Continuous Grasp Recognition using Hidden Markov Models

Diplomarbeit

Ikeuchi Laboratory, Robotics Group
Institute of Industrial Science
The University of Tokyo

IRF Dillmann,
Fakultat für Informatik
an der
Universität Karlsruhe (TH)

Keni Bernardin

Tag der Ausgabe : 15. April 2002
Tag der Abgabe : 14. Oktober 2002

1. Betreuer an der Universität Karlsruhe : Prof. Rüdiger Dillmann
2. Betreuer an der Universität Karlsruhe : Prof. Alex Waibel
Betreuer an der Tokyo University : Prof. Katsushi Ikeuchi

Ich erkläre hiermit, die vorliegende Diplomarbeit selbständig verfasst zu haben. Die verwendeten Quellen und Hilfsmittel sind im Text kenntlich gemacht und im Literaturverzeichnis vollständig aufgeführt.

Tokyo, den 11.10.2002

Kurzfassung

Für das effiziente Programmieren von Servicerobotern wurde in den letzten Jahren eine neue Technik vorgestellt, das Programmieren durch Vormachen (PdV). Nach diesem neuen Konzept sollte ein Roboter in der Lage sein, die Ausführung einer neuen Handlung in einer ähnlichen Weise zu erlernen, wie es auch Menschen tun: Durch Beobachten einer Benutzervorführung und anschliessendes Herleiten einer abstrakten symbolischen Repräsentation dessen, was getan wurde. Wenn ein Mensch in einer solch natürlichen Weise mit einem Roboter kommunizieren kann, indem er Objekte manipuliert, Zeichen gibt oder durch Sprache Anweisungen erteilt, dann öffnet das die Tür zu völlig neuen Anwendungsgebieten.
Die Ausführung einer Handlung beinhaltet üblicherweise das Greifen oder sonstige Manipulieren von Objekten. Deswegen wurde viel Forschung ueber die Erkennung von menschlichen Handgriffen betrieben. Viele der Ansätze konzentrieren sich auf einfache Aufgaben, wie Pick & Place, d.h. sie erkennen nur den Zeitpunkt eines Griffes, die behandelten Objekte und ihre Lage. Andere beschränken sich auf Einzelgrifferkennung. Bis jetzt wurden Systeme, die in der Lage sind, ganze Sequenzen von Gesten zu klassifizieren, nur auf einem anderen Gebiet, mit unterschiedlichen Anforderungen, entworfen: die Erkennung von kommunikativen Gesten, wie Fingerdeuten, symbolische Kommandos oder Zeichensprachen. Sie können nicht ohne weiteres auf die Erkennung von Griffsequenzen angewandt werden, denn im Gegenteil zu kommunikativen Gesten, können Griffe meist nicht einfach auf Grund der Handform erkannt werden.

Hier wird ein System zur Erkennung von kontinuierlich ausgeführten Sequenzen von Griffbewegungen vorgestellt. Durch Verwendung von Hidden Markov Models erreicht es sowohl die Segmentierung der Benutzervorführung, d.h. die Erkennung der Zeitpunkte, an denen ein Objekt gegriffen und wieder losgelassen wird, als auch die Klassifizierung der benutzten Griffe in einem Schritt, mit einem soliden, statistisch fundierten Ansatz. Keine bedeutenden Einschränkungen des Handlungsflusses während der Vorführung, der Anzahl Benutzer, der betrachteten Objekte, oder der Arbeitsumgebung werden vorgenommen.
Eine Kombination von Eingabemodalitäten wird verwendet, um die Benutzerhandlung zu beobachten. Sowohl ein Cyberglove Datenhandschuh, als auch ein Satz kapazitiver Drucksensoren wird eingesetzt, um genaue Daten

über die Lage der Finger und ihre Kontaktpunkte mit gegriffenen Objekten zu erhalten. Die Drucksensoren erwiesen sich als besonders nützlich zur Verbesserung der Segmentierungsqualität, da sie erlauben, den genauen Start- und Endzeitpunkt eines Griffes zu ermitteln, selbst wenn keine klare greifende Bewegung mit den Fingern ausgeführt wird.
Das System wurde für die Erkennung der 14 Griffe aus Kamakuras Klassifikationstabelle entwickelt. Diese Taxonomie berücksichtigt den Zweck eines Griffs, die Form der Hand sowie seine Kontaktpunkte mit gegriffenen Objekten, und ist generell genug, um in den meisten Handhabungsaufgaben Anwendung zu finden. Eine grosse Auswahl an Objekten verschiedener Grössen und Formen, die in Alltagssituationen benutzt werden, wird berücksichtigt. Für jeden Grifftyp in der Tabelle wurde ein Hidden Markov Modell mit flacher Topologie erzeugt. Es wurde auch ein spezielles "Garbage" -Modell enwickelt, um unbeabsichtigte, störende Handbewegungen herauszufiltern. Die Hidden Markov Model Parameter wurden offline auf einem Trainingssatz von 112 Vorführungen angepasst, die von 4 Benutzern geliefert wurden. Dieselben Benutzer lieferten auch einen unabhängigen Testsatz, der für die Auswertung der Erkennungsrate benutzt wurde.

Die Ergebnisse zeigen, dass eine robuste Klassifizierung möglich ist, selbst für mehrere Benutzer, und mit Berücksichtigung einer grossen Vielfalt von Objekten. Das System kann sich an verrauschte Sensordaten anpassen, an grössere Unterschiede in der Handgeometrie der Benutzer, an unterschiedlich oder ungenau ausgeführte Griffe, und an verschiedene Ausführungsgeschwindigkeiten. Techniken, die bisher nur für die Erkennung von kommunikativen Gesten angewandt wurden, konnten erfolgreich auf die Erkennung von Griffen angepasst werden, und eine Erkennungsrate von 92,2% für ein Einzelbenutzersystem und 90,9% für ein Mehrbenutzersystem wurde erreicht. Das vorliegende System konzentriert sich nur auf die Art der Griffe und ihren Ausführungszeitpunkt. Es erkennt nicht die Objekte, die gegriffen werden, oder ihre Lage, liefert also nur begrenztes Wissen über die ausgeführte Handlung. Es eignet sich aber gut als Baustein für ein grösseres, allgemeineres PdV System.

Summary

For the efficient programming of personal or service robots, a new technique, Programming by Demonstration (PbD), has been proposed in recent years. Following this concept, a robot should be able to learn to execute a task much in the same way a human does: by simply observing a user demonstrate the task and inferring from this demonstration a high level, symbolic description of what has been done. If a human can communicate with a robot in such a natural way, using his hands to manipulate objects, making signs or using speech to give instructions, numerous new applications of robot systems become possible.
Executing a task mostly involves grasping or otherwise manipulating objects. That's why lots of research has been made on recognizing human hand grasps. Many of the existing approaches focus only on simple operations like Pick and Place, meaning they recognize only the time point of a grasp, the objects involved, and their placement. Others are limited to the classification of single hand gestures. Until now, systems designed to classify whole sequences of gestures are applied to a domain with different requirements: the recognition of communicative gestures, such as pointing motions, symbols, or sign languages. They cannot be directly applied to the recognition of grasp sequences because grasps, unlike communicative gestures, are generally not expressive enough to be classified based only on the hand shape.

Here, a system to recognize continuously executed sequences of grasping gestures is presented. Using Hidden Markov Models, it both segments the user's demonstration, detecting the moments in time where objects are grasped or released, and classifies the performed grasps in a single step, with a statistically sound approach. No significant restrictions are made on the flow of execution, the number of users, the objects involved, or the task environment. A combination of input modalities serves to capture the user demonstration. Both a Cyberglove and an array of pressure sensitive sensors are used to gain precise information about the shape of the hand and its contact points with grasped objects. The tactile sensors were found to be particularly useful in improving the quality of segmentation, as the starting and end points of grasps could be recognized with high accuracy even in the absence of clear grasping finger motion.
Recognition is performed for the 14 different grasps from Kamakura's classification table. The taxonomy focuses on the purpose of a grasp as well as the

hand shape and its contact points with objects and remains general enough to be used for most manipulation tasks. A large selection of objects of different shapes and sizes used in everyday life is considered. For every grasp class, a flat topology Hidden Markov Model was created. Also, a special garbage model with ergodic topology was designed to filter out unintentional non-gesture hand movement. The Hidden Markov Model parameters were trained offline on the sample demonstrations from 4 different users. Each user delivered 28 recordings for a total of 112 training demonstrations. The recognition accuracy was measured on an independent test set of equal size.

The achieved results show that a good classification can be obtained, even for multiple users and considering a great variety of objects. The designed system is able to robustly adapt to noisy sensor data, big changes in user hand geometry, variability in the way grasps are performed, or their imprecise execution, different grasping speeds, etc. The techniques used so far only for communicative gesture recognition could be succesfully adapted to the recognition of grasping gestures and an accuracy of up to 92.2% for a single user system, and 90.9% for a multiple user system could be achieved. The presented system focuses only on grasps and the time of their execution. It cannot recognize the grasped objects or track their positions, and therefore only provides limited knowledge on the performed task. But it is a useful building block that can easily be used in a more complete Programming by Demonstration system.

要約

家庭用のロボットもしくはサービスロボットに対する簡便な動作プログラミング方法として、「実演に基づく教示 (Programming by Demonstration)」が近年提案された。この方法では、観察によって作業に関する抽象度の高い記述を獲得し、教示動作と同じ作業を再現することの可能な能力がロボットに求められる。一方で、もし手で物体を操ってみせる、もしくは合図や言葉で指示を出すといった自然な方法でロボットに教示をすることが可能になれば、ロボットシステムの応用分野が飛躍的に広がることが期待される。
作業の実行には、把持、つまり物体を操る動作が必然的に伴う。そのため、人間の把持形態の認識に関して多くの研究がなされてきた。これらのうち多くの研究では、単純な pick and place 動作について把持の生じた時刻を認識することが行われ、また他方では、ある瞬間の片手のジェスチャを認識する研究も行われている。しかし、手の動作全体を解析するための手法については、これまで合図のためのジェスチャもしくは手話といった他の分野では研究されてきたものの、把持を手の形態のみから分類することはジェスチャとは異なり困難であるため、これらの方法を連続する把持動作の認識に直接適用することはできない。

そこで、本論文では連続する把持動作を認識するシステムを提案する。本システムでは、実演教示のセグメンテーション、つまり把持および解放の行われる時刻の検出と、各セグメントで実行された把持の種類の分類を、時系列データを統計的に妥当に処理する手法である隠れマルコフモデルを適用することによって実現する。
手の形状と把持物体との接触状態の情報は、サイバーグローブと分散触覚センサの組合わせによって獲得される。触覚センサはセグメンテーションの精度を向上する上で特に有用であり、明確な指の動きを伴わない把持においても、把持の開始時刻と終了時刻を正確に認識することが可能になる。
把持の認識は鎌倉による１４種類の把持分類規範に基づいて行われる。この規範は、把持の形態と物体との接触情報および把持の目的に即して構成されており、日常生活で使われる大きさや形状の異なる多くの物体が考慮されているため、とんどの日常の操り動作を分類することが可能である。個々の把持は、一方向隠れマルコフモデルによってモデル化される。また、全結合型隠れマルコフモデルであるガーベッジモデルを導入することによって、把持とは関係のない動作を除外することが可能になる。隠れマルコフモデルのパラメータは、異なる４人から採取された計１１２個（各２８個）の実演データを元に学習され、認識率は、学習データとは別の同数のデータセットを使用して評価された。

実験の結果、把持対象物体の形状や大きさが異なり、またユーザが複数であるにも関わらず、良い認識結果が得られた。ノイズの多いセンサデータ、ユーザ毎に異なる手の大きさ、異なる把持の順序・把持の速度・把持対象物体といった環境の違い対し、本システムが頑健に対処できることが確認された。従来は合図のためのジェスチャ認識にのみ適用されていた手法を、把持の認識にうまく適用した結果、単一ユーザで 92.2%、複数ユーザで 90.9% の認識率を達成することができた。実現したシステムでは、把持の種類とそれが生成された時刻の認識を行うが、把持物体の認識や把持物体の姿勢の推定はできない。そのため、教示された作業について全部の情報を提供することはできないが、より複雑な「実演に基づく教示」システムのための有用な構成要素となることが期待される。

Acknowledgements

The past eighteen months I spent studying in Japan have no doubt been one of the greatest experiences in my life, not only on the academic, but also on the personal, intellectual and psychologic points of view.

I'd like to thank all the people who made this possible, starting with my supervisor in Karlsruhe, Prof. Dillmann, who gave me the support and the freedom necessary to carry out such an unconventional study plan.

A very special thanks goes to Prof. Ikeuchi. His unconditional backing has helped me overcome many difficult problems that occured in the past year. It was his resourcefulness and determination that helped me at a critical time to find new sources of fundig and extend my stay in Japan long enough to accomplish the work on this thesis.

Many thanks go to the members of the robotic group. Hashi-Ken, Yoshi, Takamatsu-san, Ogawara-san, Morita, Sonoda. From the very start, they have made me feel at home in the Ikeuchi Lab. They have created a friendly, relaxed and very open atmosphere, such that can rarely be found anywhere. Thanks to them, I have had the chance to develop my knowledge in the japanese language and culture to an extent, that would definitely not have been possible otherwise.
I would also like to thank in particular Mr. Ogawara, our post-doc in the robotics group. He showed nearly infinite patience and always took the time to answer my questions or help me when a problem came up, even when he was under great pressure himself to finish three or four papers for a near deadline. I have greatly profited from his experience and advice.

Many thanks to my friend Tobias, who kept contact with me in the final phases of this work and helped me with some important questions about the layout, and to Ivica Rogina, my former supervisor at the Interactive Systems lab in Karlsruhe, for giving me some useful last-minute tips.
Thanks to my brothers, Tony, who unexpectedly provided me a very useful Latex interface that considerably sped up my working speed, and Jomo, who

made some last minute and very thorough proofreading.

A special thanks also to my parents, for teaching me the value of always srtiving for personal improvement.

Finally, many thanks to all those I have not mentioned explicitly, but who have helped me keep my spirits high in the last few difficult months, through their friendship, their support, their encouragement, or simply by heping me take my mind off work from time to time.

Abstract

A Hidden Markov Model based system is presented for the recognition of continuously, naturally executed grasp sequences, within the Programming by Demonstration framework. Existing systems concentrate on the recognition of communicative gestures, signs, or are limited to single gesture recognition. This system achieves high recognition rates for whole sequences of grasps in better than real time, while imposing as little restrictions as possible on the work environment, the types of objects grasped, user comfort and smoothness of execution, and with only little training.
To observe the grasping hand, Virtex Technology's Cyberglove is used in combination with an array of pressure sensitive sensors fixed on the finger and palm surfaces. The glove delivers finger joint angle measurements while the tactile sensors provide the system with information on the contact points of the hand with grasped objects.
Classification is made according to Kamakura's grasp taxonomy. It separates grasps into 14 different classes, according to their purpose, the hand shape and its contact points with grasped objects, and allows to distinguish all the various grasps used by humans in everyday life. Every grasp class is assigned a distinct HMM, the parameters of which are adjusted by Baum-Welch reestimation on a set of 112 training demonstrations. Recognition is then performed using the Viterbi algorithm on an equally large, independent set of test demonstrations.

The results show that a good classification can be obtained, even for multiple users and considering a great variety of objects. The designed system is able to robustly adapt to noisy sensor data, changes in user hand geometry, variability in the way grasps are performed, or their imprecise execution, different grasping speeds, etc. Through the efficient use of tactile information, a correct recognition of the beginning and end points of grasps in the sequence is made, even in the presence of noise or involuntary hand movement. An accuracy of up to 92.2% for a single user system, and 90.9% for a multiple user system could be achieved.

This work is supported in part by the Japan Science and Technology Corporation (JST) under the Ikeuchi CREST project, and in part by the Grant-in-Aid for Scientific Research on Priority Areas (C) 14019027 of the Ministry of Education, Culture, Sports, Science and Technology (Japan).

Contents

Chapter 1

Introduction

1.1 Programming by Demonstration

For the efficient programming of personal or service robots, a new technique, Programming by Demonstration (PbD), has been proposed in recent years. As opposed to textual programming or the recording and playback of user demonstrations, this technique would allow robots to operate in unstructured, unknown environments or communicate and interact with humans, without the need for time consuming and painstaking reprogramming by an expert, every time the task description changes.

Following this concept, a robot should be able to learn to execute a task much in the same way a human does: by simply observing a user demonstrate the task and inferring from this demonstration a high level, symbolic description of what has been done.

When executing complex tasks or operating in unstructured environments, simply repeating exactly the moves done by the demonstrator will not yield success, because the robot does not possess the same dexterity, the same manipulative degrees of freedom, because the conditions are not initially the same or change over time. The robot must rely on sensory information to understand its surroundings and adapt to changing conditions. Therefore, it must possess a series of skills, much the same way humans do, such as grasping an object, placing it on another object, avoiding obstacles, etc, and rely on these skills to execute the task [29]. Understanding a demonstration sequence would then, for the robot, mean recognizing what primitive skills were used at what time to achieve success.

1.2 Applications

One main advantage of the PbD technique is the simplicity and speed with which a robot system could be reprogrammed for a new task, without the need for a programming expert. Robots are already widely used for applications in a well known and controlled environment, such as welding, spray painting in factories, etc. But other areas such as the assembly of objects with possibly movable parts or cables, the disassembly of old engines or equipment, require the robot to possess a great degree of skill, such that conventional programming techniques either fail or result in unacceptably long development times.
This is where the PbD technique could be effectively used. If a factory robot, for example, could be taught to assemble a simple device from spare parts, just by showing it the required steps, a general purpose robot could be built and used for a number of different tasks without the need for a specialized design or special programming.

The classical bottleneck in human-robot communication has always been the interface. The need to use a keyboard or mouse and the difficulty of giving complicated commands through these has always been one of the reasons limiting the spread of robotic systems in everyday human environments. If a human could communicate with a robot in a natural way, such as signs, gestures and speech, as he would with other humans, it would open the door to numerous new applications.
One of them could be a service robot operating in a kitchen, a typically unpredictable, constantly changing environment. Ideally, the robot would have to recognize speech or gesture commands from the user, and be able to observe and mimic the handling of a multitude of quite different objects in its workspace.
Manipulator arms, such as required in this scenario, are especially hard to program. One of the main problems is to decide how to grasp an object, since the required grasp depends on many factors, such as the size, shape and weight of the object, its rigidity and its intended use. It is extremely difficult to automatically decide on the optimal grasp while taking into account all these factors. In the PbD framework, the robot profits from the knowledge of the human demonstrator: It simply observes which grasp was used for a given task and uses either the same one or a grasp with similar properties, according to its manipulator's capabilities. The idea is that the demonstrator can take

into account many more factors than the robot, with its limited knowledge, ever could.

Lots of research has therefore been made on recognizing human hand grasps, and the points in the demonstration sequence where they occur [24, 11, 19, 20]. Many of the approaches recognize only simple operations like Pick and Place, meaning they focus only on the time point of a grasp and on the actions performed with the grasped object [25]. Only few, like [19, 8] actually analyze the type of grasp used. In fact, most of the work on analyzing the hand shape is done for a domain with slightly different requirements: the recognition of communicative gestures, such as pointing motions, symbols, or sign languages [40]. Some of the techniques developed here offer the advantages of considering the dynamic information included in the gesture and allowing the recognition of continuous sequences of gestures. But they cannot be applied directly to the recognition of grasp sequences because unlike communicative gestures, which are generally expressive enough to be classified based only on the hand shape, manipulative gestures, grasps, can be quite different while exhibiting very similar shapes. In this thesis, a technique to exploit dynamic hand movement data and achieve segmentation and classification for continuous sequences of grasps is presented.

1.3 Approach

The field of gesture recognition presents many parallels to speech and handwriting recognition. In each of these domains, the main task is to recognize configurations, patterns that evolve with time. The main difference lies in the type of input and the way it is treated before it is fed to the pattern recognizer. Whereas the field of gesture recognition is still relatively new and many approaches still focus on recognizing static hand poses, the techniques developed for speech recognition are already advanced, allowing to handle continuously spoken sentences, coarticulation effects, speaker variability, a relatively large vocabulary and much more. The most successful speech recognition systems nowadays are statistical recognizers based on Hidden Markov Models (HMMs). This is because HMMs are particularly well suited for recognizing long sequences of multiple patterns without clear boundaries between conceptually distinct segments and offer a clear Bayesian approach for doing so.

Here, a Hidden Markov Model based system is presented for the recognition of continuously, naturally executed grasp sequences, within the framework of Programming by Demonstration. While systems capable of continuous recognition have already been developed for communicative gestures, signs, this has not yet been done for the domain of manipulative gestures, grasps, for which the requirements are different. The system aims to impose as little restrictions as possible on the work environment, the types of objects grasped, user comfort and smoothness of execution, to keep training time and effort low and to stay as close as possible to real-time recognition.

To capture the user demonstration, Virtex Technology's Cyberglove is used. This data glove provides information on the shape of the hand by measuring its finger joint angles. To provide the system also with information on the contact points of the hand with grasped objects, an array of tactile sensors has been fixed on the inner side of the glove. These sensors cover parts of the hand where detection of contact is essential for distinguishing between grasps that are hard to recognize by hand shape alone.

