etapas:

1. Fase de pré-*grasp*
2. Fase de *grasp* estático
3. Fase de manipulação

3.2 Classificação de *Grasps*

Várias classificações de *grasps* têm sido propostas ao longo do tempo. Schlesinger (1919) classificou os tipos de *grasp* de acordo com os formatos dos objetos que os *grasp* foram aplicados. Napier (1956) propôs uma classificação bastante simplista de tipos de *grasp*. Ele dividiu em *grasp* de força, que

² Definição original em inglês - “A *grasp* is every static hand posture with which an object can be held securely with one hand.”

geralmente utiliza a palma da mão, e *grasp* de precisão, que só usa os dedos e o polegar. Cutkosky (1989) usa uma hierarquia de árvore de classificação de tipos de *grasp*, começando com os tipos de *grasp* básicos propostos por Napier, no caso, *grasp* de força e *grasp* de precisão. Kang e Ikeuchi (1992) identificaram que os modelos mencionados acima carecem de uma descrição formal do *grasp* o que os tornam impróprios para o reconhecimento automático do tipo de *grasp*. Eles propuseram a “teia de contato” (*contact web*), como uma classificação formal de *grasps* que é baseado nos pontos de contato entre o objeto e os dedos. Feix et al. (2009) encontraram 147 descrições de tipos de *grasp* em uma revisão de 17 publicações. Eles também identificaram que desses 147 tipos somente 45 tipos era distintos. Uma classificação com base na definição de *grasp* citada anteriormente reduziu esses 45 para 33 tipos de *grasp* validos. Estes foram organizados de forma hierárquica com base nos dedos envolvidos, a distinção de Napier, o tipo de oposição, e a posição do polegar.

Esse TG tomou com base para classificação dos tipos de *grasp* a taxinomia proposta em (FEIX et al., 2009). Nessa taxonomia, como foi comentado anteriormente, há 33 tipos diferentes de *grasp*. Na Figura 4 mostramos os 31 tipos de *grasp* e na Figura 6 os 2 tipos omitidos.

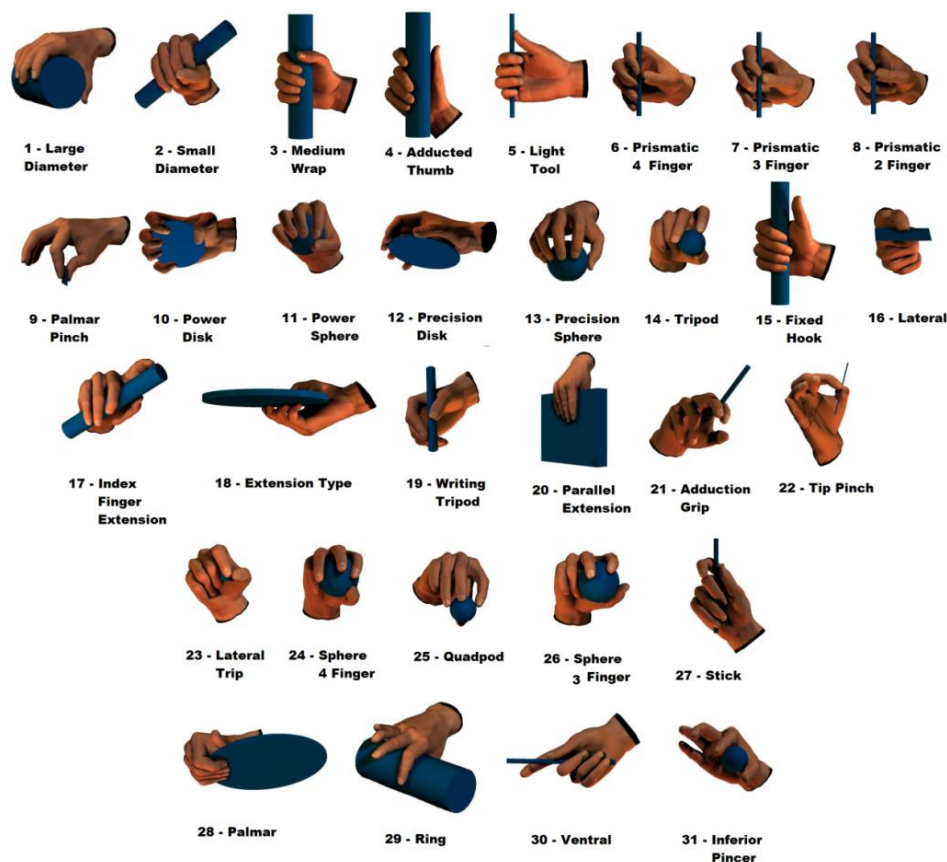


Figura 4 - Tipos de Grasp Usados.

Fonte: Criação própria a partir das figuras de <<http://grasp.xief.net/grasplist.php>>.

Os tipos de *grasp* foram então organizados hierarquia representada na Figura 5. Abaixo, fazemos uma breve descrição das hierarquias.

3.2.1 Decrição da Taxonomia

Grasp de força (*power*) - A necessidade primária é a força.

Grasp intermediário (*intermediate*) - Um estágio intermediário entre força e precisão. Nesses tipos de *grasp*, é necessário força e precisão.

Grasp de precisão (*precision*) - Exatidão em pegar o objeto é o principal objetivo. Comparado com os tipos força e intermediário, somente uma pequena força é aplicada.

Tipo de oposição (*opposition type*) - refere-se à orientação básica em que uma mão humana pode aplicar forças:

Oposição *Pad* - As forças são aplicadas entre os dedos da mão, ao longo de uma direção mais ou menos paralela ao eixo distal / proximal.

Oposição *Palm* - A palma da mão é usada para agarrar o objeto e do outro lado o objeto é segurado por um ou mais dedos. O eixo ao longo do qual as forças estão sendo aplicadas é aproximadamente perpendicular à palma da mão.

Oposição *Side* - o lado de um ou mais dedos é usado para segurar o objeto na mão. A direção da força é mais ou menos paralela ao eixo radial/ulnar.

Dedo Virtual (*Virtual Finger*) - No conceito de dedos virtual, dois deles estão a trabalhar uns contra os outros para criar a força necessária para segurar firmemente o objeto. A palma da mão e/ou os dedos podem ser mapeados em um dedo virtual definindo seu "tamanho". Existem algumas abreviações usadas:

- P palma
- 1-5 - Polegar (1) ... dedo mínimo (5)
- VF1 Virtual dedo 1
- VF2 Virtual dedo 2
- VF3 Virtual dedo 3, usado para se opor a uma tarefa relacionada força ou momento.

Polegar (*Thumb*) - Define a posição do polegar

Abd - A articulação CMC do polegar é abduzida.

Add - A articulação CMC do polegar é aduzida.

Opposition Type: Virtual Finger 2:	Power						Intermediate			Precision				
	Palm		Pad				Side			Pad				Side
	3-5	2-5	2	2-3	2-4	2-5	2	3	3-4	2	2-3	2-4	2-5	3
Thumb Abd.														
Thumb Add.														

Grasp Taxonomy, 13.10.2010

GRASP / Otto Bock / Thomas Feix

Figura 5 - Taxonomia abrangente de *grasps*, que inclui 33 tipos de *grasp*.

Fonte: Criação própria a partir das figuras de <<http://grasp.xief.net/taxonomy.php>>.

Foram listados 31 tipos de *grasps* nesta taxonomia. Os tipos de *grasp* “Distal Type” e “Tripod Variation” foram removidos devido a sua natureza muito específica. A figura abaixo mostra os dois diferentes tipos de *grasp* que foram omitidos nesse trabalho.