The chosen grasp classification table has been introduced by Kamakura in [17]. In addition to the purpose with which an object was grasped, it considers both the hand shape and the contact points with objects to separate the grasps used by humans in everyday life into 14 different classes. The aim is to make a classification that remains general enough to be used for most manipulation tasks. Every grasp class is assigned a distinct HMM and trained on user demonstrations using the Baum-Welch Algorithm. The trained HMMs are then used together with the Viterbi algorithm for recognition of independent test demonstration sets.

The achieved results show that a good classification can be obtained, even for multiple users and considering a great variety of objects. The designed system is able to robustly adapt to noisy sensor data, big changes in user hand geometry, variability in the way grasps are performed, or their imprecise execution, different grasping speeds, etc. The tactile sensors were found to play a key role in detecting the beginning and end points of a grasp in the presence of involuntary hand movement. While not all the objectives concerning freedom of execution, user comfort, and a natural task environment could be met, the techniques used so far only in communicative gesture recognition could be succesfully adapted to the recognition of grasping moves.

An accuracy of up to 92.2% for a single user system, and 90.9% for a multiple user system could be achieved. The presented system focuses only on grasps and the time of their execution, not on the recognition of objects and object positions also needed for a true understanding of the task. But it can very well serve as a module in a more general Programming by Demonstration system.

1.4 Outline

Following this introduction, chapter 2 first describes previous work done in the domain of hand gesture recognition. In Chapter 3, the main problems to be tackled in grasp recognition and reasons for using an HMM approach are presented. Chapter 4 explains the design of the grasp recognition system while Chapter 5 shows the experiments performed and analyzes the obtained results. Finally, Chapter 6 gives a summary and an outlook to future research.

Chapter 2

State of the art in hand gesture recognition

Lots of work has been done on recognizing the human hand. This is because after speech, the hands are perhaps the most expressive tools people use to communicate amongst themselves. Depending on the configuration of its fingers, its movement in space, its speed or orientation, the hand can convey a great variety of information quickly and efficiently. Hand gestures are therefore a very convenient tool for human-computer interaction. A variety of recognition approaches have been proposed. They can be distinguished according to the type of devices used to capture the hand data, the type of gestures considered (static or dynamic, isolated or continuous, communicative or manipulative) and the algorithms used for their recognition. Here, for simplicity, we will first split the approaches according to the observation devices into vision-based and glove-based approaches, and later focus on the recognition algorithms used.

2.1 Vision-based systems

Many researchers use vision-based techniques to recognize human hand gestures [35, 39, 19, 25, 7, 47]. The reason why they are so popular is that tracking the hand with a CCD camera, for example, frees the user from cumbersome interface devices such as gloves or pens, which are often attached by cables to the recognition hardware or otherwise impair the naturalness of the interaction. The ideal is to allow the user to communicate with the

computer just like with another human being, without any special preparation, interface device or environmental setup. They require that the hand first be spotted in the scene and then tracked. Depending on the application, its position, orientation and the configuration of its fingers, or other simpler features like edges, contours, etc are extracted for further processing.

One of the most cited contributions in this domain is the recognition scheme by Kang and Ikeuchi [19], which proposed the recognition of grasp gestures based on the contact points of the hand with an object. To detect these points, range and intensity images of the manipulating hand, obtained from a light-stripe rangefinder and a CCD camera, are taken and the position of the finger and palm segments is tracked by fitting the data to a hand model. Model based tracking is also made for the object and the fingers and palm are deemed to be in contact with the object if they are within a 5mm range of distance from it. The system relies on three basic assumptions, though: that the hand itself does not move during tracking, that the grasp gesture starts from an initial configuration and that there is no significant interphalangeal occlusion. It detects the positions of the fingers at the beginning of the gesture and continuously adapts the model in subsequent frames.
The approach uses the concept of virtual fingers, first introduced by Iberall et al. [15] to calculate a grasp cohesive index and the type of grasp is determined based on this index and by checking if the palm was involved or not. The grasp taxonomy itself is based on the number and spatial distribution of the effective contact points and distinguishes first between grasps with or without use of the palm, and then subdivides further according to the number of fingers, etc...
The classification algorithm is purely analytical and is based on the static part of the grasp, i.e. the shape of the hand at the moment when the fingers stabilize around the object. The correct calculation of virtual finger values is crucial for classification. Thus, high accuracy is required in tracking the finger segments. Moreover, the time point of the static grasp phase has to be determined with a separate algorithm. The detection of this time point is part of the segmentation problem, where a continuous user demonstration is separated into grasping phases, free movement, manipulation phases, etc. The authors themselves have proposed a segmentation technique in [20], using information from a data glove and tracker.

A technique to achieve segmentation using only visual information was pro-

posed by Kuniyoshi et al. in [25]. The authors note that, using vision, it is difficult to detect the typically small movements involved in grasping and releasing reliably, making a segmentation based solely on positional information impossible. Therefore, besides features such as hand and object position, others like silhouette differences, coplanar edges of objects, etc are also used, and the segmentation is made based on qualitative changes in the scene. For this, a world model consisting of the table top and the locations of the hand and other objects is being constantly updated.
The approach focuses on Pick and Place operations and does not analyze the shape of the hand when grasping the object. It uses snapshots of the scene taken at specific segmentation points to classify the observed motion into actions such as "approach", "fine motion", "depart", etc. Only the manipulation of simple building blocks was considered and a simple table construction operation using these blocks could be recognized in 3min. 50 sec.

Both approaches have in common that the workspace is relatively small and the camera is placed near the hand allowing more precise recognition of the hand action without significant outside disturbance or background noise. In a natural, noisy environment, or when the camera is placed further away from the user, finding the hand itself in the image, or special points on it can become a problem. Even in a controlled environment, the feature extraction algorithms can be quite complicated. To alleviate this, a few techniques have been proposed. One of them is the use of colored gloves.

In [7], Davis and Shah achieve the tracking of hand fingertips by using specially marked gloves. The system is designed to recognize 7 hand gestures used as commands for a computer system: Left, Right, Up, Down, Grab, Rotate, and Stop. The fingertip locations were extracted from the images by histogram segmentation. The hand is required to be in a fixed start position at the beginning of the gesture and the finger movement to be executed slowly until the end position. The start and end points of the finger tip trajectories are then analyzed to extract a set of fingertip vectors, which are compared to reference vectors for classification. A finite state machine is used to model four qualitatively distinct gestural phases: Initial phase, motion to gesture phase, gesture recognition phase and motion to initial phase.

Other researchers attempt to locate the hand in the camera image without such special gloves, using for example skin color based detection and tracking,

which has proven quite effective in recent years. Though an extraction of hand features using only color information is not sufficient for detailed analysis of finger configurations, it can serve as a starting point for more detailed algorithms. In [47], Wilson and Bobick present a system to recognize gestures executed by users in a common workplace environment without any kind of special interface device. Using a wide baseline stereo camera system collecting views from the top of the scene, and flesh tracking, the 3D position of the head and hands can be obtained at about 20Hz. The reason why a color space based analysis is applicable, in spite of illumination changes and differing skin tones, is that the human skin possesses a characteristic footprint that can be distinguished in the image. While skin color systems could theoretically allow to recognize the gestures of multiple users in the scene, this issue raises a number of other questions, such as occlusions and assignment of gestures to users, that are not too easily resolved.

2.2 Glove based systems

Compared to input devices such as a mouse or pen, instrumented gloves that measure the finger joint angles of the hand directly allow for input of more complicated commands quickly because of the much greater number of degrees of freedom. They also allow a much more natural interaction than the keyboard. Compared to vision-based techniques, data gloves offer the advantage that the information about the hand shape is measured directly and is not affected by its position or orientation. Furthermore, additional sensors can be attached on the glove, such as force or tactile sensors that can be useful for detecting grasps of real objects, or force feedback devices for reporting back grasps of virtual objects. Gloves are often used when recognizing manipulative gestures, because they do not suffer from the problem of occlusions that occur when grasping objects or passing behind them, while delivering relatively precise data. For communicative gestures, this advantage disappears, as one can in general expect a relatively occlusion-free view from the camera on the hand, and so they are often considered unnecessarily obstructive.

Still, to avoid the problems caused by changes in hand orientation, illumination, or to avoid the use of complex or computationally expensive tracking algorithms, some researchers also use data gloves for communicative gestures.

In [30], Nam and Wohn present a system to recognize command like gestures for interaction with virtual reality systems. They use a VPL Dataglove to measure the hand shape and a Polhemus magnetic tracker [34] to determine its position and orientation. In this work, the main focus point was on the recognition of movement primes, shapes drawn by the hand when moving, while its actual configuration remained fixed. The 3D positional data of the tracker is first fitted to a 2D plane before gestures, such as "put down", "zig zag" or "ball" are recognized.
Weissmann and Salomon [16], on the other hand, focus on the recognition of the hand posture itself. They use a Virtex Technologies Inc. Cyberglove to recognize 20 static gestures such as "index finger", "gun", etc. The classification is done by neural networks trained with the data of 5 different users. One of the problems of glove-based approaches is that the gloves fit differently on users with different hand geometry and can thus produce somewhat different sensor outputs for a same gesture. The results obtained here show that a good recognition can be achieved with gloves even for multiple users if a sufficiently robust algorithm is used. But the approach concentrates on isolated static gestures and cannot recognize continuously executed sequences.

The most useful application fields of data gloves, however, are in recognition of manipulative gestures or grasps [32, 23, 44, 20, 8]. In [23], Kawasaki et al. show an application using a data glove equipped with force feedback devices. It is intended to recognize Pick-and-Place operations of a human demonstrator effectuated in a virtual environment. The aim is automatic programming of a multi-fingered robot. The system is designed to recognize Pick-and-Place operations consisting of six segments: move, approach, grasp, translate, place and release. 5 parameters, based on the object and hand velocities, finger positions, speeds and fingertip virtual forces are calculated and their profiles used for the segmentation.
Similarly, Voyles and Khosla [44] use data from a Cyberglove and Polhemus tracker to segment Peg-in-Hole tasks. However, since their system is meant for real-world manipulation demonstrations, the gloves are equipped with special force sensing fingertips. Instead of speed or force profiles, an agent-based approach is used. Recognition agents for "touch" gestures, "hand motion", "force", and the "volume sweep rate", a measure defined by Kang and Ikeuchi in [20] to detect breakpoints, run in parallel. Their output is then used by interpretation agents for segmentation of the demonstration into primitives such as straight-line motion, guarded moves, gripping, etc.

Again, these approaches recognize the time point at which objects are grasped or ungrasped in the sequence, but they stop short of analyzing the type of grasp used, a valuable piece of information when teaching a robot to grasp objects of different sizes, shapes and functions.

A system to achieve both task segmentation and grasp classification is presented by Kang and Ikeuchi in [20]. The authors show a combined approach, using a glove-based system for segmentation before vision-based classification of the used grasp is made. They distinguish 3 phases in a grasp: The pregrasp, grasp and manipulation phases. To find the segmentation breakpoints between grasp phases and other hand movement, a measure, the volume sweep rate, the product of the fingertip polygon area and the hand speed is used. The input devices are a Cyberglove and a Polhemus tracker. Minima in the temporal profiles of the hand speed and volume sweep rate serve to detect the segmentation bounds. Once this is achieved, the classification into grasps is done vision-based according to the technique presented in [19]. Thus, this approach makes the complete recognition in two steps.

A similar path is followed by Dillmann et al. in [8, 12]. A Programming by Demonstration (PbD) System is designed, fusing both the information from a ceiling mounted stereo vision system and from a VPL data glove and Polhemus combination. The camera is used to track the objects and the hand, locate fingertips and touch points, while the data glove provides detailed information on the hand posture. As for Kang and Ikeuchi [20], the segmentation is done using only the data from the glove. After the breakpoints have been set, the segment identified as grasping segment can be analyzed by a hierarchical neural net which classifies the grasp type according to Cutkosky's taxonomy [5].
In [50], a method for fusing the glove data with that of force sensors is presented. For manipulation tasks, the recognition of contact between the hand and object is useful to effectively determine grasp and ungrasp breakpoints. Therefore, Force Sensing Resist (FSR) sensors are attached on the glove's fingertips. The obtained force profiles are used in combination with the finger pose and velocity profiles, and a search with respect to minima is done to find the breakpoints. The authors have shown that tactile sensors can well be used to achieve a robust segmentation. The system offers the advantage, that even in the event of heavy occlusion or failure of the camera system, a task segmentation and subsequent grasp classification can still be made,

based only on the glove data. The vision system simply increases the overall accuracy of the recognition.

2.3 Recognition techniques

The techniques employed for recognition of hand gestures differ greatly according to the requirements of the task and the types of gestures considered. While some researchers concentrate solely on the classification of static gestures, others go a step further to assure naturalness of execution and incorporate dynamic properties in their recognition routines. In tasks where a single user command or sign must be recognized, and the beginning and end of the gesture are approximately known, isolated gesture recognition is employed. For other complex tasks, like the recognition of sign language for example, advanced techniques that recognize continuous sequences of signs are required.

2.3.1 Static gesture recognition

Since only a feature vector representing the hand pose at a specific point in time must be classified, analytical approaches, EM-classifiers and neural network techniques can be applied.
The recognition system by Kang and Ikeuchi [19] uses an analytical method. Information about the contact points of the hand with the object, together with the derived grasp cohesive index was used to classify grasps into a taxonomy similar to that of Cutkosky [5]. Depending on palm contact, the grasp is first classified as volar or non-volar grasp. Then, considering the value of the grasp cohesive index, the number of fingertip contact points and the degree of thumb abduction, a progressive matching to more detailed subclasses is made.

With an EM-based technique, Wu and Huang, in [49] designed a classification system for 14 command gestures. The authors used preprocessed 2D hand images and both mathematical features extracted by PCA and physical features. Their approach is based on the combination of Multiple Discriminant Analysis with Expectation Maximization techniques to include great amounts of unlabeled data in the training of their classifier. They so alleviate a problem common in many statistical classification algorithms: the

chronic lack of training data. Their system has shown good results in recognizing the static hand gestures regardless of the orientation of the hand with respect to the camera.

A powerful tool for static pattern recognition is the neural network classifier. It has been used with success by Weissmann and Salomon [16] for the recognition of sign gestures such as "index finger", "gun", etc. and by Friedrich et al. [13] for grasp recognition. The latter work was used by Dillmann et al. in [8] for their PbD system. The network is able to classify grasps according to Cutkosky's taxonomy, taking into account only the finger flexion values delivered by a data glove. The authors note, however, that for certain similar grasps, such as the disc-shaped, spheroid and circular precision grasps, a distinction based only on static joint angle data is difficult, and propose the use of visual information to gain features about the shape of the grasped object, that could aid in the classification.
Neural network techniques are popular because they form their internal structure automatically, can classify raw sensor data and are robust in the presence of noise or incompleteness. They however have serious drawbacks such as the need for thousands of labeled examples, lengthy training times and the need to repeat training from the start when a new gesture is added ([46]).

2.3.2 Dynamic gesture recognition

Lately, more and more research is concentrating on recognizing dynamic gestures. These comprise not only gestures done by moving the hand in space, but also signs and grasps for which the static configuration of fingers is the main distinctive characteristic. This is because the movement of the fingers before reaching the static phase of the sign or grasp also contains useful information that can be used for classification. The techniques used range from Finite State Machines [7] over Support Vector Machines [51] to Hidden Markov Models [36, 38, 10, 4, 27, 26]. As opposed to static gesture recognition, the temporal pattern of a gesture is analyzed by considering features from multiple sequential time frames in the demonstration.

In [51], Zoellner et al. show a system for the recognition of dynamic grasps that occur during fine manipulations. Dynamic grasps are defined as operations in which the finger joints are changed while an object is grasped. 3 types are recognized: Screwing, twisting and insertion motions. For obser-

ving the user demonstration, a camera, a Polhemus device and a data glove with mounted force sensors are used. If a dynamic grasp occurs, it is classified using a Support Vector Machine (SVM), a statistical tool that recognizes patterns by classifying their feature vectors in a high order dimensional space using a small set of support vectors learned in through training on an example database. The advantage of the SVM over neural network techniques, for example, is that it requires less training data and shows good generalization performance. However, the length of a sequence that can be classified by a SVM is limited and a prior separation into segments is necessary. This was done here by analyzing finger position, speed and force profiles, and identifying the grasping phases.

A tool that has gained tremendous popularity in recent times in the domain of gesture analysis is the Hidden Markov Model. Its success in speech and handwriting recognition has prompted researchers to apply it also to gesture recognition. HMMs are well adapted to temporal pattern recognition because they allow for dynamic time warping (DTW) of the input sequence. They also have elegant and efficient algorithms for learning and recognition, such as the Baum-Welch algorithm and the Viterbi search algorithm. Also, they allow for recognition of continuous sequences of patterns, without the need for prior segmentation.

Bobick and Wilson [48], present a system for adaptive recognition of a simple "up"-"down" gesture from whole body color images of a demonstrator. They use a simple 3-state Markov Model comprising the states "rest", "down" and "up". Their work focuses less on the recognition procedure but more on the training of the Markov Model parameters and the selection of features in the images. While they do not use a traditional HMM approach, they have shown that with an online training approach, the problem arising when train and test conditions differ can be overcome.

Ehrenmann et al. [10] built a system based on HMMs to recognize dynamic gestures for directing a mobile robot. Using skin color segmentation, the hand is tracked and its trajectory filtered. A Hidden Markov Model is designed for each of 5 hand movement gestures to be recognized, and trained with 10 distinct examples. However, in recognition, a prior segmentation of the demonstration was made to find the start and end points of the gesture, and meaningless motion could not be efficiently filtered out.

Lee and Kim, in [27] designed a HMM-based system to recognize continuously executed sequences of gestures without prior detection of breakpoints. The segmentation is done automatically by the HMM recognizer and a set of 10 gestures used in browsing Power Point slides are classified. They point out the difficulty for HMM recognizers to represent non-gesture patterns and propose the use of a specially trained garbage model to threshold the output of the other gesture models. Their technique allows to correctly identify and segment out transitional non-gesture moves that occur between two consecutively executed gestures. Thus the command gestures could be spotted in a continuous sequence with an accuracy of 93.14%.

Starner and Pentland [40] deserve special mention for their research on recognition of the American sign language (ASL). The ASL is a good example of a complicated gesture recognition task, if the naturalness of execution is not to be limited. Both the hand posture and its movement have to be considered, sequences have to be properly segmented without requiring specific pauses between gestures, coarticulation effects, variability in user hand shapes and in the execution of a gesture have to be handled, a large vocabulary has to be considered. Starner and Pentland's system is designed to recognize 40 different signs: six personal pronouns, nine verbs, twenty nouns, and five adjectives. They were chosen so as to provide coherent sentences when used by a random generator.
Visual features of the hands are used for input. The user demonstrations are recorded by color video cameras mounted on the desk or on a cap worn by the user. The data from 494 demonstrations was recorded and both the Baum-Welch algorithm for training and the Viterbi search algorithm for testing were used. The authors report a word accuracy of 92% for the desk-mounted system and of 98% for the head-mounted version. The approach has shown that HMM-based techniques can handle a variety of problems occurring in the recognition of continuous hand gestures, even with a large vocabulary.

Until now, HMMs have been applied to the field of communicative gestures, but the achieved results encourage their application also in the domain of manipulative gestures, for example to recognize grasping movements of a user in a programming by demonstration system. As opposed to the now existing systems, that separate a manipulation task into segments and subsequently analyze the gesture in the grasping segment, a system could then be built that both spots and classifies the grasps in the manipulation sequence

Chapter 3

The problem of analyzing manipulation sequences

In this section, the main problem points to be overcome when recognizing continuously executed manipulative gestures are analyzed. The main objectives of our grasp recognition system are presented and differences to previous research are shown. For a correct recognition of grasps, the following questions have to be answered:

1. How do we capture the data used for recognition, i.e. what devices do we use to observe the hand of the demonstrator.

2. What kind of grasps do we wish to recognize and what kind of application are they useful for.

3. What algorithms and techniques do we use to find and identify the grasps in the manipulation sequence.

3.1 Observing human hand motions

When using video cameras to observe the human hand manipulate objects, a basic problem arises: The objects themselves occlude the hand. For meaningful manipulation applications, one can expect the environment to contain many different objects the user wishes to interact with. So when reaching for an object or moving it to another location, the hand can pass behind other objects and become temporarily invisible, depending on the location of the

camera. Furthermore, the grasped object itself can become an obstacle preventing the camera to detect features or details on the hand, such as fingertips or contours.

When Pick-and-Place operations of simple objects are considered, it may be enough to track the location of the hand and determine which object was grasped or released at what time. When repeating the task, the robot would use a standard grasping technique when the time comes to pick up an object. In applications that require the grasping of more complicated objects of different shapes and sizes, however, the grasp used by the robot has to be adapted to fit the attributes of the object. Also the purpose of the grasp plays an important role. For example the grasp we use for picking up a pen is different from the one we use to write with it or to point with it. A program that would automatically take into consideration the size, shape and weight of the object, its rigidity, the points at which it should be grasped (for example in the case of a coffee cup to prevent spilling or burning of the fingers) and the purpose with which it was grasped would be a welcome solution, but it is not realizable at the current state of knowledge.