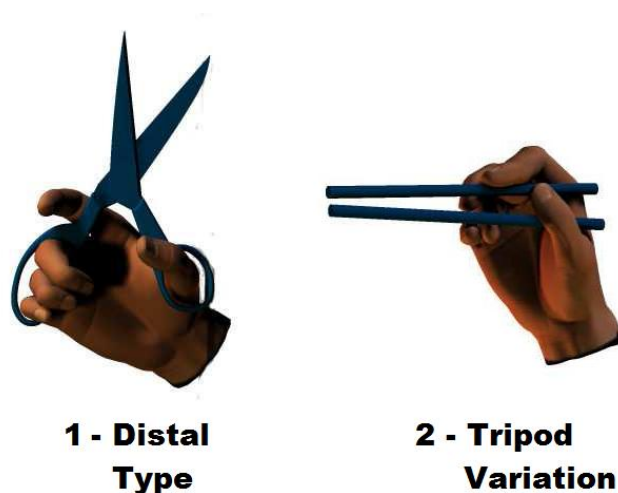


Figura 6 - Os dois tipos de *grasp* omitidos nesse trabalho.

Fonte: Criação própria a partir das figuras de <<http://grasp.xief.net/taxonomy.php>>.

3.3 Plataforma de Programação por Demonstração

Programação por demonstração (PbD) é uma técnica em que um computador ou um robô aprende novos comportamentos, através da demonstração da tarefa por um operador. O contexto de reconhecimento de gestos da mão para robôs é esquematicamente apresentado na Figura 7. Essa técnica envolve a interpretação de manifestações humanas e uma conversão automática, dessa interpretação, em código de programa que podem ser executados por um dado sistema robótico. A metodologia PbD para *grasp* robótico requer três tarefas principais: segmentação de sequências de *grasp*, reconhecimento do *grasp* e geração de código robô (ALEOTTI e CASELLI, 2006). A primeira tarefa é a partição do registro de dados em uma sequência de episódios, no qual cada um contém (representa) um único instante da execução dos movimentos da mão. A segunda tarefa é reconhecer o *grasp* realizado. Finalmente a tarefa demonstrada é convertida em um programa de nível de tarefa que pode ser executada por uma mão robótica artificial ou por uma prótese de uma mão. Este programa executa o *grasp* primitivo correspondente a *grasp* humano observado e se adapta ao dado sistema robótico. Nesse trabalho, seguimos com um detalhamento dos problemas de modelagem e reconhecimento de sequências de *grasp* humano. Não abordaremos, na sequência, este programa de alto nível, que gera código para o robô.

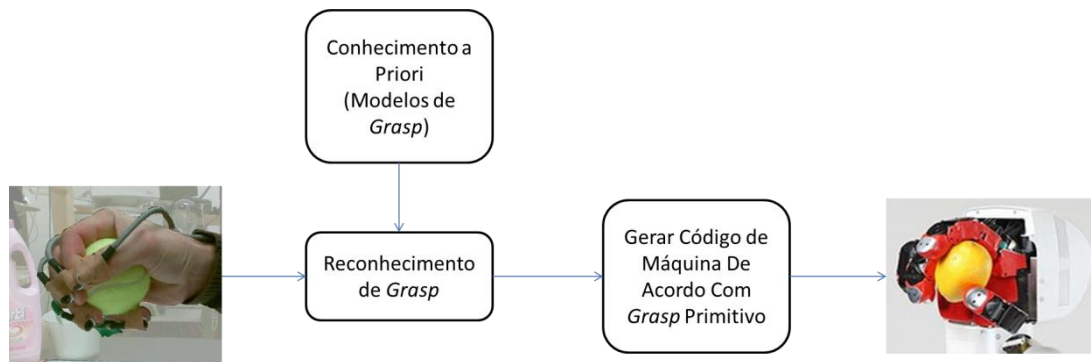


Figura 7 - - Programação por demonstração em um contexto de *grasp*.
Fonte: Criação própria com base na figura 1 de (JU et al., 2008).

A plataforma experimental consiste de um dispositivo de captura de movimento da mão e um ambiente de simulação da mão. A maioria dos trabalhos relacionados usa de luvas de dados disponíveis no mercado para captura dos movimentos. Nesse trabalho, usamos uma base de dados que fez a captura através de um sistema de Liberdade *Polhemus* com seis sensores magnéticos. Mais detalhes sobre os experimentos e a base de dados vide o

capítulo de experimentos.

Para a modelagem da ponta dos dedos, tamanhos menores ou maiores de mãos têm pouca influência nos resultados da modelagem, desde que eles não se afastem demasiadamente uns dos outros. Já que mãos humanas usam um número limitado de tipos de *grasp*, o processo de reconhecimento pode ser restrito a uma taxonomia de *grasp*. Nesse trabalho, como foi dito anteriormente, a taxonomia abordada foi (FEIX et al., 2009).

Capítulo 4

Rede Neural para *Grasps*

Nesse capítulo apresentamos um novo método para o reconhecimento do tipo de *grasp*. Fizemos adaptações na rede neural de mapas auto-organizáveis (*self-organizing Maps* - SOM) (KOHONEN, 1990) para solucionar o problema do reconhecimento. Também, usamos a abordagem proposta em (KANGAS, 1990, 1991) para que a rede neural leve em conta a dependência do tempo das características. Em um primeiro momento, vamos dar uma visão geral da estrutura da rede neural de mapas auto-organizáveis. Logo após, detalhamos como inserimos a característica temporal nos padrões a ser aprendidos pela rede neural. Por fim, demonstraremos como adaptamos a rede neural para o problema do reconhecimento.

4.1 SOM – Mapas Auto-Organizáveis

As redes mapas auto-organizáveis (*Self-organizing Maps* - SOM) também chamadas de mapas auto-organizáveis de característica (*Self-organizing Feature Maps* - SOFM) foram desenvolvidas por Teuvo Kohonen na década de 80 (KOHONEN, 1982) e tem sido usado com sucesso para resolver um grande número de aplicações em reconhecimento de padrões, Ciências e Engenharia. O mapa auto-organizável de características de Kohonen é um modelo de rede neural competitiva com a propriedade reduzir padrões de alta dimensionalidade em um mapa de dimensão inferior. Em geral, o mapa é constituído por uma grade uni ou bidimensional de nodos (veja Figura 8). O mapa topográfico é automaticamente organizado através de um processo cíclico de comparação dos padrões de entrada com o vetor peso de cada nodo. Dessa

forma, o algoritmo de aprendizagem organiza os nodos em uma malha onde cada região desta malha atua como um "categorizador de características" dos dados.

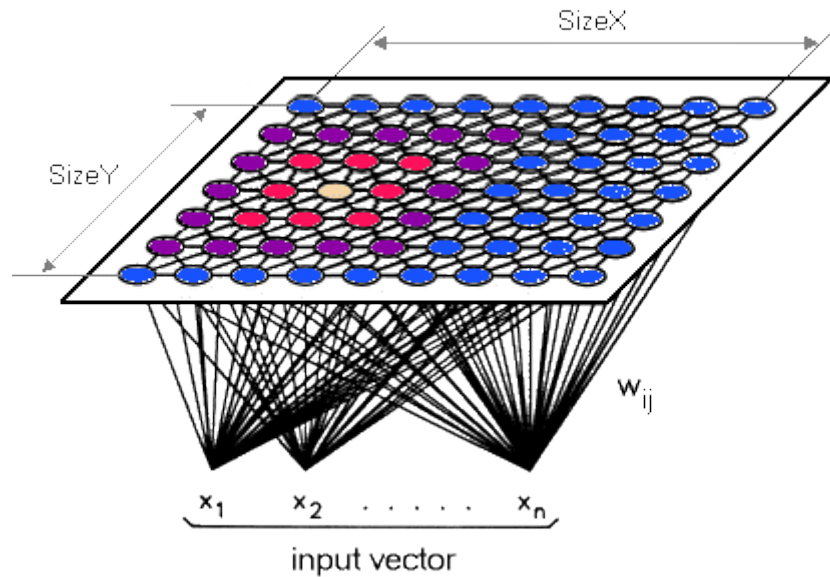


Figura 8 - Exemplo de topologia usada para SOM. As unidades de entrada são conectadas a todos os nodos da camada de saída.