The solution proposed by the Programming by Demonstration technique is to gain as much information as possible about the type of grasp used by the demonstrator and to use the same kind of grasp or at least a grasp with similar properties during repetition. Here the knowledge the human demonstrator has about the task is used to reduce the planning effort of the robot. Therefore, it is necessary to analyze the configuration of the demonstrator's fingers during the grasp, the contact points with the object or the forces exerted.
This is where the occlusion of hand parts becomes a problem. If too little visual features can be extracted by the camera, a correct classification of the grasp type may be impossible. Of course the careful placement of the camera, or the use of multiple cameras can alleviate this problem, but not eliminate it completely.

This brings up another drawback of the vision-based systems: The visual features obtained can be quite different, depending on the location of the camera. Starner and Pentland, in [41] already report a notable difference in recognition accuracy for their subset of the American sign language, depending on whether the camera is desktop-based or mounted on a cap worn by

the user. The position of the hands and their size in the image also plays a role and some effort must be spent on finding the hand and tracking it [47, 49, 10, 27].
Moreover, the extracted visual features are generally not invariant to the orientation of the hand with respect to the camera. This is why many systems require a fixed position, orientation or distance of the hand to the camera [19, 7] or at least assume that the hands will always be seen under a relatively constant angle [47, 10, 27, 41].
Lighting conditions and changing backgrounds can also affect the extraction of visual features, which is why some systems require a fixed background or the use of colored gloves [25, 7].

The data glove based techniques overcome all these problems since the information about the finger joint angles is read directly by a physical device from the hands. Occlusions are not possible, the position and orientation of the hand, illumination and background play no role. If a magnetic tracker is used, the hand position and orientation can also easily be determined. These systems do have their drawbacks, though. The magnetic tracker is generally not very precise and can be confused by metallic objects. This is why Dillmann et al. [8] have proposed to use it in conjunction with a camera system, which delivers quite accurate position information, but cannot cope with occlusion. Then, both systems work together to cancel out their respective weak points.

The data glove, on the other hand can impair the mobility and sensitivity of the user's hands and render fine manipulations difficult. Also, the cables used to attach the glove to the computer system can be heavy and cumbersome, or collide with other objects in the manipulation environment. As a whole, gloves tend to reduce the naturalness of execution, which is why many researchers use vision systems, despite the restrictions required, or the complicated algorithms to overcome them.

One more point to be considered in the recognition of manipulative gestures is the precision reached by vision systems in detecting finger positions. Kuniyoshi et al. [25] note that it is difficult to detect hand-object contact points using vision only. It can be very difficult to detect if 2 or 3 fingers are involved in a grasp, if the palm is in contact with the object, etc. To do this, the fingertips have to be detected and tracked with high precision, a precise

object model used and fitted even under possible occlusion and geometric calculations made to detect the contact points.

The problem is more severe if the hand is not at a fixed position or distance from the camera. When the hand is close to the camera, details may well be extracted, but if it has to be tracked and segmented out of a bigger scene, image resolution problems can also diminish the accuracy. While detection of contact points can be used in determining the grasp type [19], detection of contact itself is very useful for recognizing when a grasp starts and when it stops. This is why in segmenting grasp sequences, some researchers resort to tactile sensor information [50, 44, 51, 14]. With tactile sensors attached to the inner side of the hand, the information about contact points and even the force used in grasping can be obtained simply and directly, with the expense, of course, of adding more cables and reducing further the dexterity of the user and the naturalness of the interaction.

3.2 Choosing task independent grasps

In their review of hand recognition techniques [33], Pavlovic et al. propose a classification of hand/arm movements as follows (see figure 3.1): They are first divided into gestures and unintentional movement. The gestures themselves can have two modalities: communicative or manipulative. Communicative gestures have an inherent communicational purpose and are usually accompanied by speech in a natural environment. They are further subdivided into acts and symbols and those are further classified into mimetic or deictic acts, and referential or modalizing symbols. Manipulative gestures, on the other hand, are used to act on objects in an environment.

Most of the work in human-computer interaction focuses on symbolic gestures since they can often be represented by static hand postures. The problem is that there are a great number of possible symbolic gestures. In ASL for example, there are literally thousands. Also in most applications, the set of gestures to be recognized, and their associated meaning, is determined ad hoc [1, 2, 30, 49, 48, 10, 27]. No general set of gestures that could be used in many applications is defined. This is due to the communicative nature of the interaction. The gestures we use when controlling a Power Point slide presentation [27] differ a lot from those used for musical conducting [48].

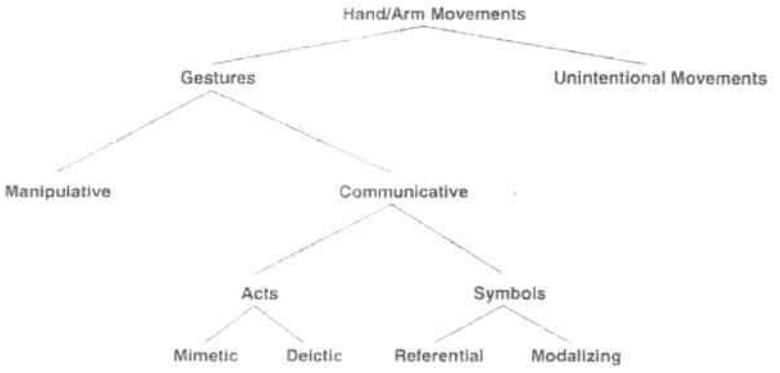

Figure 3.1: A general classification tree for hand/arm movements.

But for teaching of manipulation tasks to robots, the requirements differ somewhat. Not the precise shape of the gesture or its temporal evolution, but the purpose of the gesture is most important. When holding up a small coin for example, the position of the middle, ring and index fingers may be completely unimportant. The main point is that the object is very small and needs to be carefully manipulated. That is why a grasp is chosen, where only the thumb and index finger are in contact with it. This is even more evident, when one considers that the majority of robot hands are different from human ones, have a different number of fingers, etc. Even today's humanoid hands do not reach the level of sensitivity or dexterity of their human counterparts. Nevertheless, for some tasks, they may suffice: A primitive but well designed robot gripper could hold a closed book or a block with almost the same stability as a human hand could.

So the objective here should be to make a more general distinction into grasp classes that would reflect the type of grasped object and its use. Of course, in everyday life, we use a variety of purely playful grasps also. For example when holding up a pen and letting it revolve around our fingers. Considering this, the amount and variability of grasps to be classified seems huge. However, if we consider only grasps that are purposely used for manipulation tasks, they can be separated into a reasonable number of classes. For the pen example, we use a grasp type to pick it up from a table, one to write with it, one to

point with it, with very little variation.
Thus, a relatively concise grasp taxonomy can be decided on. The defined grasp classes should be general enough to span through many different tasks. As the objects to be manipulated and their purpose vary greatly, choosing a small set of task specific or arbitrary grasps would limit the use of the system.

Most of the research on manipulation sequences nowadays focuses on simple operations like Pick-and-Place, where the type of grasp is not considered at all. These systems are limited to experimental setups or very limited tasks with simple objects [25, 23, 44]. A notable exception is the system developed by Friedrich [13] which performs a neural net classification according to the taxonomy presented by Cutkosky in [5].

According to this taxonomy, which is based on previous work done by Napier [31], grasps are first divided into power grasps and precision grasps. Power grasps are those where the palm of the hand is involved in the grasp to allow for maximum stability, whereas in precision grasps, only the fingers are in contact with the object to allow for greater mobility. The grasps are then further subdivided into cylindrical and spherical grasps, according to the shape of the grasped object. Power grasps are then further classified according to the object size, the position of the thumb, and precision grasps according to the number of fingers involved. A total of 16 grasps is considered.

While the taxonomy has the merit of separating the grasps considering the shape and purpose of the objects, it has a few limitations. First of all, it is limited to manufacturing tasks, and to circular and prismatic grips. A number of grasps used in everyday life, such as when holding a spoon or a plate, or when taking a book out of a shelf are not included. Secondly, the separation done below power and precision grasps sometimes requires special a priori knowledge of the task requirements, such as additional information on the object itself, for classification. For example the disc and spheroid precision grips do not differ either in the hand shape or in the contact points with the object. Sometimes, the classification appears too detailed. The separation between thumb -4 finger or -3 finger prismatic precision grips, for example, seems irrelevant when considering the purpose of the grasp. Since many manipulator arms do not have the same amount of fingers as the human hand,

the grasp is not directly transmittable to the robot architecture and recognizing the number of fingers involved without identifying a new meaning to the grasp does not bear any advantage.

Another grasp taxonomy was presented by Kang and Ikeuchi in [19]. It is based on the effective contact points of the hand with the object. By matching the contact points to virtual fingers and calculating a grasp cohesive index, it also allows to distinguish grasps into a more abstract classification table, roughly similar to that of Cutkosky. This table first distinguishes between grasps with or without palm contact an then further subdivides according to the number of contact points and their spatial distribution. In [21], the authors show how to map these grasps to robot grasps. As opposed to Cutkosky's table, no a priori information about the object is required. But it suffers the same drawbacks concerning the application domain and for precision (non-volar) grasps, focuses more on the number of contact points and their configuration than on the grasp purpose.

For the recognition of grasping gestures occurring in everyday manipulation, we would like a classification of grasps that covers as many applications and task domains as possible. To remain as concise as possible and to be transmittable to many robot manipulator architectures, it should abstract as much as possible from the exact hand shape and number of contact points involved and focus on the utility of the grasp relative to the objects and their use.

3.3 Analyzing continuous sequences

Just as important as the chosen grasp types are the algorithms used to recognize them. To allow a natural interaction with the robot system, user demonstrations of various lengths containing multiple grasping and ungrasping actions should be recognized. Some of the research on hand gestures concentrates on isolated gesture recognition [47, 16, 49]. For the recognition of continuous sequences, two main problems have to be solved:

1. To find a meaningful temporal partition of the sequence into gestures and non-gesture movement, and

2. to determine the class membership of the contained gestures.

The solution of the second problem presupposes that the parts of the demonstration sequence that can be assigned to one gesture have been identified. A gesture, be it manipulative or communicative is comprised of three phases [28]. They are described as preparation, nucleus (peak or stroke), and retraction. The preparation phase consists of a preparatory movement that sets the hand in motion from some resting position. The nucleus of a gesture has some definite form and enhanced dynamic qualities. Finally, the hand either returns to the rest position or repositions for a new gesture phase. Grasps are dynamic gestures as well, consisting of a pregrasp, a grasp and a release phase. For static gesture recognition, only one of the frames of the demonstration sequence belonging to the nucleus (grasp phase) is considered. Dynamic gesture recognition techniques make use of many frames, including those contained in the preparation (pregrasp) phase. To make the classification, the representative feature vectors are mapped to a class based on their minimum distance to class representatives. These are either given ad hoc or determined through some learning algorithm, like averaging, K-means, or Hidden Markov Models. Thus, if the boundaries for a grasp are not well set, feature vectors from another grasp or from non-gesture movement may be wrongfully used.

So the solution of the temporal partitioning problem has a big influence on the outcome of the classification. Some of the proposed methods make restrictions such as requiring the user to make a pause between gestures [10] or to start or stop with the hand in a specific position [7]. Often, the segmentation is made using heuristics such as performing minimum search or applying thresholds on hand, finger speed or contact force profiles [20, 23, 8, 51].

In some cases arising in natural manipulation, such as transiting from one grasp to another without releasing the object, these approaches can not be applied. For example, when picking up a small hammer from the table top, we may take it using only our fingertips, then wrap our hand around it to hold it tightly, without releasing it. In such a case, a method that uses contact force profiles would have difficulties to segment the action into distinct grasps.
Since many approaches rely on the result of the segmentation to do the subsequent classification of gestures, they will fail to recognize this kind of action correctly. For methods that do static gesture analysis within the grasp segment [20, 8], using the start point or end point would result in classifying only one of the two grasps performed. Taking an intermediate value or the mean

of the hand configurations included in the segment would yield unpredictable results. Even approaches that attempt dynamic grasp analysis like [7, 51, 10] would fail to recognize two grasps and try to fit the information from both grasps to one class. To overcome this problem, we would like an algorithm that performs both segmentation and classification simultaneously.

Hidden Markov Models are quite well adapted to the task. Their states can easily be associated with the temporal gesture phases and their topology can be adjusted to suit the complexity of the task. The recognition procedure uses dynamic time warping (DTW) and finds both the optimum classes and their optimum boundaries to fit a set of feature vectors from the demonstration sequence. They also allow the use of a task grammar to reduce the amount of possible sequences that the user may execute, constraining the recognition problem and increasing recognition accuracy. So far, HMMs have been used for communicative gesture recognition with notable success [47, 36, 30, 10, 27, 40].

3.4 Goals

We would like to apply the same advanced techniques that have been used so far for communicative gestures to the recognition of manipulative gestures, grasps, as well. While a few systems exist that analyze and classify grasping hand shapes, they are limited to single gesture recognition. A unified approach that spots grasping phases in a demonstration sequence and classifies them in one step has not yet been proposed. The system should be designed to recognize and classify grasps occurring in everyday manipulation tasks, without significant restrictions on the flow of execution, the users, the objects involved, or the workspace. It should meet the following requirements:

- Allow continuous, natural movement of the hands. The user should not be required to start or stop in a fixed position, to make specific pauses for segmentation, or to hold the hand at a specific angle when grasping objects. He should be able to move at natural speeds, without having to wait for slow recognition hardware. This is directly influenced by the type of hardware used for observation of the hand movement and interaction, the type of features extracted and above all, by the algorithm employed to segment the demonstration.

- Retain task independence. The system should be able to recognize grasps used in all kinds of situations of everyday life, with objects of different sizes, shapes and uses. A careful selection of grasps that are general enough to be used across tasks, yet specific enough to convey enough information for a successful later repetition by the robot system is necessary. The chosen grasp taxonomy should focus on the type of object grasped and its intended use and the grasp classes should be distinguishable with the chosen observation devices and input features.

- Allow multiple users. If the system is limited to one user (the one providing the training samples for example), its application is greatly limited. Demonstrations from different users should be recognized with only little or no extra effort for adapting the system parameters to the new conditions. This implies a robust algorithm that can cope with variations in the user hand geometry or small differences in the way the grasps are executed and that can be adjusted at a later time with examples provided by new users.

- Allow a natural manipulation environment. In manipulation tasks, the user interacts with many objects that can become obstacles for the recognition. The system should function in a relatively noisy, cluttered workspace, impose no specific setup like a fixed background, clean table top or small number of objects. This again is directly related to the type of input features used.

- Stay as close as possible to real-time recognition. If the system is to remain user friendly, the result of the recognition should be available after a short period of time, preferably during or immediately at the end of the demonstration, so errors could be found and if need be, the demonstration repeated without significant loss of time.

Chapter 4

The recognition of continuous grasp sequences using Hidden Markov Models

Here, an HMM-based system is presented for the recognition of continuously, naturally executed sequences of grasps within the framework of Programming by Demonstration. In order to allow a natural working environment and avoid the problems common in vision-based systems, such as occlusion, hand tracking, dependency of the visual features on hand orientation, etc, a glove-based approach is chosen. Furthermore, to gain information about the hand-object contact points useful for task segmentation, grasp detection and analysis, an array of tactile sensors is attached to the glove's surface. A task independent grasp classification table is chosen that allows to work with all kinds of objects of different shapes and sizes, a Hidden Markov Model is built and trained for each of the 14 grasps of the table and recognition is performed using the Viterbi algorithm, taking advantage of its Dynamic Time Warping capabilities and robustness.

4.1 Hidden Markov Model Recognition

Just like speech, gesture recognition is about analyzing temporal sequences of patterns. The domains are closely related and share a number of common problems. In speech we may sometimes stutter or swallow word parts; when grasping we can break off a started grasp, or rearrange the fingers to a better,

more stable position. Just as in speaking, we sometimes produce sounds, like "err", "hum", etc that are not words, when manipulating objects, our hand may also make unwanted moves in between actions. Problems arising from different speaking speeds or styles, made more acute when multiple users are to be considered, also find their equivalent in the gesture domain.

Of course, compared to speech, in gesture recognition the main complexity of the problem lies in a different area. While speech recognizers have to battle with a huge vocabulary containing, depending on their application domain, tens of thousands of words, complicated concatenation rules, grammatical or not, and with the immense search space complexity this induces, gesture recognizers (with the exception of sign language recognizers) usually have a much smaller set of patterns to recognize. The main problem here is the input dimensionality as our hands evolve in a complex three dimensional space with plenty of objects they may interact with and where relative orientation and position play a big role. Concepts such as pointing directions, trajectories, distances to other objects, may be used to describe, on a higher level, what has been done. It is hard to determine in advance what information is generally needed, what features should be extracted from the manipulation scene to understand it.

Still, there are enough common points to justify the application of techniques that have been developed and refined over the years in speech recognition to the field of grasp recognition. Hidden Markov Models have established themselves as the tool of choice in speech recognition for some years already as they allowed a great leap in recognition of continuous, natural language. This is why, in the following, a brief overview is given of Hidden Markov Models and their training and evaluation techniques. A more detailed description can be found in [37, 45, 9].

4.1.1 Description

Hidden Markov Models are based on the assumption that the process being modeled can be described as a first order Markov process. This is a process in which the system possesses a set of discrete states between which it can be expected to switch from time to time. The system's parameters at any given time are described by the state in which it currently resides. So, when the system changes in time, its parameters might change and a different state

might be more appropriate to describe it. The process of changing states itself is stochastic. The Markovian property states that for a first order Markov process, the probabilities of transitions between states depend only on the current state. For a second order Markov process, the transitions between states depend only on the last two states, etc.

The difference between a Hidden Markov Model and a first order Markov process, is that in the HMM framework, the current state of the system is not observable. Instead, at each time step the system outputs a symbol, generated stochastically by whichever (unobservable) state in which the system currently resides. This means there is a stochastic process determining the state in which the system is in, and another stochastic process, characteristic of the current state, determining which observable symbol the system ouputs. Such a system is called doubly stochastic.

A Hidden Markov Model with n states can be completely described by the following quantities:

- the initial probablilities, π: a vector of dimension n where π_i is the probability of starting in state i
- the transition probabilities, A: an $n \times n$ matrix where a_{ij} is the probability of jumping from state i to state j
- output probabilities, B: HMMs can be either discrete or continuous. The output of discrete HMMs is one of m possible symbols. In this case, B is an $n \times m$ matrix and b_{jk} describes the probability of state j outputting symbol k. In the continuous case, the observable symbol output by the system is a continuous random vector. B now describes parameters for a set of probability density functions (typically a mixture of Gaussians) which give probabilities for different observable vectors.

Thus, an HMM can be fully described by $\lambda = (\pi, A, B)$.

4.1.2 Topology

To use HMMs as a recognition tool, several steps must be taken. First, a topology should be chosen for each HMM. In general, knowledge of the physical

properties of the system modeled can help to select an appropriate topology. For example, imagine an HMM to model the weather for the days over the year, specifically if it is sunny or rainy. In summer, there should be some more sunny days than in spring, and there should be a lot of rainy days in autumn. The HMM to model this behavior may consist of four sates (one for each season), and every state would have a quite different output probability function.

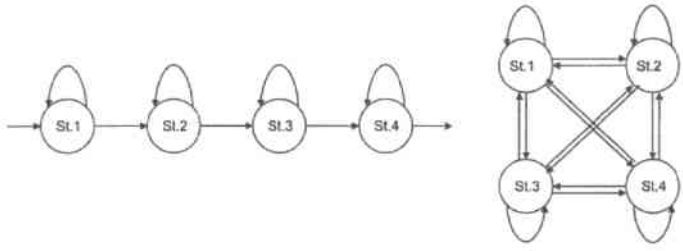

Figure 4.1: Example Hidden Markov Model topologies. Left, a 4-state HMM with flat topology. Right, a 4-state HMM with ergodic topology.

Of course, in practice we rarely consider such simple systems. Sometimes it is not easy to make a clear distinction of the states and their transitions. That's why in general, one starts with a topology which is suspected to be more complicated (i.e. it has more states and more paths between the states) than the system to be modeled. While training the model, it is then possible to prune the topology by removing states and links which are seldom used. Attempts to build the HMM topology automatically based on training evidence by gradually increasing the number of states have been made in [3].

4.1.3 Feature vectors

Another question to be answered when designing an HMM system is the choice of what to use as the observation vector. Consider the example of the weather again. The observation vector is the sky condition on a day. In our simplified case, the HMM would be a discrete system. The observations are simply one of two choices: sunny or rainy. The state (the season of the year) remains hidden because there is no directly observable quantity which immediately specifies it. We can only deduce the state by analyzing the pattern of

an observation sequence in time. For example, we could observe the number of rainy and sunny days over a given period, and then compare this to our knowledge from previous years to try to discern what season we are in today. This step is exactly what the HMM recognition system will eventually do: determine the most likely state by analyzing the observation sequence. Of course, the final result is not guaranteed to be accurate, for we cannot be sure the system switched, for example, from spring to summer at exactly the right day.

The choice of observation vectors when implementing real systems is affected by issues such as available sensors, desired invariances (e.g. invariance to hand rotation, user independence, etc.), and the amount of available training data.

4.1.4 Training

The next task to accomplish is training the system, using example data to learn appropriate transition and output probabilities. The goal is to take an observation sequence known to have come from a specific model and change this model's parameter set λ such that the probability that the given model produced the observation sequence is maximized. In general, the Baum-Welch algorithm is used to accomplish this task. This algorithm is an iterative re-estimation routine, guaranteed to find a local maximum of the probability. While the probability surface is likely to be quite complex, experience has shown that the Baum-Welch algorithm can quickly and effectively arrive at adequate models.

4.1.5 Recognition

Finally, once models have been trained for all of the patterns to be identified (i.e. words in a speech recognition system, grasps in a grasp recognition system), the Viterbi algorithm is used to perform recognition. The Viterbi algorithm is based on dynamic programming techniques and bears close resemblance to dynamic time warping. The task of recognition in the HMM framework is to take a given observation sequence and determine which of the HMMs was most likely to have emitted it.

The procedure works by maintaining a lattice structure of probabilities. Each column in the lattice, δ_i, has n nodes, each of which represents the probability

of being in a given state. There are as many columns in the lattice as there are observations in the sequence. The lattice gets filled recursively, starting with the first observation. The initial nodes are given a probability of :

$$\delta_i^1 = \pi_i b_i(O_1)$$

Then, nodes at time t are filled in with:

$$\delta_j^t = max_i[\delta_i^{t-1} a_{ij}] b_j(O_t)$$

The lattice gets filled in until the last observation, at which point the node with the maximum final probability is chosen, and the sequence can be recovered by backtracking through the lattice.

Of course, the described procedure is for isolated recognition only. To recognize sequences containing many patterns, a few adjustments have to be made. Instead of just one simple HMM, a composite HMM is constructed by taking the models of all the patterns (words, grasps) that could occur in the sequence and allowing transitions from the end state of one model to the starting states of all the others. If there is knowledge on what combinations of patterns are allowed, for example in form of a task grammar, these transitions can also be weighted. Then, the composite HMM is used to recognize the entire sequence.

4.2 Grasp taxonomy

Now let us turn our attention to the main object of the recognition: The grasps themselves. As explained in the previous chapter, the types of grasps we wish to recognize greatly influence the buildup of our system. To stay relatively task-independent and allow for grasping of all kinds of objects, the taxonomy presented by Kamakura in [17] has been adopted (although shorter and older, an English version can be found in [18]. It aims to divide all grasps used by humans in everyday life manipulations into a small set of representative classes, based on the purpose of the grasp, the shape of the hand, and the contact surfaces of the fingers or palm with the grasped objects.

It distinguishes 14 types of grasps divided into 4 main categories: 5 power grasps, 4 mid-power-precision grasps, 4 precision grasps, and one thumbless grasp. See Table 4.1.

Table 4.1: Grasp Taxonomy by Kamakura

Category	Class	Notation
Power Grips	Power Grip-Standard Type	PoS
	Power Grip-Hook Type	PoH
	Power Grip-Index Extension Type	PoI
	Power Grip-Extension Type	PoE
	Power Grip-Distal Type	PoD
Mid-Power-Precision Grips	Lateral Grip	Lat
	Tripod Grip-Standard Type	Tpd
	Tripod Grip-Variation I	TVI
	Tripod Grip-Variation II	TVII
Precision Grips	Parallel Mild Flexion Grip	PMF
	Circular Mild Flexion Grip	CMF
	Tip Grip	Tip
	Parallel Extension Grip	PE
Thumbless Grips	Adduction Grip	Add

As one can see, the initial separation into power grasps and precision grasps that has been made by Napier [31] and adopted by Cutkosky [5] and Kang [22] is also made here. This reflects the most basic distinction humans make when grasping an object. If the object is heavy, if great forces are to be exerted on it during manipulation, or if the focus is to keep it as stable as possible in the hand, a power grasp is chosen, and the palm of the hand is used to increase the contact surface with the object. This however limits the mobility of the object which is kept firm in the hand and all movement comes essentially out of the wrist.

If the focus is to be put more on fine manipulation a precision grasp is chosen. This occurs when picking up small objects like coins or needles, when taking

objects from difficult to grasp positions, such as picking up a spoon from a table or taking a book out of a shelf. Precision grasps are also used if fast and precise moves have to be executed with objects, such as when unscrewing a bottle cap. Only the fingers are in contact with the object, often only the fingertips.

Between these two main categories of power and precision grasps, an intermediate category is distinguished whose focus lies somewhere between power and dexterity. For the grasps in this category, the sides of the fingers are used instead of the palm, to better secure the object which is grasped by the fingertips. Here, we find the tripod (pen-) grip and the lateral pinch grip, also distinguished by Cutkosky and Kang. Such a grip is useful for example when turning a key in a lock, as it is too small to be held firmly using the palm, and the fingertips alone are not strong enough to make it turn.

In the last category, we find just one grip which is very often used for holding pens (or cigarettes). It differs from all the others in that the thumb is not used to hold the object.

In the following, a short explanation of the grasps in Kamakura's taxonomy is given, by stressing their differences with respect to their purpose and the object shape.

1. The Power Grips:

   The Power Grip Standard Type (PoS). It is one of the most stable grips and is commonly used. The fingers are well wrapped around the object and almost all the inside of the hand is in contact with it. It offers good stability in all directions. It is used when holding a hammer, an umbrella, a frying pan, etc.

   The Power Grip Hook Type (PoH). It differs from the PoS in that the hand is somewhat more open, the stress being put on countering pulling forces from the direction of the fingers. It is used when pulling on a lever, carrying a suitcase, etc.

   The Power Grip Index Extension Type (PoI). In this power grip, the

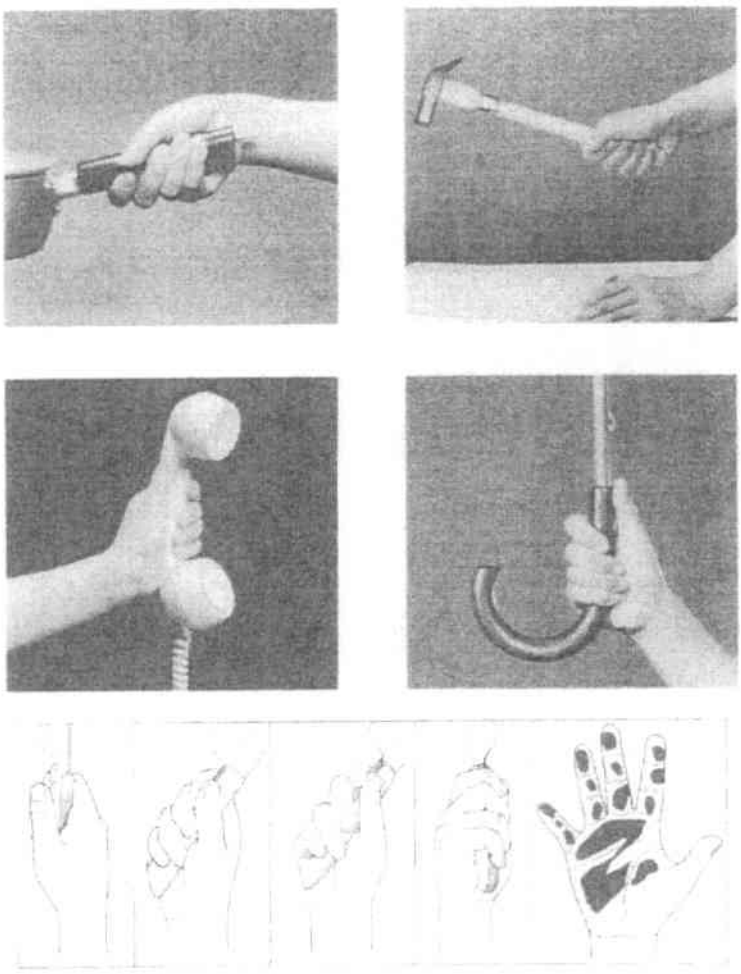

Figure 4.2: The Power Grip Standard Type (PoS)

index finger is extended and touches the extremity of the object. It is used when great forces are expected to act in one direction on the tip of the object and the finger serves as extra stabilizing support. It is used when picking with a fork, cutting with a knife, etc.

The Power Grip Extension Type (PoE). This grip is used when the object has to be held stably, firmly, but is too flat to allow the fingers

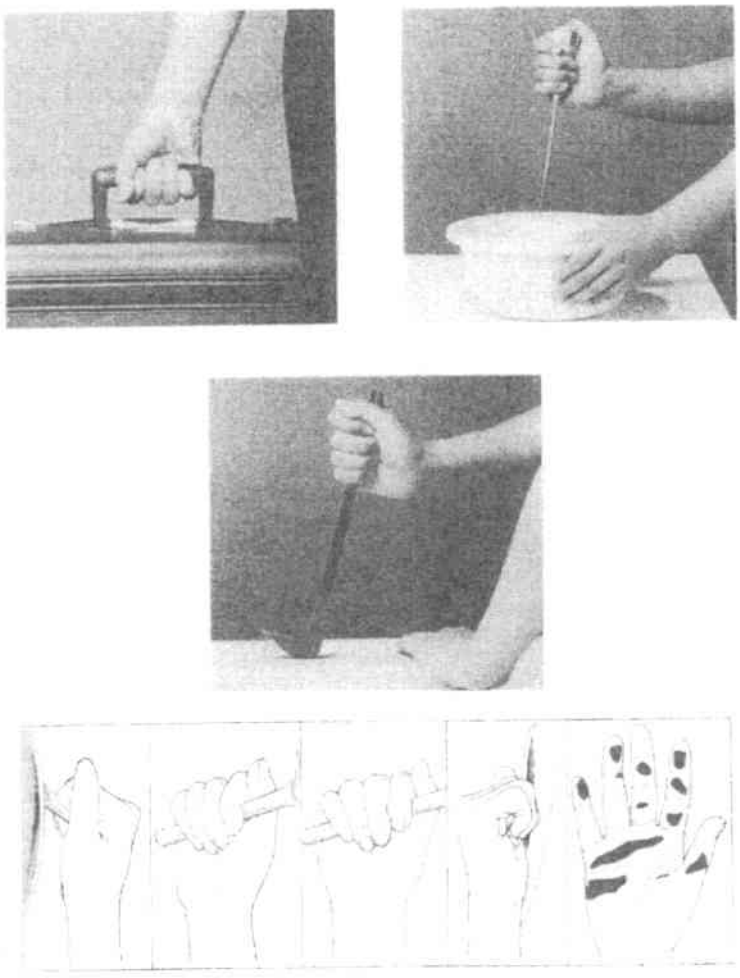

Figure 4.3: The Power Grip Hook Type (PoH)

to wrap around it completely. The edge of the object rests in the palm of the hand and the fingers are bent as much as possible to exert pressure. The PoE is used when carrying a plate, securing a large bowl on a table, etc.

The Power Grip Distal Type (PoD). This is an exception in the power grasp category, in that the palm is not used for grasping. This occurs

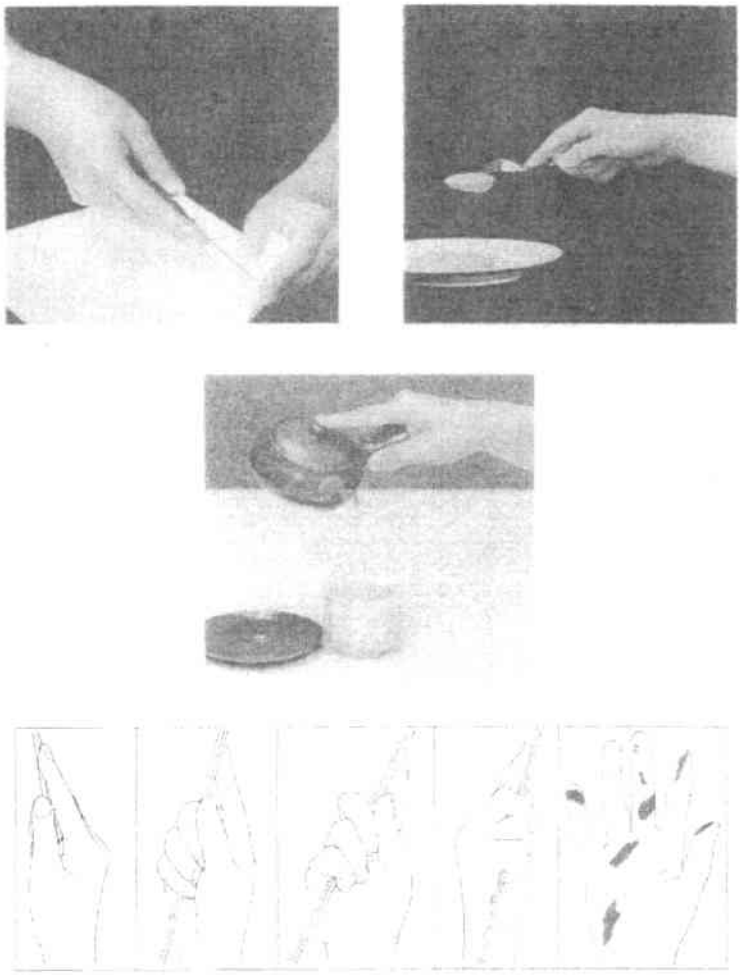

Figure 4.4: The Power Grip Index Extension Type (PoI)

when the object is small compared to the hand and can be fully wrapped in by the fingers alone. It is also used for big objects whose contact surface (gripper, handle) is small and can help put the object better into the main axis of the arm. It is used for toothpicks, nail clippers, scissors, etc.

2. The Mid-Power-Precision Grips:

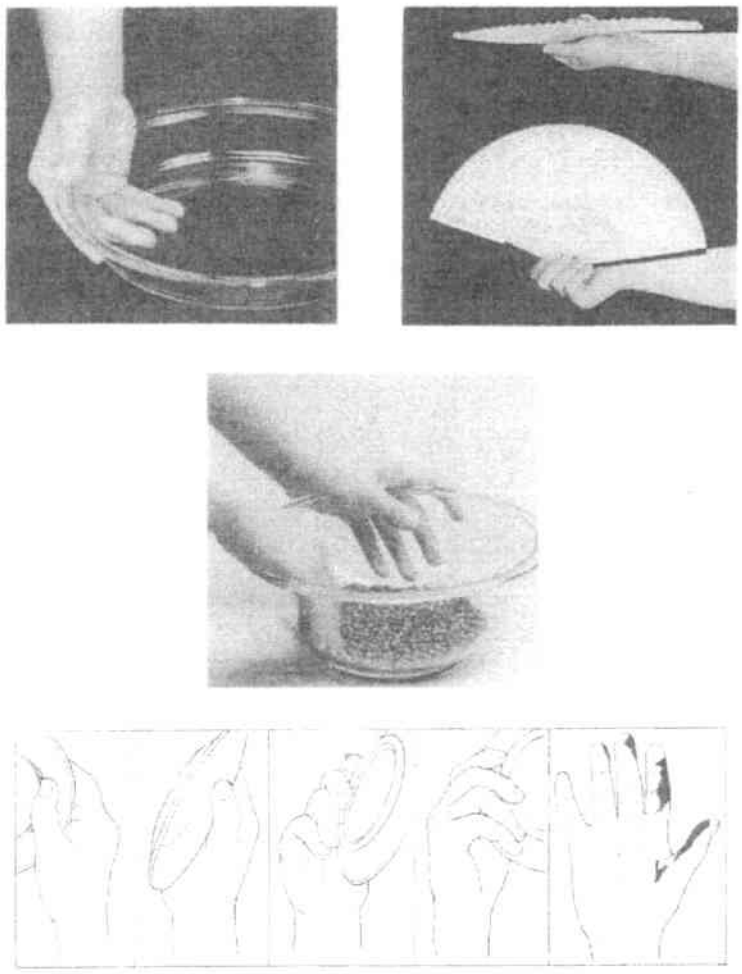

Figure 4.5: The Power Grip Extension Type (PoE)

The Lateral Grip (Lat). This grip is used for fine manipulations when the object or the task requires a greater degree of stability than could be achieved with the fingertips alone. The side of the index finger is used to increase the contact surface. This grip is similar to some forms of the PoD, but is much looser and well adapted to flat object surfaces. It is used for keys, at it puts them well into the rotational axis of the wrist, for handing over credit cards, etc.

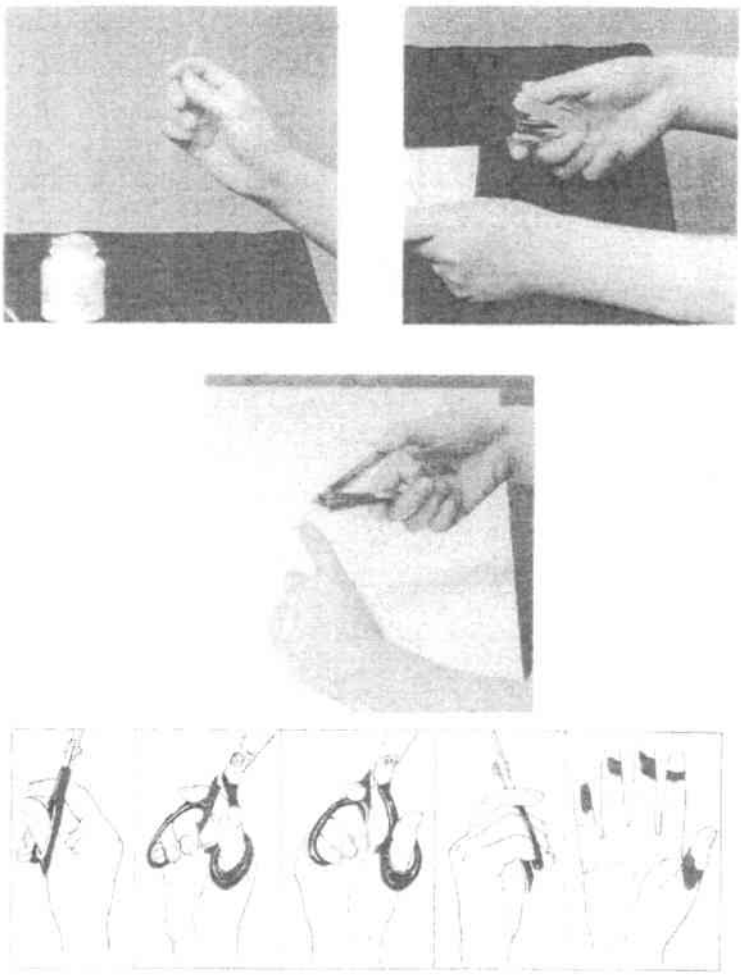

Figure 4.6: The Power Grip Distal Type (PoD)

The Tripod Grip Standard Type (Tpd). In this grip, the thumb, index and middle finger are used to make dexterous manipulations with the tip of a generally cylindrical tool. It allows for great mobility. The object is basically held between the tips of the thumb and index, and pressed by them against the side of the middle finger for more stability. Often, the posterior extremity of the tool is in contact with the side of the hand also. It is used when writing with a pen, holding lipstick, etc

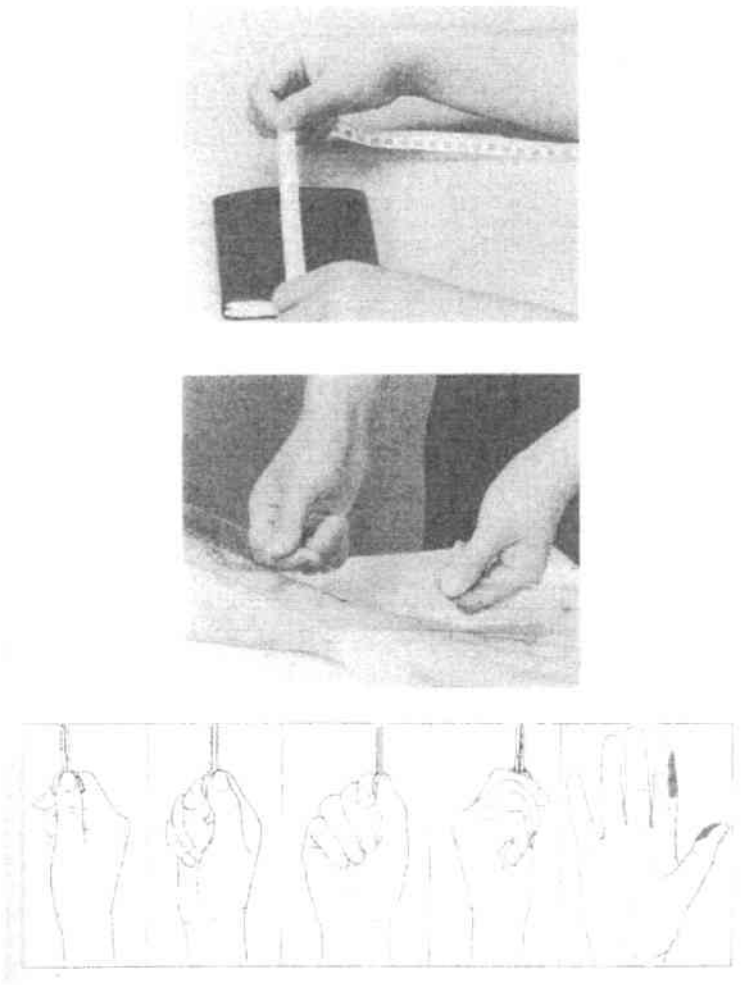

Figure 4.7: The Lateral Grip (Lat)