Fonte: <http://www.lohninger.com/helpsuite/kohonen_network_-_background_information.htm>

A rede SOM funciona basicamente da seguinte forma: quando um padrão de entrada $\mathbf{p}$ é apresentado, a rede procura a unidade mais parecida com $\mathbf{p}$, isto é, o nodo mais próximo do padrão de entrada segundo uma métrica. Esse é chamado de nodo vencedor. Durante o seu treinamento, a rede aumenta a semelhança do nodo vencedor e de seus vizinhos ao padrão $\mathbf{p}$ através de uma regra de ajuste que está especificada abaixo. O processo para quando o mapa converge. Dessa forma, a rede constrói um mapa topológico onde nodos que estão topologicamente próximos respondem de forma semelhante a padrões de entrada semelhantes.

A regra de ajuste dos pesos tem a seguinte forma:

$$\mathbf{w}_i(k+1) = \mathbf{w}_i(k) + \eta(k) \cdot h_{bmu,i}(k) \cdot (\mathbf{x}(k) - \mathbf{w}_i(k))$$

, na qual $\mathbf{w}_i(k)$ é o vetor de pesos da unidade i no instante k e normalmente $\mathbf{w}_i(0)$ é um valor aleatório; $\mathbf{x}(k)$ é o vetor de entrada apresentado no instante k ; $\eta(k)$ é a taxa de aprendizagem representada por uma função que monotonicamente decresce à medida que k cresce e tem valor entre 1 e 0; $h_{bmu,i}(k)$ é a função vizinhança cujo valor depende da unidade vencedora $v(k)$, das outras unidades i e do instante k . $h_{bmu,i}(k)$ é uma função com valor máximo

unitário e mínimo nulo e decrescente à medida que se distancia do nó vencedor. Para assegurar a convergência $\eta(k) \rightarrow 0$ e $h_{bmu,i}(k) \rightarrow 0$ quando $k \rightarrow \infty$. Em geral, a função gaussiana é usada para este fim:

$$h_{bmu,i}(k) = e^{\left(-\frac{\|r_{bmu}-r_i\|^2}{2\sigma^2(k)}\right)}$$

, em que $r_{bmu} \in \mathbb{R}^2$ e $r_i \in \mathbb{R}^2$. $\sigma(k)$ define a região de vizinhança e é uma função monotônica decrescente. O treinamento é realizado um número fixo de vezes, $k = 1, 2, \dots, k_{max}$ ou até que os pesos tenham pequena variação. Mapas auto-organizáveis são, possivelmente, o modelo de rede não-supervisionada com maior número de aplicações. As propriedades dos mapas auto-organizáveis os tornam uma ferramenta bastante útil. O SOM tem se mostrado útil em um grande número de aplicações, nos mais diversos campos, a saber (HAYKIN, 1999): controle de braços de robô, quantização vetorial, equalização adaptativa, análise e reconhecimento de voz, visualização de faltas em máquinas, etc. Abaixo temos o pseudocódigo do treinamento dessa rede neural.

Pseudocódigo do treinamento:

1. Inicializar a rede
 - 1.1. Informa as dimensões da rede
 - 1.2. Inicializar pesos e parâmetros
 - 1.2.1. Definir $w_{ij}(k)$ ($0 \leq i \leq n - 1$) sendo o peso do nodo j referente ao atributo i no tempo k . Inicialmente os n pesos de todos os nodos tem valores aleatórios pequenos.
 - 1.2.2. Definir valor inicial do raio de vizinhança de cada nodo. Geralmente, um valor muito grande que engloba todo o mapa.
2. Repetir
 - 2.1. Para cada padrão de treinamento x faça
 - 2.1.1. Definir nodo vencedor

Selecionar o nodo mais próximo ao padrão de treinamento
 - 2.1.2. Atualizar os pesos deste nodo e de seus vizinhos.
 - 2.1.3. Se o número do ciclo for múltiplo de N

Então Reduzir taxa de aprendizado e área de vizinhança
3. Até o mapa de características não mudar

4.2 Consideração Temporal nos Mapas Auto-Organizáveis

Para esse problema do reconhecimento do tipo de *grasp* há necessidade de

incorporar a dependência temporal nos padrões a ser aprendido pela rede SOM. Para esse problema, usamos de uma proposta de Kangas, que consiste da concatenação dos dados.

A solução de concatenação dos vetores de entrada no tempo é bastante simples. Os padrões de entrada são transferidos para um vetor onde são anexados um atrás do outro a cada passo de tempo. Esse vetor com a concatenação dos padrões de entrada é chamado de vetor janela de tempo (veja Figura 9). Nessa janela, contém a história do padrão de entrada e seu tamanho é fixado de antemão.

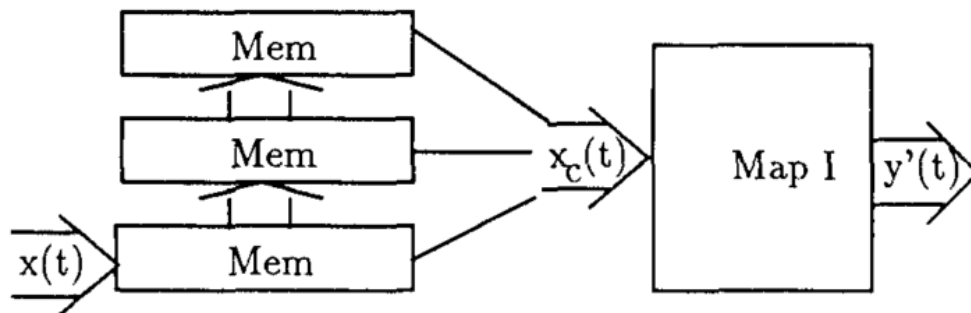


Figura 9- O mapa auto-organizável treinado com padrões de entrada concatenados. No exemplo, três padrões de entrada são concatenados para formar o vetor de entrada da rede.
Fonte: Figura 2 de (KANGAS, 1991).

Segundo Kangas, através dessa técnica pode ser possível obter bons resultados de reconhecimento com base no conhecimento da história, pois a informação dos padrões de entrada anteriores está sempre presente (KANGAS, 1990). Mas essa técnica não elimina o ruído. Outro detalhe, é que o tempo de amostragem é fixo, ou seja, sempre há o mesmo número de vetores de entrada concatenados.

4.3 Adaptações da Rede SOM

Nessa seção relatamos as adaptações que realizamos no SOM para atacar o problema de reconhecimento de tipos de *grasp*. No caso, mantivemos a estrutura geral das redes SOM e modificamos a rede neural para que ela aprenda as relações temporais entre os padrões e as posturas das mãos no movimento de *grasp*. Para isso usamos a proposta de Kangas descrita na seção anterior. Também inserimos alterações nas etapas de busca do nodo vencedor colocando o conceito de “termo de responsabilidade”, como também, modificamos a expressão de alteração dos pesos inserindo critério “cópia da

primeira entrada.” Por fim, descreveremos o processo de categorização utilizando o SOM.

4.3.1 Alteração da Concatenação dos Padrões

No problema de reconhecimento de tipo de *grasp*, temos uma execução de gestos de uma mão fazendo o ato de agarrar um objeto. Isto é, para cada instante de tempo temos um conjunto de dados sobre a postura da mão. Ao invés de passar cada conjunto de dados referente a uma postura da mão para que seja aprendida pelo SOM, utilizamos da técnica de concatenação dos conjuntos de dados referente às posturas da mão. Sendo assim, o padrão de entrada a ser aprendido pela rede é um vetor de postura da mão no ato de executar um determinado *grasp*. No caso, podemos fazer uma concatenação de todas as posturas ou somente uma sequência de posturas. Abaixo temos uma descrição matemática do vetor de entrada do SOM.