The Tripod Grip Variation I (TVI). This is a slightly altered version of the standard tripod grip. The distinction into a separate class is justified by considering the purpose of the grasp. In the standard tripod grip the tool tip is held to perform precise manipulations in the extension of the thumb and index fingers. In this variation, the tool tip is perpendicular to the orientation of the thumb, and the tool is pressed on the side of the index. This allows for greater mobility through flexi-

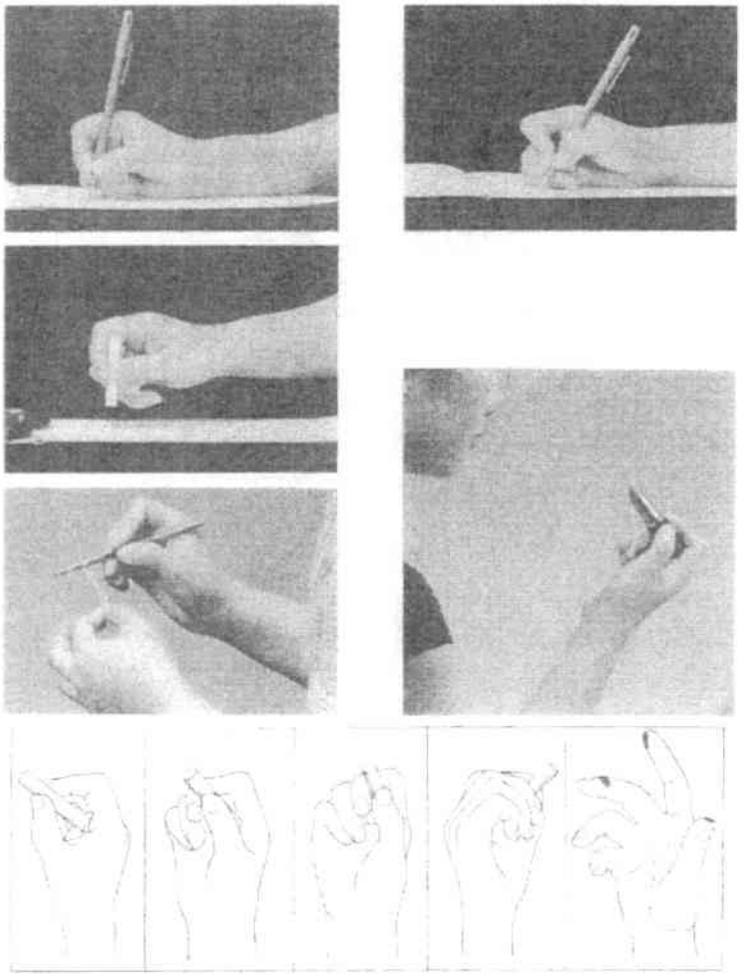

Figure 4.8: The Tripod Grip (Tpd)

on of the index finger or rotation of the wrist. It is used when holding spoons, mixing liquids with a pipe, etc.

The Tripod Grip Variation II (TVII). Yet another tripod grip. Compared to the variation I, the middle finger tip is used and even the ring finger is involved, and both are more extended. This is because the emphasis is put more on precise control of the long tool's tip than on

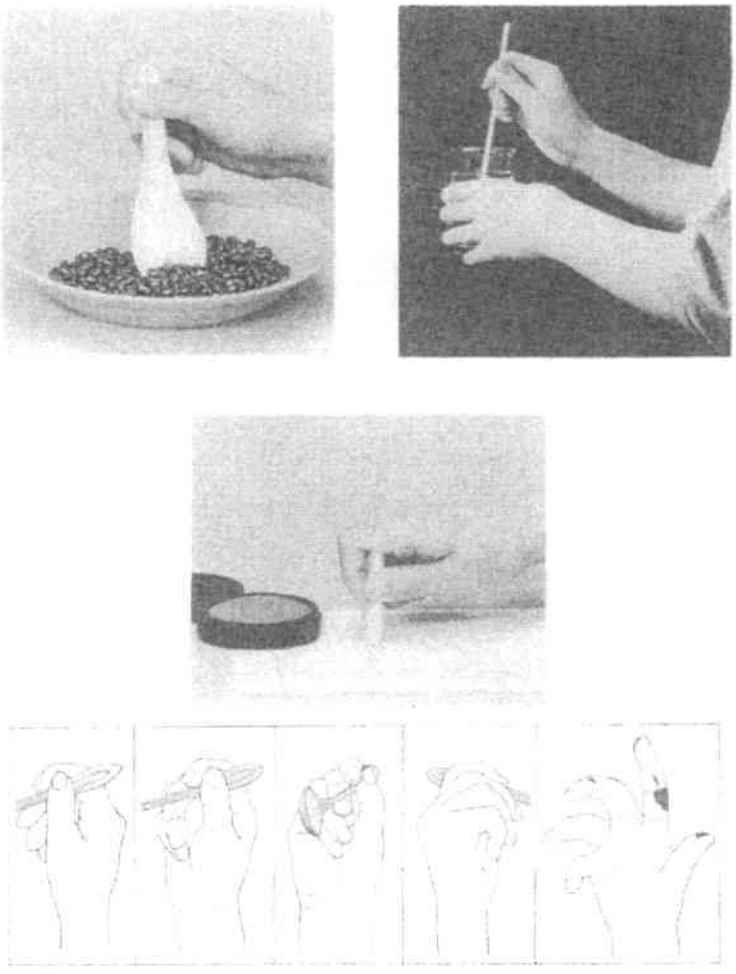

Figure 4.9: The Tripod Grip Variation I (TVI)

mobility. It is used for handling brushes or chopsticks.

3. The Precision Grips:

   The Parallel Mild Flexion Grip (PMF). This is a basic grip we use when picking up objects or holding them lightly. Only the finger tips

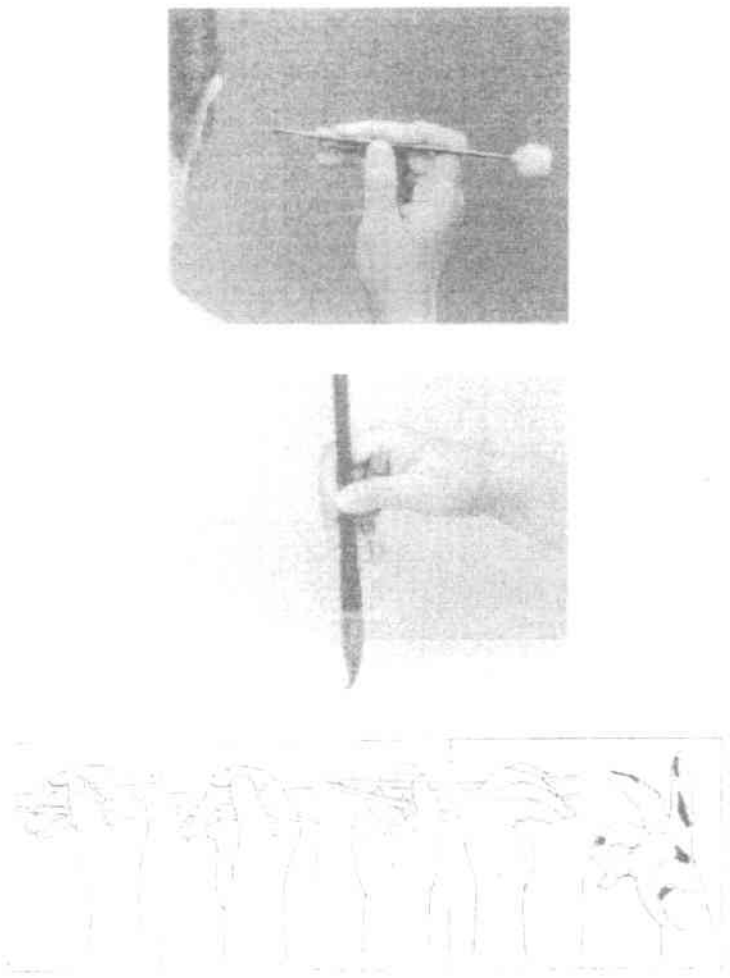

Figure 4.10: The Tripod Grip Variation II (TVII)

are involved to allow for very fine movement. The tips of the fingers in contact with the object form a roughly straight line opposed to the thumb, as dictated by the object's shape. It is used for all kinds of objects, pens, cups, tubes, etc.

The Circular Mild Flexion Grip (CMF). It is basically the same as the PMF, except that the finger tips form a rather circular shape. It is

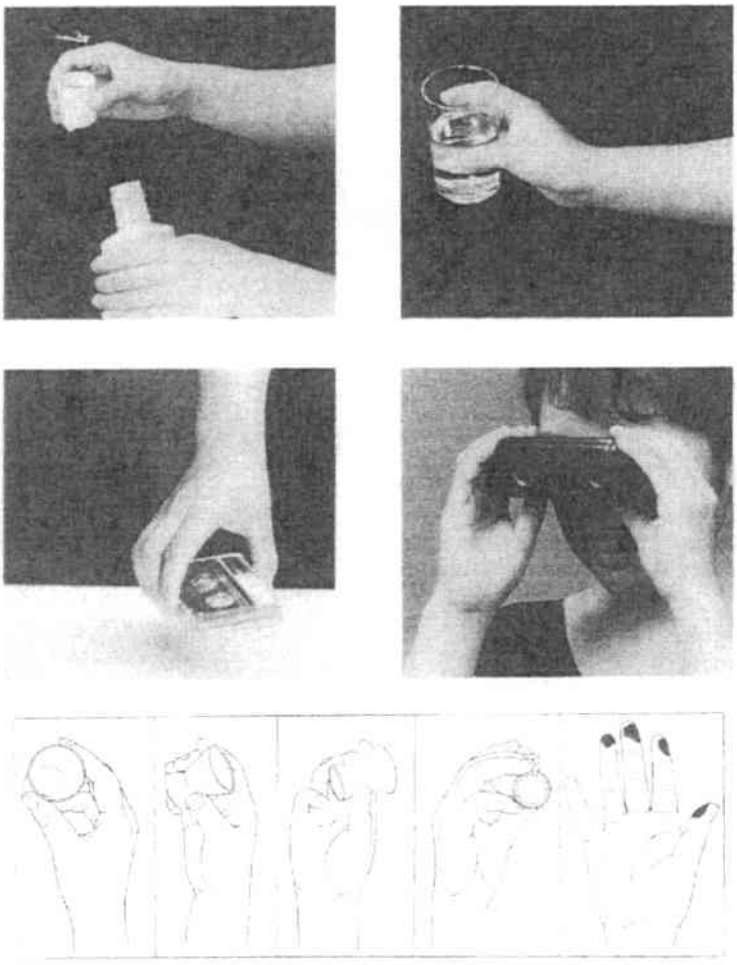

Figure 4.11: The Parallel Mild Flexion Grip (PMF)

often used when a better form closure is needed, as when opening a lid or unscrewing a cap, even if the object itself is not really round.

The Tip Grip (Tip). In this grip, only the very tips of the thumb and index fingers are in contact with the object. This is to manipulate very small objects that could hardly be touched by more fingers simultaneously, and to perform very fine manipulations. It is used when

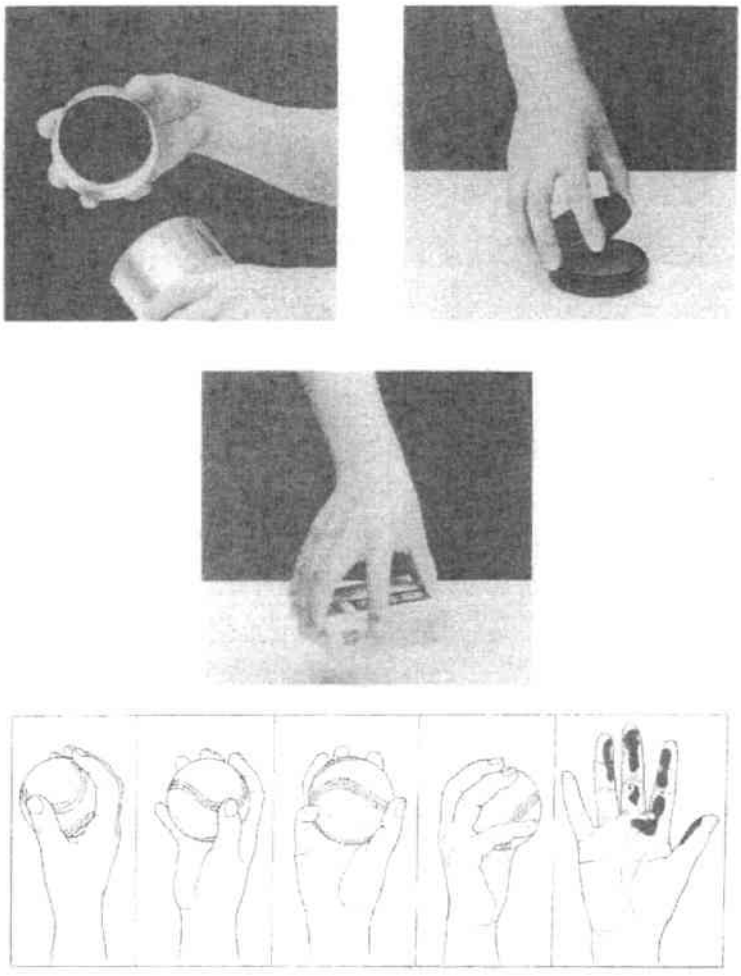

Figure 4.12: The Circular Mild Flexion Grip (CMF)

holding needles, coins, pins, etc.

The Parallel Extension Grip (PE). The PE grip is used for flat objects to keep them relatively stable while avoiding to touch one side, either so it remains visible, or because it is intended to be put in contact with other objects. The fingers are in a line opposed to the thumb, but compared to the CMF grip, they are kept extended to present more

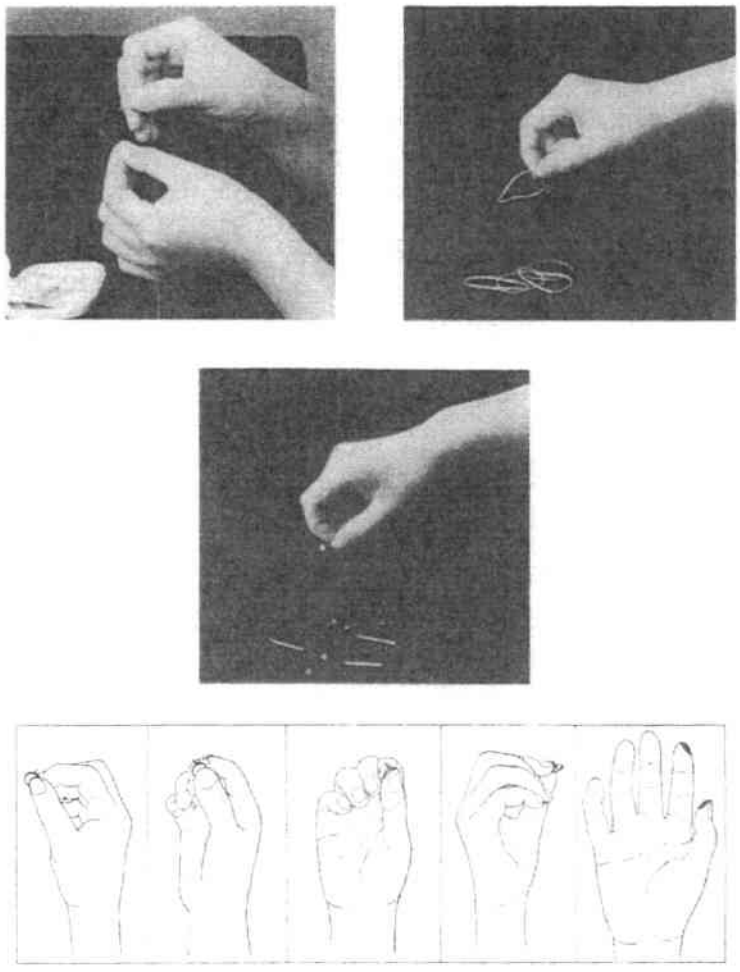

Figure 4.13: The Tip Grip (Tip)

support surface for the object. It is applied when using a handkerchief, showing game cards, etc.

4. The Thumbless Grips:

The Adduction Grip (Add). This is the only grip where the thumb is

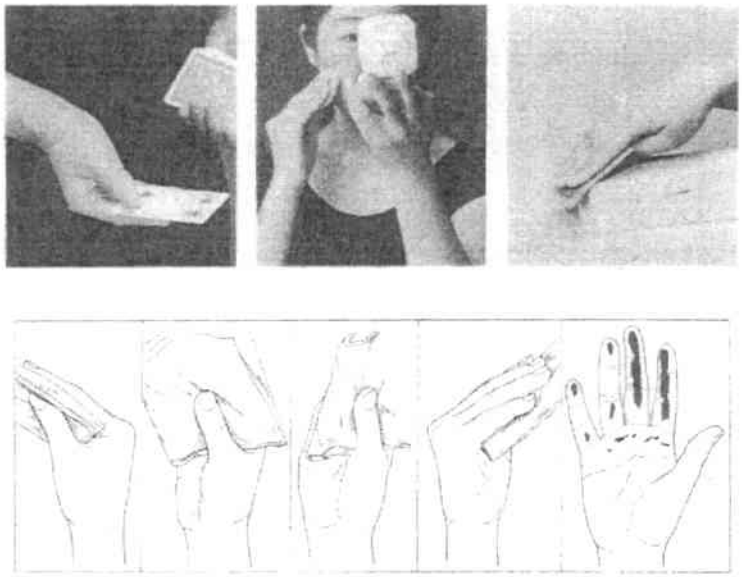

Figure 4.14: The Parallel Extension Grip (PE)

not used. It offers very little stability and is used simply for relaxed holding of the object, for example a pen or a cigarette.

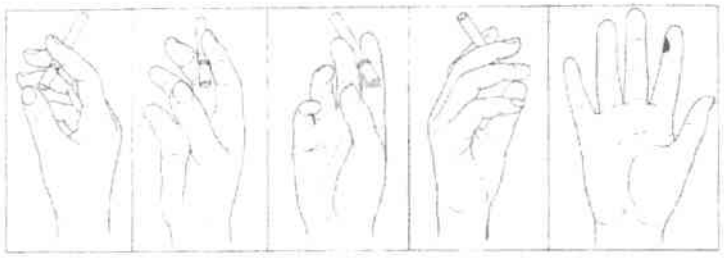

Figure 4.15: The Adduction Grip (Add)

As mentioned before, the taxonomy can be used to classify all grips that are used by humans in everyday manipulation. It offers an analysis of the

finger positions and the contact points to be expected in every grasp class, which is useful for selecting features for the recognition algorithm. However the separation into grasp classes itself is not done by putting the focus on the number of fingers involved or the shape of the hand, but according to general object and manipulation requirements. This will be of great help for making the mapping from human to robot grasps when manipulator configurations differ.

4.3 Input features

Let us now concentrate on the features used to classify the grasp types. The type of input features chosen greatly affects the performance of the recognizer. They in turn depend on the model we use to represent the hand. Two dimensional models, which are often used in vision systems, are suited to recognize communicative gestures, as these are very expressive and are meant to be recognized (by other humans) through shape or movement of the hand alone. They require the extraction of visual features from the image, such as edges, contours, 2D finger positions or relative distances. Since the object is communication, the sender generally orientates his hands towards the observer and shapes them in a way, such that the message is easily understandable, resulting in unambiguous 2D features. These models are not well suited to recognize the much less expressive manipulative gestures, though, as these same assumptions generally do not hold. Of course, one could require from the demonstrator that, when showing grasps to the robot, he does so in a "communicative style", i.e. that he orientates his hands in a certain way, and uses slow, demonstrative, easy to understand moves. His gestures would then have both manipulative and communicative character. But this would require the user to learn the "proper" way to execute a grasp and distance us from our goal of allowing natural execution.
3D models, on the other hand are well applicable to both communicative and manipulative tasks. Their drawback is that they are computationally more expensive when used with vision systems, as they require the extraction of precise positional data, geometric calculations, fitting, etc.

For these reasons, a 3D model of the hand is used in combination with a data glove, bypassing the problems of tracking the hand, extracting and fitting the features. Of course, the glove has its disadvantages, as it reduces the dexterity

of the operator, but for our system, the benefits outweigh the drawbacks.

4.3.1 Data glove

The very popular Cyberglove by Virtual Technologies Inc. [42, 43] is used. This glove is light weight, made of soft, flexible but robust fabric, and relatively non-obstructive (Figure 4.16).

Figure 4.16: The Virtex Technologies Inc. Cyberglove.