Padrão: $\xi = [x_1 \cdots x_n]^T$ onde x_i = vetor de dados da postura da mão no instante de tempo i .

4.3.2 Alterações no Treinamento do SOM

Na etapa de treinamento da rede SOM, alteramos o processo de busca do vencedor adicionando um detalhe que chamamos de “termo de responsabilidade”. Também, nessa etapa, modificamos a função de ajuste de pesos do vencedor colocando o conceito de “copia da primeira entrada”.

O termo de responsabilidade consiste do conceito que se um nodo já foi vencedor, ele só pode ser novamente vencedor caso ele seja o nodo mais próximo do padrão de entrada e esse padrão de estar dentro de um raio de responsabilidade que determina a vizinhança expressando a semelhança admitida entre o protótipo e o vetor por ele representado. Em nossa implementação, usamos, como raio de responsabilidade, uma hiper-esfera de raio r . Caso o nodo ainda não tenha vencido nenhuma competição anterior, para ele ser o nodo vencedor basta ele ser o nodo mais próximo do padrão de entrada. Se todos os nodos da rede já foram vencedores, fazemos com que todos os nodos tenham raio $r = \infty$. Introduzimos esse termo de responsabilidade para que os primeiros nodos vencedores não tenham vantagem em relação aos nodos que ainda não foram vencedores.

O critério de “copia da primeira entrada” consiste de uma modificação no processo de atualização dos pesos do nodo vencedor caso ele não tenha nenhum registro anterior de vitória. Nesse caso, o padrão de entrada é copiado

pelo nodo, sendo assim, o vetor de pesos desse nodo se torna uma copia do padrão de entrada. Se o nodo vencedor tiver registro de vencedor em uma iteração anterior o seu processo de atualização de peso continua semelhante à proposta original do SOM. Adicionamos esse critério para que o processo de aprendizagem seja acelerado nos primeiros passos do treinamento. Abaixo temos o pseudocódigo da solução:

Pseudocódigo do treinamento:

1. Inicializar a rede
 - 1.1. Informa as dimensões da rede
 - 1.2. Inicializar pesos e parâmetros
 - 1.2.1. Definir $w_{ij}(k)$ ($0 \leq i \leq n - 1$) sendo o peso do nodo j referente ao atributo i no tempo k . Inicialmente os n pesos de todos os nodos tem valores aleatórios no intervalo $[0,499 \text{ e } 0,501]$.
 - 1.2.2. Definir valor inicial do radio de vizinhança de cada nodo. Geralmente, um valor muito grande que engloba todo o mapa.
2. Repetir
 - 2.1. Para cada padrão de treinamento x faça
 - 2.1.1. Definir nodo vencedor

Selecionar o nodo mais próximo ao padrão de treinamento

Termo de responsabilidade para o nodo vencedor

Se o nodo ainda não foi vencedor

Então esse é o nodo vencedor

Se o nodo já foi vencedor e $\|w_i - \xi\| \leq r$

Então esse é o nodo vencedor

Caso contrário,

Escolha o nodo não vitorioso mais próximo como nodo vencedor

Se todos os nodos já foram vencedor

Então para todos os nodos faz $r = \infty$
 - 2.1.2. Atualizar os pesos deste nodo e de seus vizinhos

Cópia da 1ª entrada

Se for a 1ª vezes como vencedor

Então $w_i = \xi$

Caso contrário,

$w_i = w_i + \eta(k)(\xi - w_i)$
 - 2.1.3. Se o número do ciclo for múltiplo de N

Então Reduzir taxa de aprendizado e área de vizinhança
3. Até o mapa de características não mudar

4.3.3 Rotulação dos Nodos e Classificador

Após a etapa de treinamento da rede, usamos um processo de votação para

rotulação de cada nodo do SOM. Para rotulação, usamos o conjunto de treinamento no qual cada instância desse conjunto é apresentada à rede para determinação do nodo vencedor. Este último atribui um voto para ser rotulada como a classe da instância que foi mostrada para a rede. A fim determinar o rótulo de todos os nodos (considerando todas possíveis categorias), são computados os votos de cada nodo e nodo recebe a rotulação da categoria que tem maior votação. Caso o nodo não tenha nenhum voto, ou seja, ele não foi vencedor para nenhuma instância de treinamento, então ele é descartado.

Para categorizar uma instância de teste, basta encontrar o nodo vencedor para esta instância e atribuir a ela a classe do nodo. Abaixo temos o pseudocódigo da solução para rotulação dos nodos e categorização das instâncias de teste:

Pseudocódigo para rotulação:

1. Para todas as instâncias do conjunto de treinamento
 - 1.1. Definir nodo vencedor
 - 1.1.1. Selecionar o nodo mais próximo ao padrão de treinamento atual
 - 1.2. Incrementar um voto para a classe da instância atual no nodo vencedor
2. Para todos os nodos da rede
 - 2.1. Se total de votos para nodo atual é nulo
Então remover nodo
 - 2.2. Caso Contrário,
Atribuir o rotulo da classe com maior votação a esse nodo

Pseudocódigo para categorização:

1. Para todas as instâncias do conjunto de teste
 - 1.1. Definir nodo vencedor
 - 1.1.1. Selecionar o nodo mais próximo ao padrão de teste atual
 - 1.2. Definir a classe da instância de teste a classe do nodo vencedor

Capítulo 5

Experimentos

Neste capítulo apresentamos avaliação experimental realizado para a solução proposta. Para tais experimentos, foi utilizada uma base de dados de gestos de *grasps*. No caso, a base de dados é a “*Human Grasping Database*”. Primeiramente, discutiremos sobre a base de dados usada nos experimentos. Logo após, descreveremos os experimentos realizados para determinar as pose da mão que mais discrimina os tipos de *grasp*. Por último, mostraremos os experimentos de reconhecimento de tipo de *grasp* com seus resultados e discussão.

5.1 Base de Dados Human Grasping

A base de dados “*Human Grasping Database*” (ROMERO, EK, & KRAGIC, 2009) foi criada para fornecer dados para o desenvolvimento do estudo de classificação dos tipos de *grasp* e avaliação das propriedades dos diversos tipos. Ela foi resultado da pesquisa feita pelo laboratório Otto Bock, em colaboração com KTH (*Kungliga Tekniska Högskolan*) de Estocolmo, na Suécia. Essa base de dados faz parte dos trabalhos realizados pelo projeto GRASP da União Europeia. A medição foi realizada no LMU (*Ludwig-Maximilian University*) em Munique no Laboratório de Heiner Deubel em colaboração com Javier Romero de KTH, Estocolmo, Suécia.

Human Grasping Database possui dados gerados por 5 indivíduos (3 do sexo masculino e 2 do sexo feminino). Todos os indivíduos são destros e não relataram qualquer deficiência de mão. O comprimento médio e largura média de mão foram 185,2 milímetros e 81,1 milímetros, respectivamente, com desvio

padrão 13,3 milímetros e 7,4 milímetros. Um sistema de Liberdade Polhemus com seis sensores magnéticos foi utilizado para a captura e gravação dos dados. A resolução espacial e angular de cada sensor é de 0,8 mm e 0,15 graus respectivamente. Um sensor foi aplicado para cada dedo, colocado na unha e um foi colocado sobre o dorso da mão. Veja a figura abaixo para saber como os sensores foram aplicados à mão.

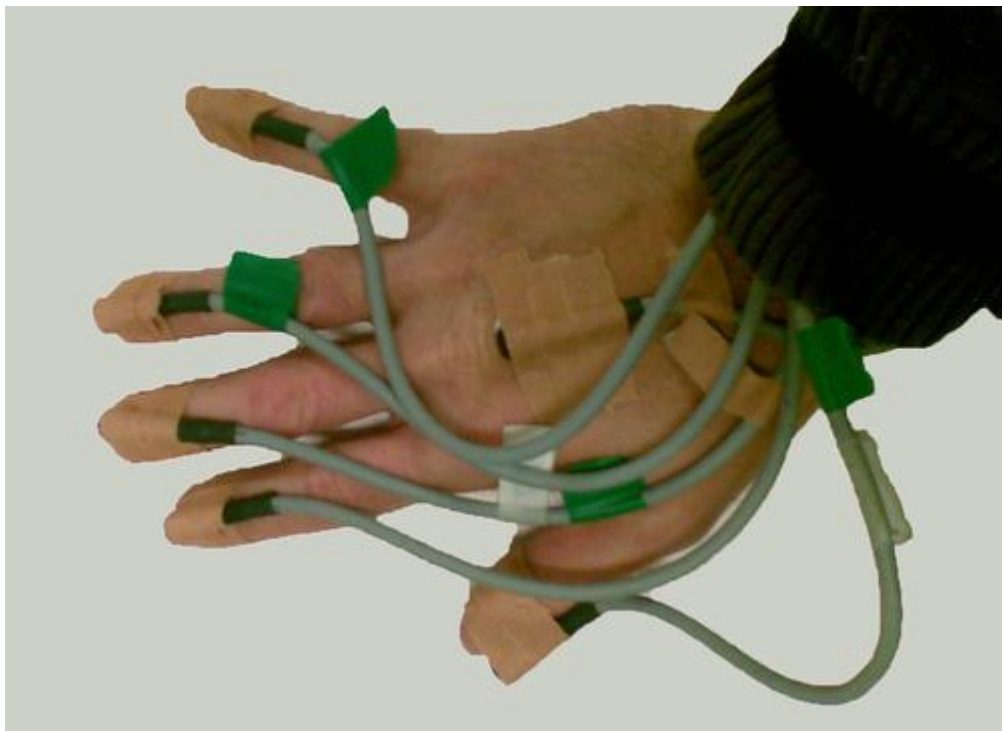


Figura 10 - Mão com os 6 sensores.
Fonte: < <http://grasp.xief.net/dataset.htm> >.

Os sujeitos foram solicitados a realizar 31 tipos de *grasp* de um conjunto de 33 que foi mostrado no Capítulo 3. Como foi dito anteriormente, foram excluídos os tipos de *grasps* “*Distal Type*” e “*Tripod Variation*”, devido à sua natureza muito específica. Uma foto de cada *grasp* foi mostrada aos sujeitos e uma demonstração do *grasp* foi realizada caso o sujeito tenha apresentado dificuldades para reproduzir o *grasp*. Em seguida, os sujeitos foram instruídos a realizar o *grasp* especificado. Inicialmente, a mão foi colocada na frente da mesa em uma postura mão espalmada (ou mão aberta). A partir de um sinal de início, o sujeito agarrou um objeto com o tipo de *grasp* desejado, levantou o objeto (momento mostrado na Figura 11), colocou o objeto novamente sobre a mesa e retornou sua mão para a posição inicial. A captura e o armazenamento de dados começaram quando a mão começou a se mover e terminou quando a mão retornou à posição inicial. Cada tipo de *grasp* foi realizado duas vezes, a

primeira foi usada para criar o conjunto de teste e uma segunda para o conjunto de treinamento.

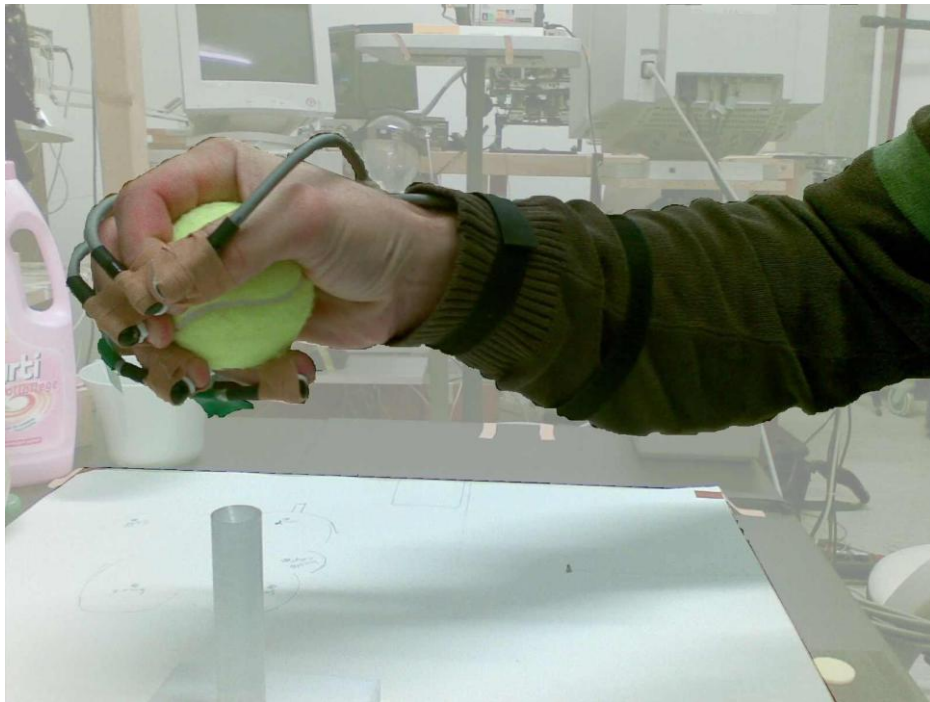


Figura 11 - Ilustração do tipo de *grasp* "Power Sphere."

Fonte: < <http://grasp.xief.net/dataset.htm> >.

Nesse caso, os sensores na ponta dos dedos foram transformados para o sistema de coordenadas em relação ao pulso para eliminar os movimentos da mão global. Então os dados foram processados da seguinte forma:

1. Calibração que alinha os sistemas de coordenadas dos sensores com a direção anatômica atual.
2. Transformação dos dados da ponta dos dedos no sistema de coordenadas do pulso. Isso é feito para que os movimentos da postura da mão sejam definidos em termos da posição relativa ao pulso. Sendo assim, dá-se uma menor variância dos movimentos de um mesmo tipo, pois não há a presença dos movimentos globais da mão.
3. Tradução da posição da origem dos dedos para o ponto central da falange distal, que é o osso que fica na extremidade do dedo, e normalização das dimensões para uma escala padrão.

Os movimentos gravados são armazenados em uma matriz, a qual podem ser baixadas em diferentes versões. O tamanho da matriz é de 5 (sujeitos) x 2 (ensaio) x 31 (tipos de *grasp*) x N (amostras) x D (dimensões). No

caso, usamos a base de dados $N = 30$ amostras.

A dimensão D do conjunto de dados é 60 (5×12) e estão distribuídos da forma ilustrada pela a figura abaixo. Porém, para nossos experimentos só usamos as posições dos dedos. Isto é, todas as categorizações serão feitas com um conjunto bem menor de informações que aquele usado por propostas anteriores.

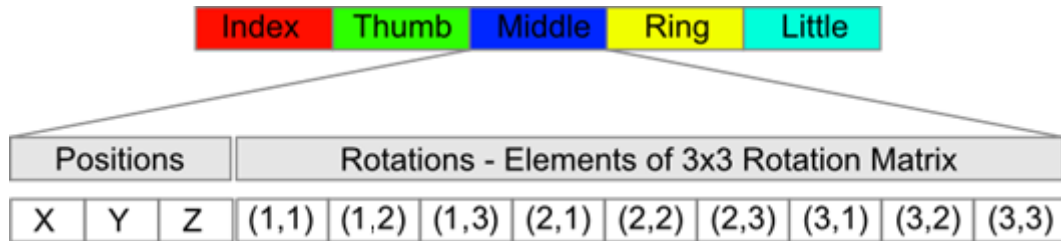


Figura 12- Ilustração da distribuição dos dados.