The 18-sensor version of the glove is used. It has open fingertips and an array of bend sensors on its back to measure the flexion of the fingers. It is connected by a 3m cable to its instrumentation box which delivers the joint angle values through a serial interface at a rate of 38400 baud.
The Cyberglove is usually used in conjunction with a Polhemus Fastrak magnetic tracker which locates the position of the hand and its rotation in a world coordinate frame. However, since as long as the hand configuration doesn't change, a grasp is the same regardless of its orientation or the place where it occurs, the information from the tracker is not used in recognition.

The glove can be calibrated to fit the hand geometry of many users. Every sensor value can be tuned and adjusted separately. For this system however, only a very simple calibration was made. The user is asked to make two distinct gestures:

1. The hand is held out open and flat, the fingers and thumb are on the same plane, joined and extended.

2. The thumb and index finger touch at the tips, describing approximately a circle. The rest of the fingers are joined and extended (Figure 4.17).

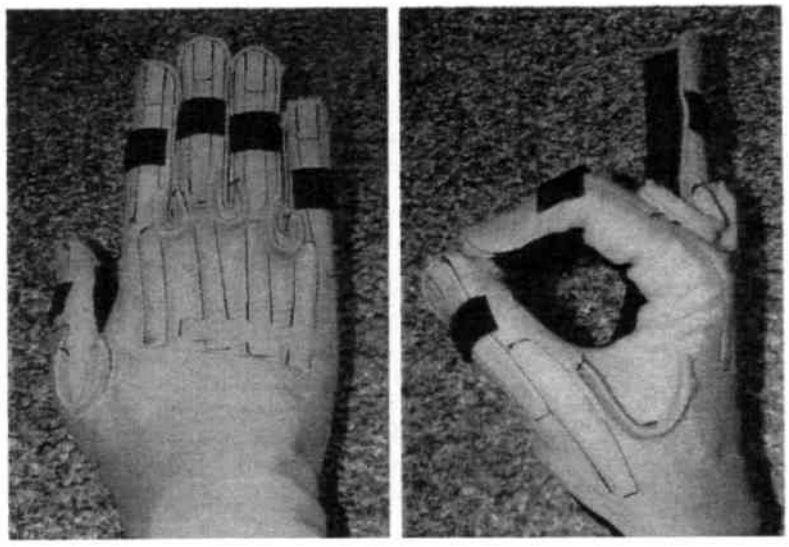

Figure 4.17: Calibrating the Cyberglove.

The joint angle values for these gestures are recorded and an automatic calibration for all the sensors is made based on this information. Of course, this simple calibration can not account for all the variations in the user's hand geometry. However, more precision is not required, as the goal is not to gain an exact representation of the actual hand posture, but to classify it into a grasp type, and the various object sizes or unintentional differences in the used grasping style account for more variation than the calibration errors.

4.3.2 Hand model

A simple skeleton model of the hand is used, the parameters being the values returned by the Cyberglove. A total of 16 parameter values are used. These are: 15 values for the finger joint angles (4 for flexion of the thumb and its abduction, 2 for flexion of each finger, and 3 for relative finger abduction angles), and 1 value for the arching of the palm.

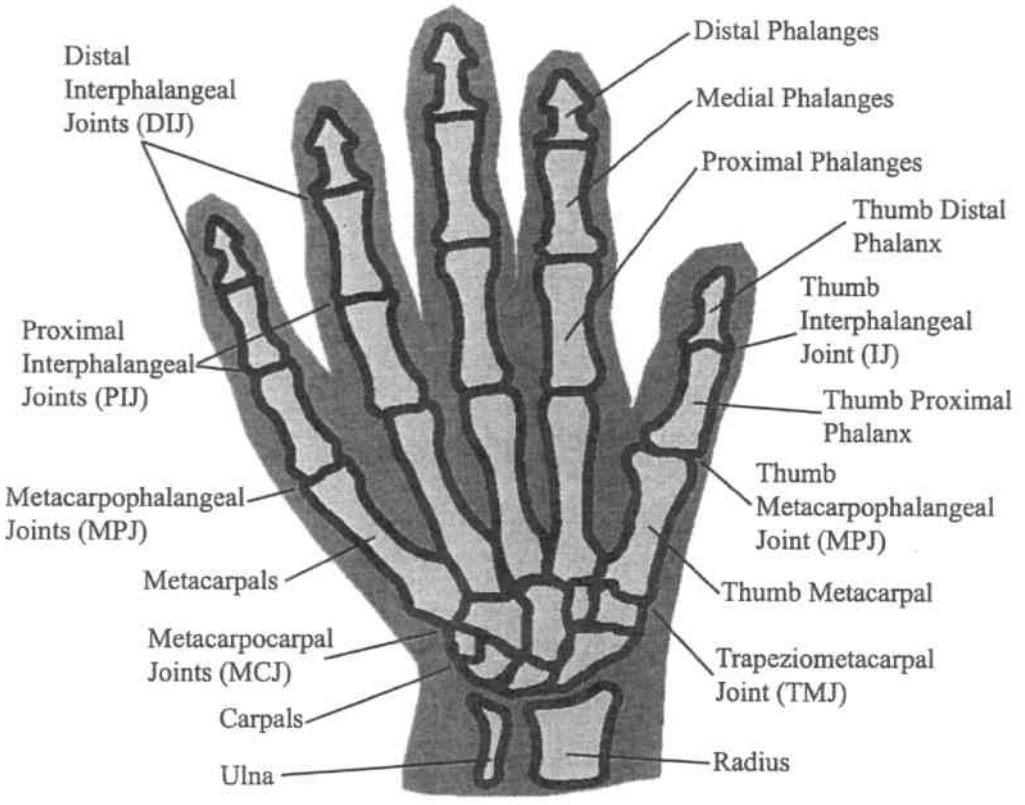

Figure 4.18: Skeleton model of the right hand (palmar view).

In the following, the joints are referred to as:

- MPJ (Metacarpophalangeal Joint): This is the junction point where the finger and palm meet.

- PIJ (Proximal Interphalangeal Joint): This is the finger joint that comes directly after the MPJ.
- DIJ (Distal Interphalangeal Joint): This is the finger joint closest to the end of the finger.
- TMJ (Trapeziometacarpal Joint): This is the joint where the thumb metacarpal connects to the carpals.

The 18-sensor Cyberglove does not have sensors to measure the flexion of the distal interphalangeal joints (DIJ) of the index through pinkie fingers. However, this value is correlated to that of the PIP joints. It is natural for most people to bend their fingers so that both the DIJ and PIJ bend together in some proportion. For simulation purposes, the value of the DIJ can therefore be estimated, using some simple heuristics. But for recognition purposes, the estimated values would only hold redundant information. They are therefore ignored.

The thumb is modeled to rotate first at the TMJ around the axis going through the index MPJ, and then rotate by the abduction angle towards or away from the index.
The abduction values for the other fingers are delivered by 3 sensors on the dorsal side of the glove: The index-middle, middle-ring, and ring-pinkie abduction sensors. These sensors do not measure the absolute abduction of the fingers relative to the palm, but their relative abduction to each other, i.e. the spread between two adjacent fingers. To obtain the absolute values for the index, middle and ring finger, a tradeoff is made. The proportion with which the middle finger should respond to values of the index-middle and middle-ring abduction sensors is set. The absolute abduction for the pinkie is then calculated from the relative value, after the position of the ring finger was determined. Although this induces some error, it is not judged to be crucial for the distinction of the grasps.

The Cyberglove also includes a sensor to measure the arching of the palm. When the tip of the thumb reaches for the tip of the pinkie finger, the metacarpus arches somewhat, changing the plane through which the ring and pinkie fingers flex. This also occurs is certain grasps, such as the Tripod Variation II (TVII) or some forms of the Circular Mild Flexion Grip (CMF), and the degree of arching can be useful for recognition.

The last two values returned by the glove, the wrist pitch and yaw values, are not used in the recognition procedure, as they rarely contain information useful for the distinction of grasps. On the contrary, when performing movements with an object during grasping, such as hammering or scooping, the changing wrist angle values introduce additional, unwanted variation, increasing the configuration space of the grasp class and thus, the number of training examples needed to stably define its boundaries.

4.3.3 Tactile sensors

The following main requirements were determined for the tactile sensors covering the hand surface:

1. They should cover large enough surfaces on the hand. Even when making the same grasp on the same object, the contact surfaces vary somewhat. If the sensor (e.g. on the fingertip) is too small it may not be activated, or only in part, decreasing recognition accuracy. To remedy this, the user can be led to change his way of grasping to make sure all sensors get used effectively. But this reduces the naturalness of execution.

2. They should be sensitive enough to detect light contact. The user should not be asked to exert untypical forces for a given grasp or object, just to insure sensor activation.

3. They should be flexible and thin, so they can be fit to the finger or palm shape. Hard sensors would greatly impair dexterity and sensitivity. Also, since the palm itself is very flexible, a good coverage can only be achieved using flexible sensors.

For this system, a custom set of capacitive pressure sensitive sensors has been used. The sensors are manufactured by Pressure Profile Systems, Inc. (PPS)[1]. They operate by the following principle (see figure 4.19): Two electrodes made from conductive cloth are separated by an air gap of distance d. They are kept apart by a compliant layer, which compresses under pressure to allow the air gap to change. As the distance d changes, so does the capacitance of

[1] Pressure Profile Systems Inc. (PPS). 5757 Century Boulevard, Suite 600. Los Angeles, CA 90045. www.pressureprofile.com

the sensor. This change of capacitance is measured to determine the applied pressure. Since the two electrodes are non-contacting, the results are more repeatable and the sensor's performance is less likely to degrade over time. It is also more sensitive and resistant to changes in temperature. The sensors are 0.85mm thick, their size and shape is customizable. They possess a full scale range (FSR) of 60psi, are sensitive up to 0.06psi and have a non-repeatability scale factor of 0.1%. They produce analog output, which must be transformed by an analog / digital converter before being further processed. They are quite flexible and provide accurate information even when bent around the fingers.

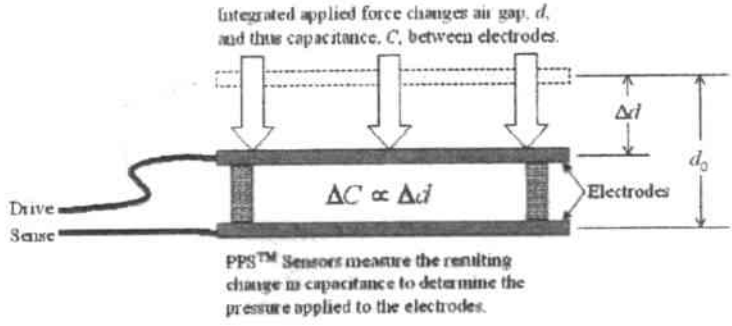

Figure 4.19: Anatomy of a capacitive pressure sensor.

For the recognition of the grasps in our classification table, precise information on contact points or the exerted force are not important, since these may vary greatly with the object. Only the information about occurrence of contact at a few main hand regions is needed.

These regions were chosen in order to maximize the chance of distinguishing grasps of the classification table that show too much similarity based on the hand shape alone. An analysis of the different grasps in the table and of their expected contact surfaces was made (Figure 4.21). Since covering the whole hand with sensors would be too costly and too obstructive for the user, the main surfaces used in the grasps were identified and only the ones most frequently used or most useful for recognizing a specific grasp were covered.

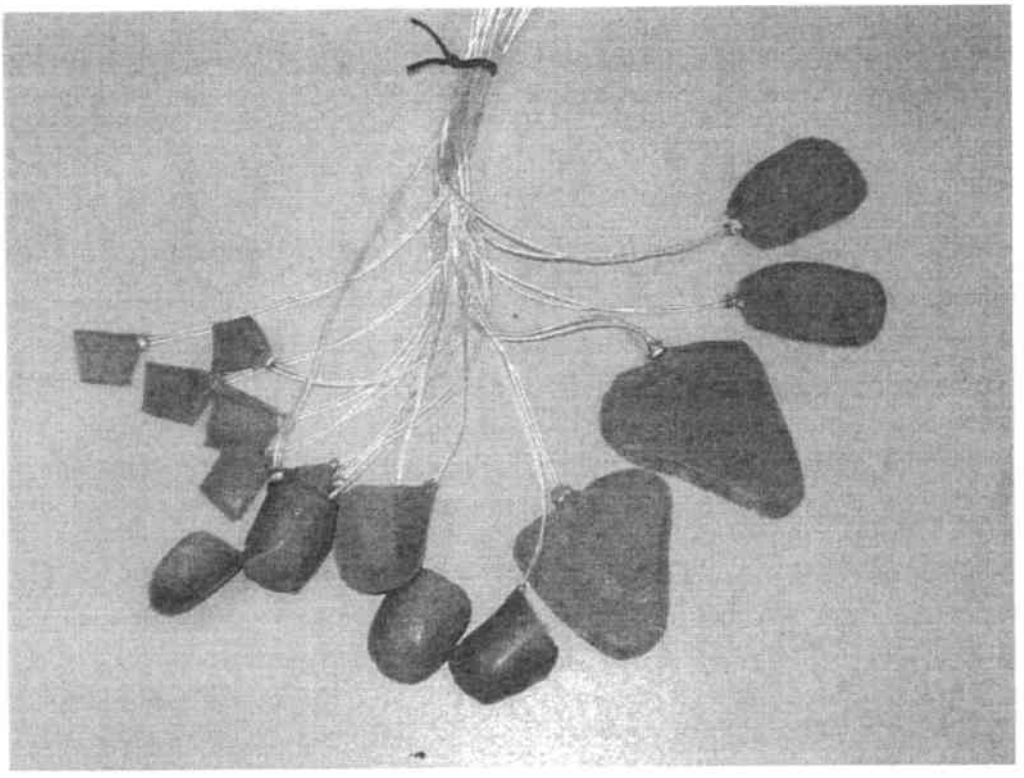

Figure 4.20: The flexible capacitive sensors.

As expected, the surfaces of the fingertips, especially the ones from index finger and thumb are the most often used. The palmar sides of the proximal finger segments are also often used in power grasps, but were not judged useful for distinguishing grasps, since the recognition can be made based on the activation of the palm.

The distribution of sensors on the palm was found not to be so critical. The palm is used almost exclusively for power grips. However, for the different classes, one can notice slight differences. While in the PoS and PoI the whole surface of the palm is covered evenly (except for small objects), the PoH tends to concentrate more on the distal part and the PoE on the radial part. The proximal ulnar metacarpal part is rarely used.

Some regions on the finger sides were found of particular interest to distinguish grasps. Especially the radial distal and middle parts of the index finger

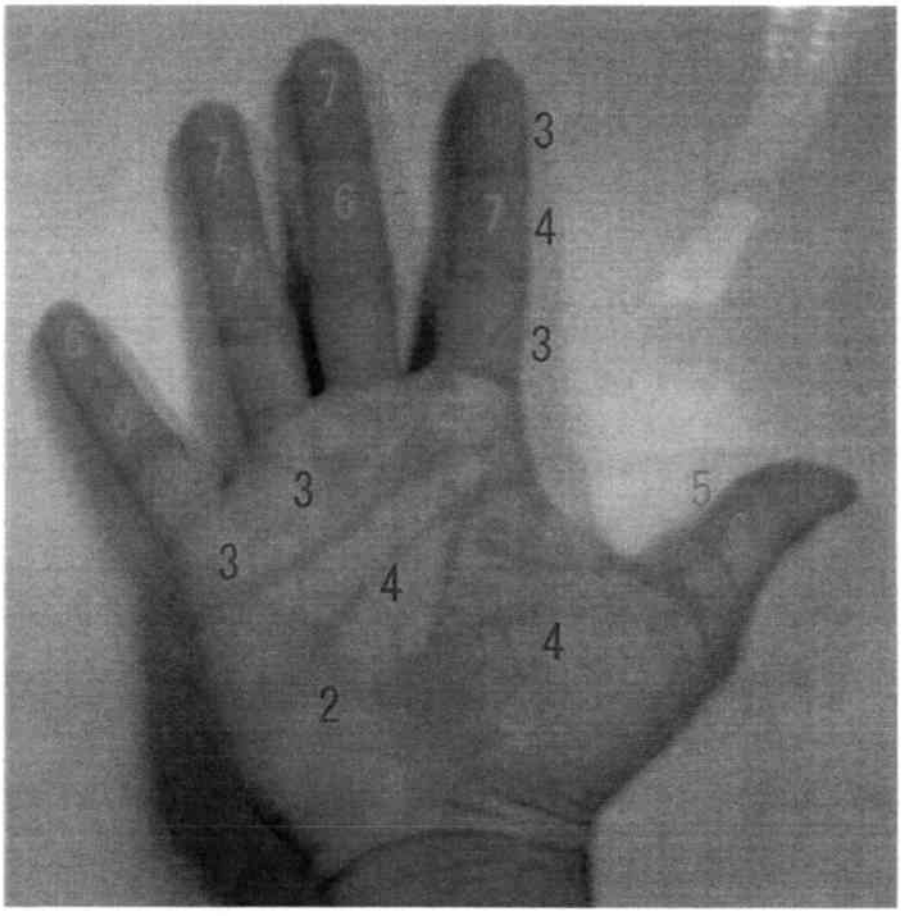

Figure 4.21: Analysis of the contact regions of the hand. The numbers are on an abstract scale representing how often the respective region is used throughout the different grasp classes. 10 means the region is generally active for almost all grasp classes. 1 means the region is used only by one or two grasp classes. The thumb and index fingertips are the most frequently used regions.

are used in the lateral grip (Lat). They can be useful for distinguishing the Lat from the distal power grip (PoD) and tripods. The radial proximal part of the index, on the other hand, can be used to distinguish the tripod variations (TVI, TVII) from the standard tripod (Tpd) or the PoD. The side of the middle finger facing the index is useful for detecting the PoD and some

tripods.

Based on this analysis, the following configuration for the tactile sensor array has been chosen:

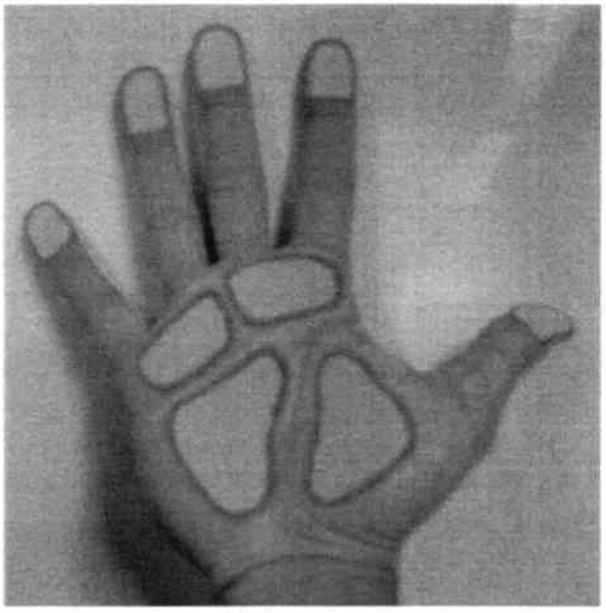

Figure 4.22: Configuration of the tactile sensors, front view.

For the thumb and finger tips, a cylindrical pressure sensor adapted to the shape of the fingertips was used. It covers not only the palmar region, but also the very tip, which is useful for detecting grasps like the Tip Grip.
The palm itself is covered with 4 large flat sensors. Here it is not important to recognize the precise contact points, since these can differ greatly for one same grasp depending on the shape of the object. Rather the general surfaces where contact occurs are to be determined. The distal radial sensor is the most important and is used in all power grips. The activation of only radial sensors can be a sign for a PoE, and the activation of only distal ones, a sign for a PoH. Simultaneous activation of all sensors is a good sign for a PoS or PoI, and the distinction is best made using the index's shape.
The sides of the fingers are covered as follows: one sensor each for the radial side of the distal, middle and proximal phalanges of the index finger. These

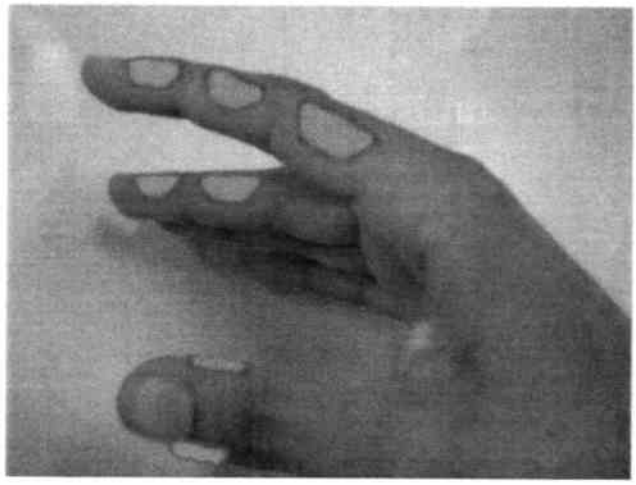

Figure 4.23: Configuration of the tactile sensors, side view.

are most useful for detecting the lateral grip and tripod variations. One sensor each for the radial side of the distal and middle phalanges of the middle finger, for detection of tripods. Finally, one sensor at each side of the distal phalanx of the thumb.
The total amount of sensors is 16, distributed as shown in Figures 4.22 and 4.23.