Fonte: < <http://grasp.xief.net/dataset.htm> >.

Além disso, os dados não processados também são fornecidos, ou seja, os dados que não foram transformados para sistema de coordenadas em relação ao pulso. Portanto, para esses dados, o movimento global da mão ainda está presente nos dados. Como nesse caso há um sensor adicional (o sensor de pulso) a dimensão de D aumenta em 12. Os dados do pulso estão entre o polegar e o dedo médio. Isso significa que a ordem dos dedos nos dados brutos é: Indicador (*Index*) – Polegar (*Thumb*) – dorso (*Wrist*, local onde está o sensor de referência) – Dedo Médio (*Middle*) – Dedo mínimo (*Little*). Nos nossos experimentos não usamos a base de dados não processadas e os dados foram normalizados, ficando entre 0 e 1.

5.2 Experimento - Posturas Mais Discriminantes

Foi feito um experimento, sem utilizar técnicas formais, para encontrar as poses da mão que mais discrimina os diversos tipos de *grasp*. Para decidir quais poses são mais discriminantes verificamos através da soma das diferenças absolutas entre as poses intraclasses (D.I.) e enterclasses (D.E.).

A diferença absoluta entre dois padrão é calculada da seguinte forma:

$$\vec{D} = \begin{bmatrix} |\vec{x}_1 - \vec{y}_1| \\ \vdots \\ |\vec{x}_n - \vec{y}_n| \end{bmatrix}$$

,onde $\vec{x}_i$ é o vetor de dados da pose i do padrão $\mathbf{x}$.

Calculamos, dois a dois, os padrões de treinamento de uma mesma classe e somamos as diferenças absolutas de cada classe. Também fizemos a soma das diferenças absolutas de cada dois padrões de classes distintas. Com isso temos a soma da variação de cada classe e entre duas classes. Abaixo temos as formulas matemáticas dos cálculos.

$$D.I. da classe A = \sum_{x,y \in A} \begin{bmatrix} |x_1 - y_1| \\ \vdots \\ |x_n - y_n| \end{bmatrix}$$

$$D.E. entre a classe A e B = \sum_{x \in A, y \in B, A \neq B} \begin{bmatrix} |x_1 - y_1| \\ \vdots \\ |x_n - y_n| \end{bmatrix}$$

Lodo após calcular essas diferenças foi calculada a média da diferença entre poses de uma mesma classe e entre classes distintas. A partir dessas médias fizemos a subtração da média interclasses menos intraclasses. Com isso, selecionamos empiricamente uma sequência de n poses cujas subtrações forma maiores. Abaixo mostramos a tabela com a média intraclasse e interclasse, como também, a subtração de ambos.

Tabela 1 - Tabela da diferença da média intraclasse e entreclasse.
Fonte: Criação própria.

Amostra de Pose	Média Intraclasse (M.I.)	Média Entreclasse (M.E.)	Diferença (M.E. - M.I.)
1	84,72339701	87,49886224	2,77546523
2	73,17729999	76,2162925	3,03899252
3	59,55731073	67,39637628	7,83906554
4	53,48128577	64,3107344	10,8294486
5	54,39290053	64,59381214	10,2009116
6	53,84549489	63,55414181	9,70864692
7	61,56665017	67,94646938	6,37981921
8	54,8050227	65,65724573	10,852223
9	52,03912894	64,87024857	12,8311196
10	50,85830351	64,1751211	13,3168176
11	48,73631575	63,40286018	14,6665444
12	48,70769525	63,19891861	14,4912234
13	48,82846675	63,35271715	14,5242504
14	49,51824276	63,05501782	13,5367751
15	49,76767862	63,5601957	13,7925171
16	49,72344975	62,92718978	13,20374
17	50,02653659	62,54400911	12,5174725
18	49,38100864	62,31377868	12,93277

19	49,23828757	61,87634342	12,6380559
20	49,43910926	61,71329992	12,2741907
21	52,79378314	62,41158093	9,61779779
22	59,34081525	66,15683322	6,81601797
23	61,63933734	68,59248608	6,95314874
24	62,0784521	67,54041955	5,46196745
25	66,43343076	70,35866947	3,92523871
26	73,06971784	77,89307345	4,82335561
27	80,13791865	83,67713814	3,53921948
28	95,64552705	96,31572039	0,67019334
29	96,52932989	97,36487979	0,83554991
30	97,2026609	98,54987542	1,34721451

No caso, verificamos experimentalmente que a escolha da sequência de poses partindo da pose 8ª até 20ª os resultados eram melhor do que usar todas as 30 amostras.

5.3 Experimentos - Reconhecimento de Tipos de *Grasp*

Como foi relatado na seção 5.1, testamos 31 tipos diferentes de *grasps* primitivos usados na confecção da base de dados “*Human Grasping Database*”. Como foi dito na seção anterior, usou 13 amostras de poses da mão para cada *grasp* para realizar uma modelagem. Nosso intuito é que essa modelagem discrimine mais os tipos diferentes de *grasp* do que se usássemos todas as 30 amostras. No caso, usamos da 8ª amostra de pose à 20ª.

Primeiro, os *grasps* primitivos (modelos de *grasp*) foram treinados com o conjunto de treinamento da própria base de dados, como também testamos o sistema de reconhecimento através do conjunto de teste da base de dados.

Para treina a rede SOM, usamos a função de aprendizagem com a seguinte configuração paramétrica.

Para o vencedor:

$$\eta(k) = 0,9 \cdot e^{\left(-\frac{k}{750}\right)} + 0.0001$$

, onde k é o número de vitórias do nodo vencedor.

Para os vizinhos:

$$\eta(D) = 0,4 \cdot e^{\left(-\frac{D}{2.5}\right)}$$

, onde D é a distância de Manhattan entre o vizinho e o nodo vencedor.

A janela de vizinhança começa como um quadrado que encobre todo o mapa onde o centro do quadrado está o nodo vencedor. Essa janela decresce rapidamente. A janela de vizinhança de cada nodo decrementa de uma unidade a cada duas vitórias desse nodo.

Abaixo temos uma tabela com os parâmetros restantes da rede SOM para a etapa do treinamento.

Tabela 2 - Tabela de parâmetros do treinamento da rede.
Fonte: Criação própria.

Parâmetro	Valor
Tipo de Treinamento	Batch
Valor Inicial dos pesos	Valor aleatório entre 0,499 e 0,501
Dimensões do mapa	100x100
Critério de parade	200 épocas
R	0,01

Como foi dito anteriormente, usamos as amostras da 8ª à 20ª poses para discriminar os tipos de *grasp*. Implementando a proposta de Kangas, discutida no Capítulo 4, fazemos um vetor de concatenação dessas 13 amostras de pose. Sendo assim, esse vetor é o padrão de entrada para rede SOM, que deve ser aprendido.

Para a experimentação da classificação dos tipos de *grasp* do conjunto de teste foi criado para cada instância desse conjunto uma vetor de concatenação das amostras correspondentes às amostras que foram verificadas serem as mais discriminantes. No caso, a janela das amostras começando da 8ª amostra até a 20ª amostras. Para esse vetor verificamos qual é o nodo vencedor e atribuímos a essa instância de teste a classe que o nodo vencedor está rotulado. Lembramos que a rotulação já foi explicada no Capítulo 4.

5.4 Resultados e Discussão

O método proposto foi avaliado no experimento descrito na seção anterior. Para obtemos estes resultados, fizemos uma prototipação do nosso sistema através

do MATLAB®. Nos experimentos, treinamos 10 vezes a rede e obtivemos um resultado de uma taxa de reconhecimento de 80% na média. Na Tabela 3, 4 e 5 mostramos a taxa de acerto por tipo de *grasp*. Dividimos os tipos de *grasp* em três grupos assim como é feito no trabalho de (PALM, ILIEV e KADMIRY, 2009):

1. Tipos de *grasp* com taxa de acerto igual ou superior a 75%
2. Tipos de *grasp* com taxa de acerto menor que 75% e maior ou igual a 50%
3. Tipos de *grasp* com taxa de acerto inferior a 50%

Tabela 3 - Tipos de *grasp* com taxa de acerto $\geq 75\%$.
Fonte: Criação própria.

Id do tipo de Grasp	Tipo de Grasp	Taxa de acerto (%)
1	Large Diameter	100
2	Small Diameter	100
3	Medium Wrap	100
5	Light Tool	100
6	Prismatic 4 Finger	100
7	Prismatic 3 Finger	100
8	Prismatic 2 Finger	100
4	Adducted Thumb	80
9	Palmar Pinch	80
10	Power Disk	80
11	Power Sphere	80
12	Precision Disk	80
13	Precision Sphere	80
14	Tripod	80
16	Lateral	80
17	Index Finger Extension	80
18	Extension Type	80
19	Writing Tripod	80
20	Parallel Extension	80
21	Addiction Grip	80
22	Tip Pinch	80
23	Lateral Tripod	80
25	Quadpod	80
28	Palmar	80
29	Ring	80
31	Inferior Pincer	80

Tabela 4 - Tipos de *grasp* com taxa de acerto <75% & ≥50%.
Fonte: Criação própria.

Id do tipo de <i>Grasp</i>	Tipo de <i>Grasp</i>	Taxa de acerto (%)
24	Sphere 4 Finger	60
26	Sphere 3 Finger	60
27	Stick	60
30	Ventral	60

Tabela 5 - Tipos de *grasp* com taxa de acerto < 50%.
Fonte: Criação própria.

Id do tipo de <i>Grasp</i>	Tipo de <i>Grasp</i>	Taxa de acerto (%)
15	Fixed Hook	40

Verificamos através das três tabelas a cima que 26 de um total de 31 tipos de *grasp* podem ser reconhecidos com taxa superior a 75%. Porém alguns tipos de *grasp* apresentam certa dificuldade de ser identificado pelo sistema em relação a outros. O tipo de *grasp* que apresentou maior dificuldade de ser reconhecido foi o “15 - *Fixed Hook*”. Vemos através da matriz de confusão (que não apresentamos aqui, pois é uma tabela muito grande) que o sistema se confunde o tipo 15 com tipo 4 em 40% dos casos que deveria ser tipo 15. Acreditamos que isso é devido identificação se basear apenas na posição das pontas dos dedos para discriminar categorias diferentes. Acreditamos que o emprego dos ângulos dos dedos e a área de contato, possivelmente, o sistema iria discriminar os dois grupos mencionados. Abaixo mostramos a comparação visual do tipo “15 - *Fixed Hook*” e “4 - *adducted thumb*”.



Figura 13 Comparação visual do tipo “15 - *Fixed Hook*” e “4 - *adducted thumb*”.
Fonte: Criação própria

Capítulo 6

Conclusão e Trabalhos Futuros

O trabalho realizado visou à pesquisa e ao desenvolvimento de um sistema de reconhecimento de tipos de *grasp*. Essa tarefa de reconhecimento de *grasp* se enquadra em um das etapas da programação por demonstração (PbD), que consiste de uma abordagem, utilizada na robótica, para facilitar a programação que, nesse contexto, é a implementação de um controle para uma mão robótica executar *grasps*.

Discutimos, nesse trabalho, o estado da arte do desenvolvimento de mãos robóticas como foco principal nas soluções de reconhecimentos de *grasp* para abordagem de PbD. Também, mostramos uma plataforma de programação por demonstração.

Apresentamos uma solução para o problema de reconhecimento de tipo de *grasps*. Em nosso sistema de reconhecimento, usamos o mapa auto-organizável (SOM) e uma proposta de Kangas para inserir a aprendizagem temporal no problema tratado. Para o desenvolvimento de nossa solução utilizamos o MATLAB® para sua prototipação. Para experimentação usamos a base de dados “*Human Grasping Database*”, que possui 31 tipos primitivos de *grasp*. Nos nossos experimentos, realizamos estudos de quais poses do movimento da mão são mais discriminantes para o reconhecimento. Nosso classificador obteve taxa de acerto de 80% e vemos possíveis melhoras nessa taxa caso usássemos os ângulos dos dedos e as áreas de contatos além das posições dos dedos para discriminar os tipos de *grasps*. Uma vantagem, em relação aos modelos anteriores, que o nosso sistema de reconhecimento tem é que o sistema categoriza, através de aprendizagem de não supervisão, um número maior de tipos de *grasp* usando apenas 13 amostras de poses da mão ao

invés de todas as amostras.

Possível trabalho futuro está em um estudo formal para saber quais poses e características do movimento da mão são as mais discriminantes. Outra possibilidade é o desenvolvimento do reconhecimento e categorização dos movimentos do operador em um ambiente robótico, usando mais informações de outros sensores, como a informação de ângulos dos dedos e áreas de contatos. Há, também, trabalho futuro focado em melhorar a eficiência do algoritmo. Como também, um trabalho futuro focado em um reconhecimento em tempo real.

Referências

ALEOTTI, J. e CASELLI, S. Grasp recognition in virtual reality for robot pregrasp planning by demonstration. **Proceedings 2006 IEEE International Conference on Robotics and Automation 2006 ICRA 2006**, n. May, p. 2801-2806, doi:10.1109/ROBOT.2006.1642125, 2006.

BEDREGAL, B. C. e COSTA, C. R. Hand Gesture Recognition in an Interval Fuzzy Approach. v. 156, p. 2005-2005, 2005.

BERNARDIN, K. et al. A sensor fusion approach for recognizing continuous human grasping sequences using hidden Markov models. **IEEE Transactions on Robotics**, v. 21, n. 1, p. 47-57, doi:10.1109/TRO.2004.833816, 2005.

BIMBER, O. Continuous 6dof Gesture Recognition: A Fuzzy Logic Approach. **gesture**, v. 1, p. 1, 1999.

CALINON, S. e BILLARD, A. Incremental learning of gestures by imitation in a humanoid robot. **Proceeding of the ACM/IEEE international conference on Human-robot interaction - HRI '07**, p. 255, doi:10.1145/1228716.1228751, 2007.

CHEN, Y.-T. e TSENG, K.-T. Multiple-angle Hand Gesture Recognition by Fusing SVM Classifiers. **2007 IEEE International Conference on Automation Science and Engineering**, p. 527-530, doi:10.1109/COASE.2007.4341729, 2007.

CUTKOSKY, M. R. On grasp choice, grasp models, and the design of hands for manufacturing tasks. **IEEE Transactions on Robotics and Automation**, v. 5, n. 3, p. 269-279, doi:10.1109/70.34763, 1989.

DE KAMP, C. VAN e ZAAL, F. T. J. M. Prehension is really reaching and grasping. **Experimental Brain Research**, v. 182, n. 1, p. 27-34, doi:10.1007/s00221-007-0968-2, 2007.

EKVALL, S. e KRAGIC, D. Grasp Recognition for Programming by Demonstration. **Proceedings of the 2005 IEEE International Conference on Robotics and Automation**, p. 748-753, doi:10.1109/ROBOT.2005.1570207, 2005.