4.3.4 Sensor fusion

The Hidden Markov Model based architecture of the recognizer allows to integrate the data from the Cyberglove and the tactile sensors in a very elegant way. A fusion of similar hand posture and contact point information was already proposed in [51]. However, the data from the tactile sensors was used only for segmenting the user demonstration using force profile analysis, and the analysis of the grasp itself was based solely on the finger joint angles measured by the glove.
In this approach, both positional and tactile information are simply used as inputs to the HMM recognizer and both the segmentation and classification are made based on the combined input stream.
The data is captured from the Cyberglove and transmitted by its instru-

Figure 4.24: The tactile glove. The sensors are attached to the inside of a slim elastic fiber glove to assure they always fit on the right place. It is to be worn under the Cyberglove. The design still allows to use both input devices separately.

mentation box through a serial cable to a Pentium III 550 MHz Windows NT machine at 38400 baud. The data from the tactile sensors is first passed through a 1m cable to their instrumentation box, where it is processed and then further sent to the A/D converter board of the capture machine. All data is first buffered by a Corba server architecture where synchronization is made and a time stamp is attached to it. Then it is passed on to other machines at a maximum refresh rate of 10ms. The software for recording demonstrations, training and recognition itself runs on a separate Linux machine and the data must first be passed to it by the Corba server. Due to network latency, the constant refresh rate of 10ms could not be maintained during recording and occasionally a 10 ms frame was skipped. This had however no significant effect on the recognition accuracy, as the HMM recognizer was robust enough to efficiently filter out the resulting noise.

Figure 4.25: The Cyberglove-tactile glove combination.

4.4 Implementation of the HMM recognizer

The commercially available Hidden Markov Model toolkit HTK was used for implementation of the HMM recognizer routines. It offers libraries for definition of HMM topologies, performing Viterbi search, calculating forward and backward probabilities, Baum-Welch reestimation, handling of large training and label files, etc. The routines can be included in source code or accessed from the command shell. Since lots of training with different parameters and demonstration sets was intended, a more comfortable graphical user interface was designed in Qt [6] to allow for fast execution of the various routines, experimenting with parameters, such as the HMM topologies, the number of training iterations, sorting, editing, labeling training and test data, etc. Also it allows to play back a simulation of the demonstration, viewing hand shape and tactile sensor activation together with classification and segmentation results.

4.4.1 Feature vectors

As features for the HMM recognizer, the finger joint angle values, their derivatives, and the tactile sensor output values were used. The derivatives are calculated by the Corba server. Also, the maximum value of the tactile sensor activation was considered. The values of the tactile sensors for a given time frame were taken, their maximum calculated and the resulting value used as separate input feature. It is expected that the tactile maximum will increase the quality of segmentation when little training data is available, for the following reason: Consider a grasp such as the Parallel Mild Flexion Grip. Sometimes all the fingertips are involved in the grasp, sometimes the ring or pinkie finger is left out. Also, it can happen that, depending on the shape of the object, the sensor on the inner side of the thumb, not the thumb tip sensor gets activated. So the system cannot depend on a particular tactile sensor or configuration of sensors to recognize the time of contact with the object. It can only approximate the region in feature space which can be considered as grasp, after being presented enough training examples containing the necessary variability. The tactile maximum, on the other hand, presents an unambiguous way of determining when contact occurs. It quite stably marks the beginning point of a grasp, no matter what particular sensors were activated. It allows an easy separation of the high dimensional feature space that can be learned even with little training data.

4.4.2 Topology

In speech recognition, the objects of recognition, the words, can be decomposed according to their pronunciation into smaller units, the phonemes, or even into phoneme parts, and HMMs created and trained for these basic units. The justification is that the phonemes reduce the amount of models needed and, since they appear much more often than words, can be trained more robustly. For gestures, it is not easy to define what basic units should be made of. For gestures which involve hand movement, units such as lines or arcs could be used and a more complex move understood as a sequence of those. For grasps, such intuitive decompositions cannot be made. But since the number of grasps to be recognized is very small (as compared to words), a robust training can be achieved even when taking a whole grasp as basic unit.
Another decomposition often made in speech is the separation of phonemes

into three parts, beginning, middle and end, often combined with context dependent modeling. This is because the middle part of a phoneme does not vary as much as the beginning or end, depending on the preceding and following phonemes. However, context modeling requires a lot of training data. Even if a grasp type is represented often enough in the data to be trained robustly, the possibility of it appearing in all possible contexts is small and a training of the context dependent models would require exponentially larger training sets. Since the process of recording demonstrations is still relatively time consuming, and there are no commonly available databases as in speech recognition, a splitting and context dependent training of the models was not made.

Therefore, in this system, every grasp is modeled by one HMM. A flat topology is chosen, where every state has a transition to itself and to its successor. In principle, when using Hidden Markov Models, one should be able to assume that the modeled process is made up of discrete states. Of course, the hand movement when grasping is continuous and when modeling it with discrete states, an approximation is made. Determining the optimal number of HMM states is a difficult matter. Choosing a too small number means our approximation would be too imprecise. Choosing a large number would slow down the recognizer and, more importantly, reduce the number of training samples per HMM state. Here, the exact number of states is to be determined experimentally, by observing when an increase in HMM sizes yields no more significant increase in recognition accuracy.

Thus, a HMM was created for each of the 14 grasps in the classification table (PoS, PoH, PoI, PoE, PoD, Lat, Tpd, TVI, TVII, PMF, CMF, Tip, Add). A common HMM was also used to model the releasing motion after the grasps (RLS). The idea is that since we want to recognize the moment an object is ungrasped, we model this movement and let the recognizer find its occurrence in the sequence during automatic segmentation just as it does for a normal gesture. Of course, we could define that ungrasping occurs at the end of a grasp segment. But since garbage motion can occur during the grasp before the release motion or directly after it, the end of the grasp segment is not always placed precisely at the right time frame. The use of a distinct model to detect ungrasping improves the placement of boundaries in the presence of garbage motion. Since a classification of release moves depending on the previous type of grasp would bring us no important information, one and the

same model is used for all cases. It has the same topology as a grasp model.

A garbage model (GARBAGE) to filter out noise, involuntary moves, etc was also created. As Lee and Kim pointed out in [27], when performing sequences of gestures, non-gesture moves are sometimes made in between two gesture moves. If these non-gesture segments are not correctly identified, they are wrongfully deemed part of the gestures and hurt the classification process. When performing grasps, we also make voluntary or involuntary moves that make the classification process more difficult. For example, we may break off an initiated grasp for a fraction of a second to make minor adjustments with the fingers. Or we may involuntarily make some moves with the fingers while going from one object to the other. Since movement occurs, the system will try to classify it and the grasp class whose model outputs the highest probability will be chosen.
Lee and Kim solve the problem by introducing a special type of garbage model, the output of which is used to threshold the output of other models. This "threshold" model contains as states copies of the states of all the getsure models and has an ergodic topology. When a correct gesture is made, the output of the correct model is still higher than that of the garbage model and it is chosen as classification result. But when a non-gesture move is made, the output of all gesture models is relatively low, and the garbage model fits the observation best. This helps find the bounds for the gesture segments more precisely and increases recognition accuracy.
In this system, the garbage model has been implemented with a fixed number of states: the same as in the grasp models. As for Lee and Kim, an ergodic topology was chosen: a transition exists from every state to every other state. It is not a threshold model, like the one Lee and Kim used, as its states are not copies of gesture model states, but it serves the same purpose: To threshold the output of the grasp models. The garbage model was trained using segments of non-gesture data that occur in between grasps. It is expected that the Baum-Welch algorithm profits from the maximum complexity of the model and automatically trains states to best fit the very irregular garbage data.

4.4.3 Task grammar

As in speech recognition, accuracy can be increased by the use of a grammar that dictates which sequences of actions are allowable and which not, reducing

the search space for the dynamic programming algorithm.
Grammars may be obtained, for example, by analyzing big text databases from newspapers, accumulated over the years, and calculating the probability of co-occurrence of two or more words. In this case they are called bigrams. Grammars that associate three words are called trigrams.
Whereas these statistical grammars are useful in speech, where the vocabulary is generally very big, in the case of gesture recognition a small grammar can be built manually, considering the requirements of the task, for example in form of a regular expression.
Here, only a very simple constraint has been imposed: A grasp motion is always followed by a release motion (RLS). So only simple grasp and release sequences, no grasp transitions are considered. Thus, the following grammar for recognition of an entire sequence is used:

$$([GARBAGE \mid SIL]\ GRASP\ [GARBAGE]\ RLS\ [GARBAGE \mid SIL])*$$

with

$$\begin{aligned} GRASP = (&PoS \mid PoH \mid PoI \mid PoE \mid PoD \mid Lat \mid Tpd \\ &\mid TVI \mid TVII \mid PMF \mid CMF \mid Tip \mid Add) \end{aligned}$$

Optional arguments are in brackets and the operator "*" represents, as usual in regular expressions, one, many or no repetitions of the preceding argument. As can be seen, the system is allowed to introduce garbage motion at any point between other symbols in the sequence and to insert silence between grasp - release groups.

4.4.4 Training and test

The training of HMMs is done with supervised learning techniques, which means that two things are required:

1. The training data consisting of the actual sequence of feature vectors from the user demonstration and

2. A transcription, or label, describing the desired segmentation and classification result, i.e. the actual symbol sequence and the time frames where they start and stop.

To obtain the labels, one can of course view the demonstration data offline and manually set bounds, assign segments to classes. But this is a tedious and time consuming procedure and rarely done. Instead, a suboptimal but convenient procedure is used. Only the transcriptions of the executed movements is needed, i.e. the sequence of symbols describing the action, without any segmentation information:
A flat start is made to initialize the models. The training set is scanned and all the component means and covariances are set to the global data mean and covariance. So initially, all the models are given the same parameters. Embedded training is then made using the Baum-Welch algorithm. Since continuous recognition of whole sequences of grasps is to be made, it has to be determined, which models to train on which part of the demonstration. For each training demonstration, a composite model is therefore synthesized by concatenating the models listed in its transcription. All model parameters are then adjusted simultaneously by performing a standard Baum-Welch pass over each training demonstration using the composite model, the algorithm iteratively adjusting the segmentation bounds to maximize the observation probability.

For recognition, HTK's Viterbi recognizer was used. The result is a text output of the recognized symbol sequence which can be compared to the initial transcription for error calculation. Accuracy is measured as usual in the continuous speech recognition literature. The substitutions, insertions and deletions are added and the total number divided by the number of symbols contained in the transcription to obtain the error rate.

$$Err = \frac{\#Sub+\#Ins+\#Del}{\#Symbols}$$

The accuracy rate is then obtained as:

$$Acc = 1 - Err$$

Chapter 5

Experiments

In this chapter, the experimental results of the designed grasp recognition system are presented. First the workspace, manipulated objects and training and test conditions are presented. A comparison of recognition results for a variety of input feature configurations is made first for single user systems, and later for a multi-user system with four users. The effect of the tactile sensor data on the segmentation quality is analyzed and the usefulness of the garbage model is demonstrated on a few sample recognition outputs. Finally, an analysis of the results is made and the level of naturalness achieved in the user execution is discussed.

5.1 Experimental setup

All demonstrations were performed on a flat table top, in front of the robot system. Theoretically, considering the way the input features are obtained, they could have been performed at any place inside the room. But the small length of the cable connecting the glove to its interface unit at the current time have restricted the their execution to this relatively small workspace. A variety of objects of different shapes and sizes have been used (see Figure 5.1). Special care was taken to insure that for every grasp type, multiple objects with different properties were available. For example, the Power Grip Standard type could be performed with the jars, the hammers, the heavy marker, etc.

Training and testing were performed offline. The training and test demon-

Figure 5.1: The objects used for recognition. As can be seen, a broad palette of objects was considered. For one same grasp type, many objects of different shapes and sizes were available (For example: Jars, cup, cassette for the PMF; plate, book for the PoE; ruler, key for the Lat; etc).

strations were collected from 4 users. Every user delivered 56 short demonstrations containing 3 different grasps from the classification table. In total, 224 demonstrations were recorded. These were then evenly split into a training and a test set. The training set contains 112 demonstrations, with 28 demonstrations from each user. Usually, the size of the training set is chosen to be somewhat larger than that of the test set. But here, 112 demonstrations were judged to be sufficient to train the 14 grasp classes and the test set was deliberately chosen very large. Since the performance of the system depends on a lot of parameters, such as the input features, the Hidden Markov Model topologies, the number of training iterations, even small changes in recognition accuracy should be noticeable in order to evaluate the importance of parameter changes.

As the complete test data set contains 336 grasps, one insertion error, for example, would cause an increase in recognition error (and consequently a

reduction in accuracy) of $1/336 \approx 0.3\%$. Therefore, with this test set size, accuracy rate values are still meaningful up to the first decimal digit. As minute changes in the parameters can quickly cause a few insertion, deletion or substitution errors, choosing a too small test set would cause the accuracy rate to fluctuate greatly due to statistical noise. Thus the size of the test set gives more soundness to the presented results, even if this means less training data is available.

The four users (in the following referred to as User1 - User4) had quite different hand sizes and shapes, ensuring that enough variation is included for a good evaluation of the multi-user system performance. Unfortunately, the author himself could not provide any training or test data, as his hands were too big to fit in the Cyberglove. This problem was made more acute by the fact the Cyberglove had to be worn on top of the tactile glove. All demonstrations were therefore performed by relatively unexperienced users that had no precise prior knowledge of the Kamakura table and the difference between its grasp types. They therefore performed the grasps more naturally than an expert probably would have.

The demonstrations were recorded in the following manner: First, a set of labels for the demonstrations were created beforehand, and the users then asked to perform the demonstrations corresponding to these labels.
It was intended to recognize grasp and release sequences. That's why for every user, a set of labels was automatically created, containing three grasp-release pairs in sequence. The sequences were not purely random, but the generating algorithm ensured that there was sufficient combinational variation and that in the resulting label set, all grasps were represented approximately the same amount of times. When recording, the user was instructed which sequence of grasps to perform, and the demonstrations captured one by one.
This method has the advantage, that the demonstration labels, used for adapting the HMMs in training and for evaluating recognizer hypotheses in test, can be easily created. It is much less time consuming than recording the demonstrations and labeling them correctly afterwards. But it comes with a slight disadvantage: the label may not match the demonstration perfectly. For example, if the user hesitates slightly or moves his fingers in between grasps, it would be useful to add a GARBAGE symbol at the appropriate point in the label. Otherwise, the label does not match the demonstration, hurting training efficiency. To remedy this, when training with a garbage model, the

label set was preprocessed. Garbage symbols were automatically added in between each grasp-release pair indistinctively, even where the execution was made very smoothly. Although this in turn introduces unnecesary garbage in labels for very clean demonstrations, it is expected that the ergodic garbage model copes well with the situation due to its extremely flexible topology.

One notable problem complicated the recording and deserves to be mentioned here. While the surface of the tactile sensors itself proved to be very flexible and robust, the fixation points of their connector cables are relatively stiff and sensitive to contact. As the sensors are fixed on the surface of the fingers and hands, these points often get bent or come into contact with objects during manipulation, causing the sensors to produce erroneous data. This problem was getting more and more acute in some of the sensors, as demonstrations were being recorded. The affected sensors were the thumb inner and outer side sensors, the pinkie and middle finger tips, the radial palm sensors, the distal side sensor of the middle finger and the proximal side sensor of the index. Since the use of uncertain data definitely hurt recognition performance, the affected sensor values were taken out of the input stream completely and training and test done only with data from those sensors that showed robust behavior. Of course, this reduces the richness of tactile information obtained. But it was hoped that the system performs well even with a reduced set of tactile sensors, until the problem could be corrected. It is intended to redesign or consolidate the cable connection points to make them more robust and prevent the problem from reoccuring in the future.

5.2 Classification results

5.2.1 Explanation of figures

In the following, snapshots of window parts of the created GUI are used to illustrate graphically the recognition results (Figure 5.2).

The top two windows represent the scene at a given time frame. The left window shows the activation of the tactile sensors. The colors range from blue, when the sensor is not activated, to bright orange, when its output is maximal. Sensors that were left out because of instable behavior are displayed in much darker colors. The right window shows the shape of the hand. Under

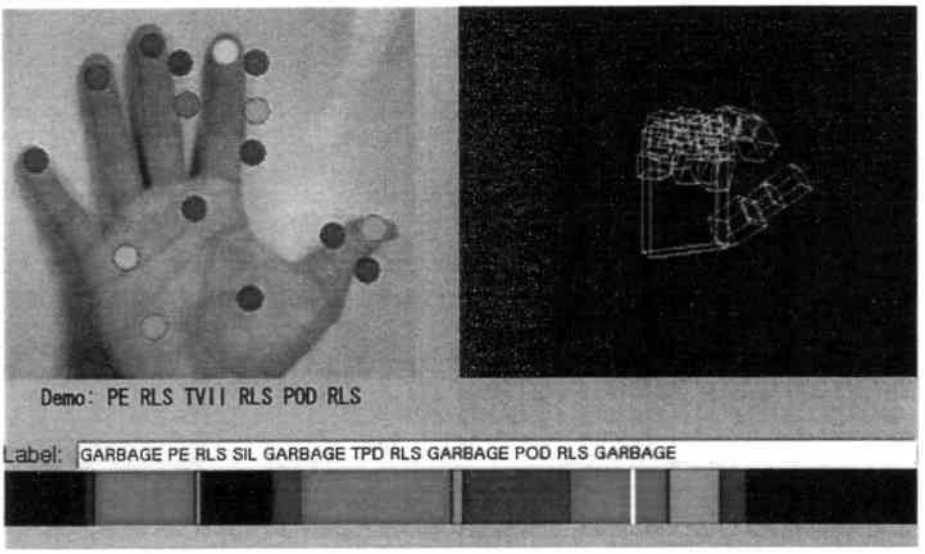

Figure 5.2: Recognition results. The upper windows show hand shape and tactile sensor activation. The colored bar shows the segmentation result. Black means silence. Red intervals are grasp, green intervals release, and blue intervals garbage segments. The recognizer hypothesis is printed above the bar.

the two windows, the label corresponding to the current demonstration is printed.
The colored bar represents the result of the segmentation and classification process. The horizontal direction shows the progression in time, starting on the left at time frame 0. Black segments represent silence. Red parts represent grasp segments, green parts represent release segments and blue parts stand for garbage. The thin white bar represents the time frame currently displayed in the upper windows. Just above the result bar, the hypothesis output by the recognition system for the current demonstation is printed.
This view allows to analyze every demonstration in detail and to find out where the system may have made avoidable or rectifiable mistakes.

5.2.2 Results for single user systems

Multiple trials were performed using different HMM sizes, starting with 5 state flat models, all the way up to 11 state models. Also, the number of

training iterations required was probed, and a variety of input feature combinations was tested. A topology with 9 states was found to yield the best results. Also, the system was found to stabilize after just 8 training iterations. The results for the different input vector configurations are shown in table 5.1.

	Ang	**Tac**	**Ang+Tac**	**Ang+AngDiff+Tac**
User 1	84.5%	76.8%	92.2%	88.7%
User 2	86.9%	82.1%	88.7%	85.1%
User 3	81.5%	67.8%	85.7%	95.2%
User 4	81.5%	57.1%	76.8%	79.2%

Table 5.1: Results for single user systems. The values represent the accuracy rate in percent. The column denoted as Ang shows the results for the system using only the finger angle values from the Cyberglove for recognition. Tac is the system using only information from the tactile sensors (and the tactile maximum value). Ang+Tac uses both finger angles and tactile data. Ang+AngDiff+Tac also uses the finger angle derivatives.

Four configurations were tested. First, the recognition accuracy using the Cyberglove alone (Column "Ang" in table 5.1) was measured. The system already achieves quite good recognition rates using only the finger joint angles. As much as 86.9% accuracy could be reached. Also quite noticeable, although quite different for every user, is the result achieved with tactile sensors alone (Column "Tac"). The accuracy ranges from 57.1% for User4 to more than 80% for User2. Although these figures seem very encouraging, one must remember that they are not too precise, since the respective test sets for every user contain just 28 demonstrations. But they show that a recognition using only one of the two input modalities is feasible up to a certain degree. As expected, much better results were achieved by the combined system using finger angle values and tactile data, reaching up to 92.2% for User 1 (Column "Ang+Tac"). The only exception is User4, for which the system using Cyberglove data alone showed better results. This could be explained by the fact that User4, as opposed to other users, made particularly loose, weak grasps, making the tactile information much less reliable for recognition (only 57.1%

achieved). The addition of the tactile information has therefore decreased the accuracy of the combined system, as compared to using finger angles alone.

For most users, the addition of finger angle differentials showed no significant improvement in system performance (Column "Ang+AngDiff+Tac" in table 5.1). The speeds at which the grasps where performed varied greatly during execution even for a same user and this information was therefore useless. One exception is User3, who took great care in executing every grasp at a constant speed, which made the finger speeds valuable features for avoiding insertion or deletion errors. In a multi-user system however, it should be impossible to guarantee such a smooth execution by all users. On the contrary, such a requirement would hurt our goal of not disturbing the naturalness of execution.

5.2.3 Results for the multiple user system

In table 5.2, the results for the multiple user system are presented. The system is trained and tested on the demonstrations from all four users.