FEIX, T. et al. A comprehensive grasp taxonomy. In: **ROBOTICS, SCIENCE AND SYSTEMS: WORKSHOP ON UNDERSTANDING THE HUMAN HAND FOR**

ADVANCING ROBOTIC MANIPULATION. **Anais...** [S.l: s.n.]. Disponível em: <<http://grasp.xief.net/documents/abstract.pdf>>. Acesso em: 10 dez. 2011, 2009.

FELS, S. S. e HINTON, G. E. Glove-Talk: a neural network interface between a data-glove and a speech synthesizer. **IEEE Transactions on Neural Networks**, v. 4, n. 1, p. 2-8, 1993.

FELS, S. S. e HINTON, G. E. Glove-talk II - a neural-network interface which maps gestures to parallel formant speech synthesizer controls. **IEEE Transactions on Neural Networks**, v. 9, n. 1, p. 205-212, 1998.

GRAY, R. A. **The American Heritage Dictionary Of The English Language, 3D edition: A review. Reference Services Review.** [S.l: s.n.]. Disponível em: <<http://www.emeraldinsight.com/10.1108/eb049203>>, 1993.

HAYKIN, S. **Neural Networks: A Comprehensive Foundation.** [S.l.]: Prentice Hall, 1999. v. 13p. 409-412

HEUMER, G. et al. Grasp Recognition with Uncalibrated Data Gloves - A Comparison of Classification Methods. **2007 IEEE Virtual Reality Conference**, p. 19-26, doi:10.1109/VR.2007.352459, 2007.

HIRAI, K. et al. The Development of Honda Humanoid Robot. **Structure**, p. 1321-1326, 1998.

HOSHINO, K. e KAWABUCHI, I. Pinching with finger tips in humanoid robot hand. **ICAR '05. Proceedings., 12th International Conference on Advanced Robotics, 2005.**, n. 0, p. 705-712, doi:10.1109/.2005.1507486, 2005.

IWATA, H. e SUGANO, S. Design of human symbiotic robot TWENDY-ONE. **2009 IEEE International Conference on Robotics and Automation**, p. 580-586, doi:10.1109/ROBOT.2009.5152702, 2009.

JACOBSEN, S. C. et al. DESIGN OF THE UTAH/M.I.T. DEXTROUS HAND. **Design**, 1986.

JU, Z. et al. Dynamic grasp recognition using time clustering, gaussian mixture models and hidden markov models. **Intelligent Robotics and Applications**, p. 669-678, 2008.

KANEKO, K.; HARADA, K. e KANEHIRO, F. Development of Multi-fingered Hand for Life-size Humanoid Robots. **Proceedings 2007 IEEE International Conference on Robotics and Automation**, n. April, p. 913-920, doi:10.1109/ROBOT.2007.363102, 2007.

KANG, S. B. e IKEUCHI, K. Grasp Recognition Using The Contact Web. **Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems**, v. 1, p. 194-201, doi:10.1109/IROS.1992.587321, 1992.

KANG, S. B. e IKEUCHI, K. Toward automatic robot instruction from perception-mapping human grasps to manipulator grasps. **Robotics and Automation, IEEE**, v. 13, n. 1, p. 81–95, 1997.

KANGAS, J. Time-delayed self-organizing maps. **International Joint Conference on Neural Networks**, p. 331-336, doi:10.1109/IJCNN.1990.137735, 1990.

KANGAS, J. Phoneme recognition using time-dependent versions of self-organizing maps. **[Proceedings] ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing**, n. 1, p. 101-104 vol.1, doi:10.1109/ICASSP.1991.150288, 1991.

KLATZKY, R. L. e LEDERMAN, S. Intelligent exploration by the human hand. In: VENKATARAMAN, S.; IBERALL, T (Eds.). **IEEE**. [S.l.]: Springer-Verlag, 1990. p. 66- 81.

KOHONEN, T. Self-organized formation of topologically correct feature maps. **Biological Cybernetics**, v. 43, n. 1, p. 59-69, doi:10.1007/bf00337288, 1982.

KOHONEN, T. The self-organizing map. **Proceedings of the IEEE**, v. 78, n. 9, p. 1464–1480, 1990.

LI, C.; KHAN, L. e PRABHAKARAN, B. Real-time classification of variable length multi-attribute motions. **Knowledge and Information Systems**, v. 10, n. 2, p. 163-183, doi:10.1007/s10115-005-0223-8, 2005.

LIU, H. et al. Multisensory five-finger dexterous hand: The DLR/HIT Hand II. **2008 IEEE/RSJ International Conference on Intelligent Robots and Systems**, p. 3692-3697, doi:10.1109/IROS.2008.4650624, 2008.

MACKENZIE, C. L. e IBERALL, Thea. **The Grasping Hand**. [S.l.]: North-Holland, 1994. v. 104

MARTINEZ, D. e KRAGIC, D. Modeling and recognition of actions through motor primitives. **2008 IEEE International Conference on Robotics and Automation**, p. 1704-1709, doi:10.1109/ROBOT.2008.4543446, 2008.

NAPIER, J. R. The prehensile movements of the human hand. **Surger**, v. 38-B, n. 4, p. 902-13, 1956.

PALM, R. Grasp recognition by Time-clustering, Fuzzy Modeling, and Hidden Markov Models (HMM)-a comparative study. **Fuzzy Systems, 2008. FUZZ-IEEE 2008.(IEEE)**, p. 599-605, 2008.

PALM, R.; ILIEV, B. e KADMIRY, B. Recognition of human grasps by time-clustering and fuzzy modeling. **Robotics and Autonomous Systems**, v. 57, n. 5, p. 484-495, doi:10.1016/j.robot.2008.10.012, 2009.

PALM, R.; MEMBER, S. e ILIEV, B. Segmentation and Recognition of Human Grasps for Programming-by-Demonstration using Time-clustering and Fuzzy Modeling. **Simulation**, p. 1-6, 2007.

PONCE, J. et al. On Computing Four-Finger Equilibrium and Force-Closure Grasps of Polyhedral Objects. **The International Journal of Robotics Research**, v. 16, n. 1, p. 11-35, doi:10.1177/027836499701600102, 1997.

ROMERO, J., EK, C., e KRAGIC, D. Human Grasping Database. Acesso em 5 de dezembro de 2011, disponível em <http://grasp.xief.net/>, 2009.

SHADOW . (2010). Shadow Dexterous Hand: <http://www.shadowrobot.com/>

SCHLESINGER, G. Der Mechanische Aufbau der Kunstlichen Glieder. Ersatzglieder und Arbeitshilfen, part II. **Ersatzglieder und Arbeitshilfen für Kriegsbeschädigte und Unfallverletzte**, p. 321–661, 1919.

STERN, H.; WACHS, J. e EDAN, Y. Human factors for design of hand gesture Human-Machine Interaction. In: IEEE INTERNATIONAL CONFERENCE ON SYSTEMS MAN AND CYBERNETICS. **Anais...** [S.l: s.n.]. Disponível em: <http://web.ics.purdue.edu/~jpwachs/papers/smc06_human.pdf>, 2006.

VICENTE, I. S. et al. Action recognition and understanding through motor primitives. **Advanced Robotics**, v. 21, n. 15, p. 1687-1707, doi:10.1163/156855307782506156, 2007.

YOSHIKAWA, T. Multifingered robot hands: Control for grasping and manipulation. **Annual Reviews in Control**, v. 34, n. 2, p. 199-208, doi:10.1016/j.arcontrol.2010.09.001, 2010.

ZOLLNER, R. et al. Dynamic grasp recognition within the framework of programming by demonstration. **Proceedings 10th IEEE International Workshop on Robot and Human Interactive Communication. ROMAN 2001 (Cat. No.01TH8591)**, p. 418-423, doi:10.1109/ROMAN.2001.981940, 2001.