	Ang	Tac	Ang+Tac
Users 1-4	88.8%	64.1%	90.9%

Table 5.2: Results for the multiple user system. The values represent the accuracy rate in percent. The column denoted as Ang shows the results for the system using only the finger angle values from the Cyberglove for recognition. Tac is the system using only information from the tactile sensors (and the tactile maximum value). Ang+Tac uses both finger angles and tactile data.

As one can see, again the system combining the finger angle and tactile data yields the best results, at 90.9% accuracy. Although the difference is small, this value is higher than for most single user systems. No drop in efficiency is registered when passing from single to multiple user recognition. But this should be no surprise, as demonstrations from all users were considered, multiplying the amount of training data by four. The system using only Cyberglove data, on the other hand, achieved an accuracy of 88.8%. This is because the system still has problems finding the right segmentation bounda-

ries when the users move their hands without grasping objects and produces unnecessary insertions. The addition of tactile information helps alleviate that problem. Using just tactile data, the accuracy stayed at a moderate 64.1%. The variety of ways the users grasped the objects, the different hand sizes and the resulting contact point differences did not allow to raise the system to a higher performance level.

In table 5.3, the results of the multiple user system using finger angle and tactile data are shown, when trained with all training demonstrations, but tested on each test set separately. As one can see, the results for User4 in particular have been improved dramatically, compared to those of the single user system. This may indicate the training demonstrations of User4 did not contain enough variety for a robust HMM parameter adjustment. The other results are quite similar to those of the single user systems.

	User 1	**User 2**	**User 3**	**User 4**
Users 1-4, Ang+Tac	91.1%	89.9%	90.5%	92.2%

Table 5.3: Results for the multiple user system when applied to the separate test sets. The system is trained by all users using both finger angles and tactile data. The results are quite stable, staying at about 90% regardless of the user.

5.3 Segmentation results

An important goal for our recognition system was, aside from identifying grasp types, to recognize the precise moments when the grasping and releasing motions occur. This offers the advantage that, if the time frame at which an object is grasped can be accurately determined, other input modalities such as magnetic trackers or vision systems can be subsequently used to find the position of the hand at that time and narrow down the search for the grasped object. The above section showed the quality of classification achieved. This section examines the segmentation in more detail.

5.3.1 A look at a few sample segmentations

To show the achieved segmentation quality, the output of the system for a few sample demonstrations is presented. Three cases are considered: A correct recognition trial, one where an insertion occured, and one where a deletion occured. All examples were taken from the multi-user system using finger angle and tactile data as input features.

1. Correct recognition:

   Figures 5.3 to 5.8 show the result for the correctly recognized sequence consisting of a PMF, a Lat and a POH. The black segments are silence phases, where no significant movement has been performed. The red segments are the grasping phases, the green ones the releasing phases, and the blue ones represent garbage motion.

   As can be seen from the figures, the grasp segments start at the moment the fingers close and the tactile sensors are activated. They last for all the length of the grasp. Very practical is the short length of the release segments, which mark the precise moments when the tactile sensors show no more activation. As explained in chapter 4, section 4.4.2, the release model was implemented to detect the exact moment an object is released, even in the presence of garbage motion before or after the release motion. Therefore, short segments that bound the event well are desired. As we can see in the figure, garbage motion has been detected between grasp-release groups, well constraining grasping segments. This is the general case and most demonstrations were segmented in a similar manner.

2. One insertion:

   Figures 5.9, 5.10, and 5.11 show the recognition result for a PoS, PoH, PoI sequence, where an additional PoH has been mistakenly inserted. This is because the start and end of the PoH grip were hard to recognize. For a PoH, the tactile sensors are not necessarily activated and the grasp is recognized based mostly on the hand shape. At the time the user ungrasped the object, a slight releasing move was made, but then the hand closed back a bit and another PoH was wrongfully recognized in place of silence, until the PoI started. A simple way of eliminating this problem would be to require a minimum amount of silence between

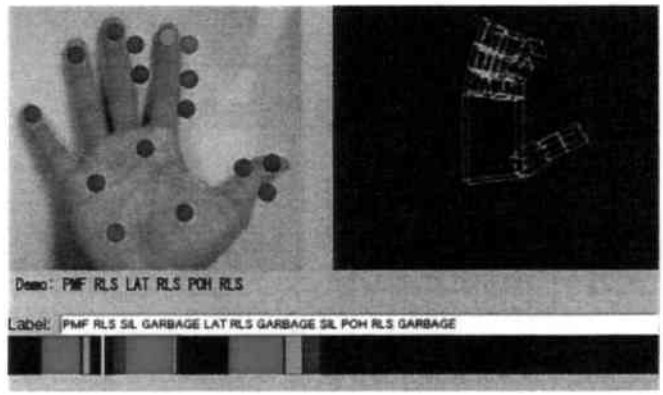

Figure 5.3: Correct classification: At this time frame, the hand does not move and only silence is detected.

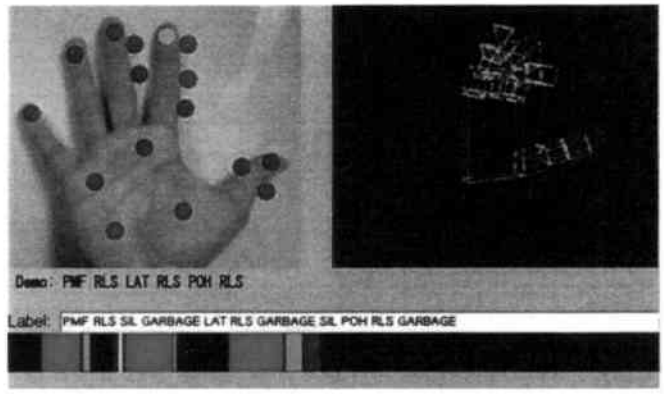

Figure 5.4: Correct classification: The fingers have started to close, but grasping itself has not yet occured. The system labels this part as garbage.

grasps, and to force the system to unify segments that are not separated by silence through the use of a strong grammar, for example. But requiring such a silence phase would definitely hurt the natural flow of execution and force the user to act in a slow, demonstrative style. Since the number of insertions is quite low (only 8 wrongfully inserted grasps for 336 demonstrated ones), this option was rejected.

3. One deletion:

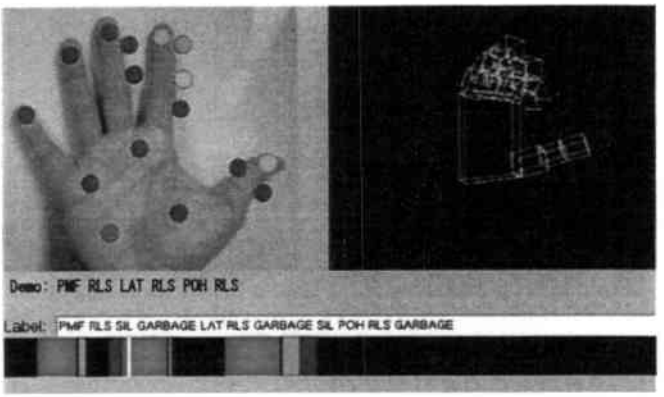

Figure 5.5: Correct classification: The tactile sensors start to show weak activation. This is the start of the grasping phase.

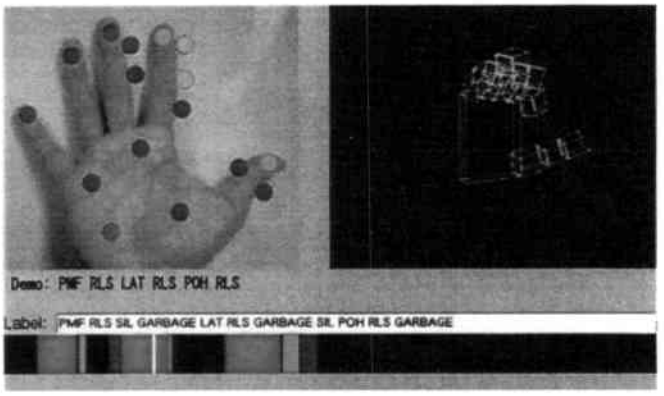

Figure 5.6: Correct classification: The tactile sensors are fully activated in the stroke of the grasp. The activation is also characteristical for a Lateral Grip, with strong activation of the middle index side sensor. The index fingertip sensor is weakly activated, as its border was also slightly in contact with the object.

Figure 5.12 shows the result for a TVI, Add, Lat sequence. Here, the Adduction Grip has been overseen. This may be attributed to many factors, such as unclean or untypical execution, insufficient activation of tactile sensors, etc. The system skips the affected time frames and recognizes the rest of the sequence correctly, starting the Lat segment

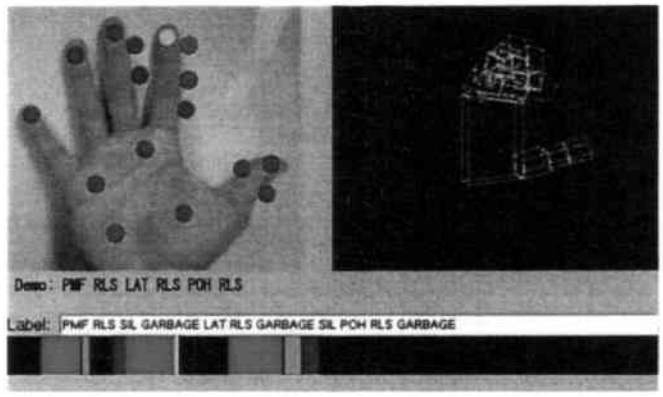

Figure 5.7: Correct classification: The release segment is placed at the moment the tactile sensors are deactivated and the fingers start to open.

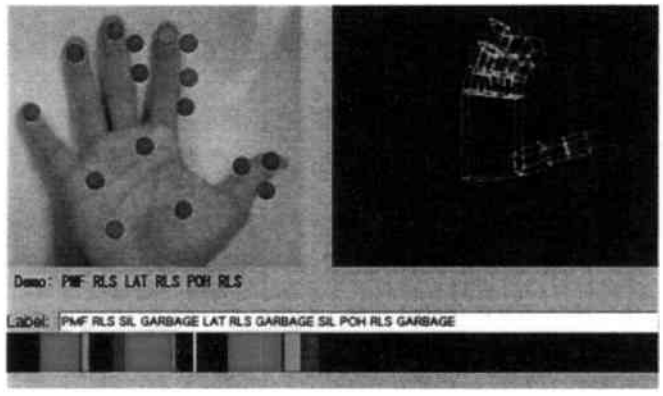

Figure 5.8: Correct classification: The fingers finish the release motion and stop moving. At this time, the user is moving his hand towards another object. This is labeled as silence.

at the moment the hand closes on the object. In total, only 7 grasps out of 336 were mistakenly ignored. The majority of deletions occured for the Adduction Grip. This grip, involving no thumb or finger tips, often produced no tactile activation at all. That's why the recognition had to be made based only on the hand shape, which does not show much difference to the resting position.

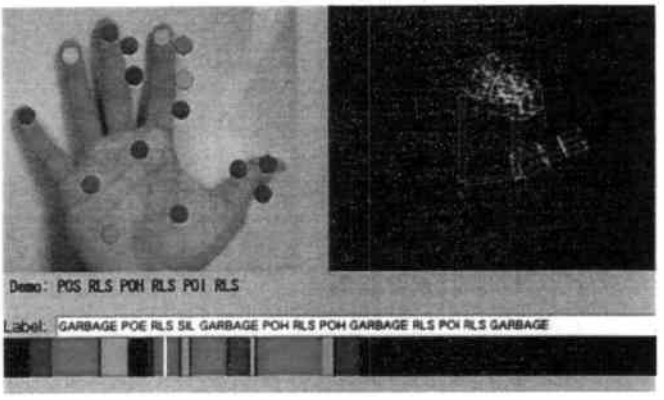

Figure 5.9: An insertion error: Lack of tactile data and unintentional movement caused the system to recognize two grasps instead of one. Here we see the shape of the hand before the object is released.

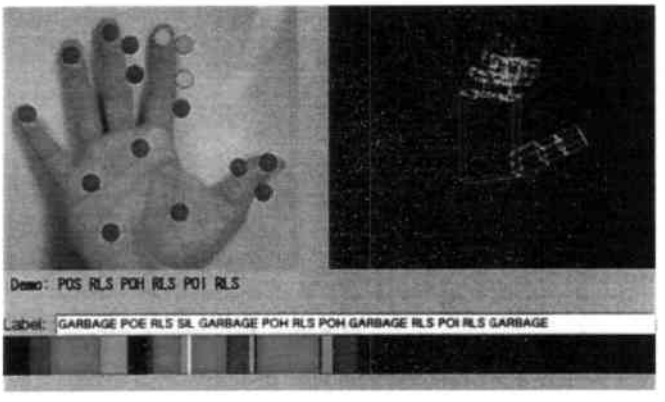

Figure 5.10: An insertion error: At this point, the object is released. Only slight finger movement is made.

5.3.2 The effect of the tactile data

Here the effect of the tactile values and their maximum on the segmentation is shown. The tactile maximum feature was introduced to allow the system to better detect the time point at which an object is grasped or released. Although information on contact is already available through the other tactile inputs, the grouping into one feature holds certain advantages.

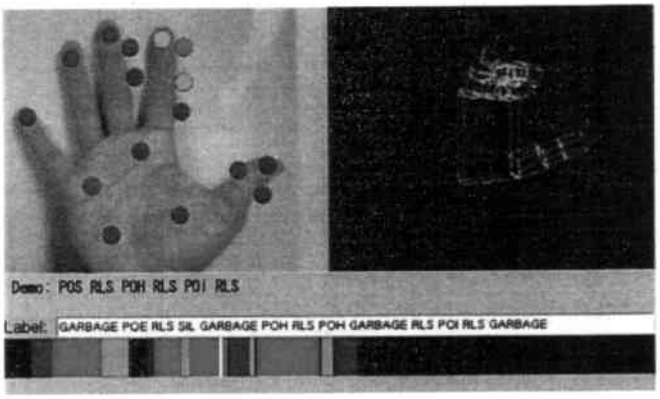

Figure 5.11: An insertion error: After the object is released, the hand takes back a shape close to the grasping shape, causing the recognition of an additional Power Grip Hook Type.

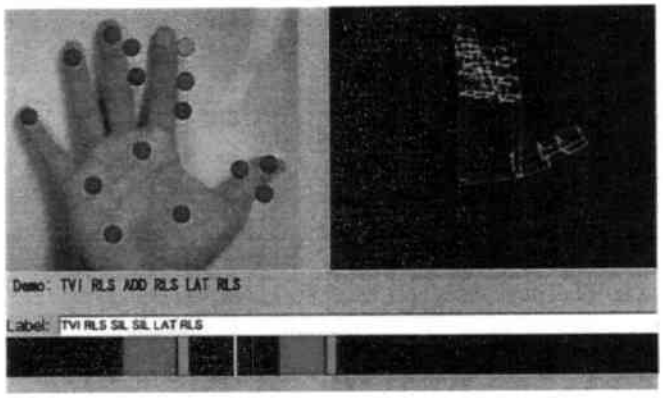

Figure 5.12: A deletion error. The Adduction grip has not been recognized by the system. The rest of the sequence was correctly recognized.

In the Tip Grip, for example, only the thumb and index tip sensors are activated. In the Lateral Grip, only the thumb tip and the index side sensors are used, and the activation level can be quite different, depending on the way the object is taken, its weight distribution, etc. So the separate values may be useful for classifying grasps, but a lot of training examples are still needed to clearly separate the feature space into grasp and ungrasp subspaces. The garbage and release models can therefore not adapt their parameters well

enough and the segmentation bounds are not precisely placed. The tactile maximum, on the other hand, provides clear information about contact, wherever it occurs. This allows a simple separation of feature space in two parts, "grasp" and "no grasp", reducing the amount of training samples needed.

To illustrate this effect, we again consider a sample recognition result from the multi-user system using finger angle and tactile features, with or without addition of the tactile maximum feature.

Figure 5.13 shows the results using finger angle data and tactile data with or without maximum, and 5.14 shows the results using or not using tactile data at all. As can be seen in figure 5.13, with the tactile maximum, the exact points when grasping and ungrasping occur have been detected and the three grasp-release groups are well separated by silence segments. When the tactile maximum is not used, the system sometimes misses the correct grasp-release bounds and tends to fill most gaps with garbage.

When no tactile information is used at all (bottom bar in figure 5.14), the grasp and release segments are often completely connected. This shows that not the points where grasping and ungrasping occur are actually detected, but rather the points where a grasp motion becomes more probable than a release motion and vice versa. These boundaries can even be set in between grasps, where no motion occurs at all.

5.3.3 The effect of the garbage model

To show the effect of the garbage model on segmentation, the recognition result of the multi-user system using finger joint angles and tactile data on a Pod, PMF, PoS sequence is presented in figure 5.15. If the system is trained with clean labels (the garbage model is not trained at all), the segmentation boundaries are not well found. The grasp and release segments are very lengthy and there is almost no pause from the time of release to the next grasp. This is because, when the fingers do not stop moving, no silence can be detected. Instead, the system tries to assign all movement to a grasp or release class, even if the probability for this class is very low.

If a well chosen threshold on the probabilities were applied, this could be avoided. The garbage model serves this purpose, as its probability is higher

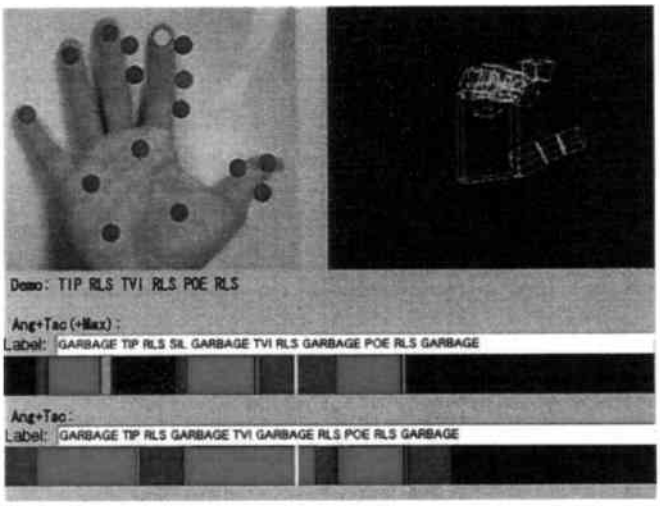

Figure 5.13: The effect of the tactile maximum. At the current time frame, the user had already released the object after a TVI Grip. The system using tactile maximum (top result bar) correctly recognized that fact. The system without tactile maximum (bottom result bar) wrongfully placed the end of the grasp segment some time after the object had already been released. Silence phases are also not well distinguished.

than that of other grasps in these movement phases. Using the technique of automatically inserting GARBAGE labels between release and grasp labels before training, the garbage model is always trained on a small part of the demonstration sequence. If there actually is garbage in the demonstration, many frames are used in training. If there is none, only small fragments at the beginning of grasp moves or at the end of release moves are used. Still, thanks to the complexity of the model, a good adaptation can be achieved by the Baum-Welch algorithm. The results are much more compact grasp-release groups, and better placed segmentation bounds.

Another good example of the usefulness of the garbage model is shown in figures 5.16 to 5.19. After executing the CMF grip, the user releases the object. His hand, however does not stop moving, and does not go back into a resting position. Instead, it preshapes into what may seem like a Power Grip Index Type (PoI) before closing on the object in a Tripod Grip. The system

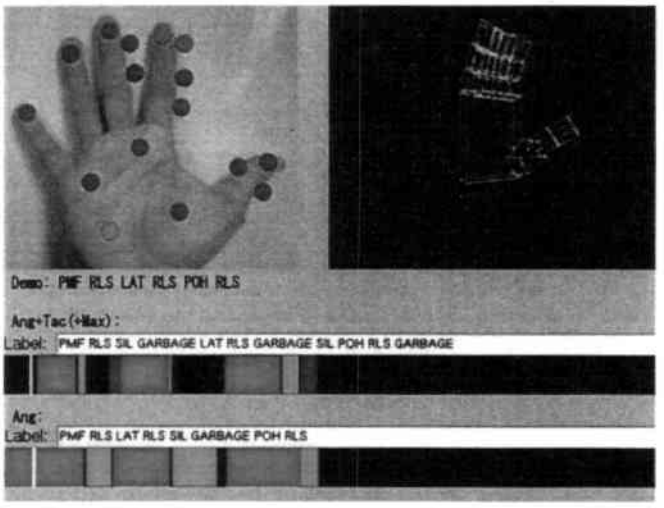

Figure 5.14: The effect of tactile data on segmentation. At the current time frame, the PMF grasp is not executed yet. The system using only finger joint angles (bottom result bar) failed to recognize this. It generally tends to place the starting bounds for the grasp segments too soon, producing mostly connected grasp-release groups.

correctly recognizes this preshaping motion as garbage (not as a PoI) and puts the starting point for the next grasp segment at the time the fingers close and the tactile sensors are activated. The system did make a classification mistake by identifying a TVI instead of a Tpd, two very similar grasps, but the segmentation did not suffer.

5.4 Analysis and Discussion

The results show that the main advantage of the tactile sensors is in achieving a good segmentation. The system using only the finger angle data achieves good classification, but suffers from noise, unintentional hand movement, too small changes in hand shape, etc. Also, we can see that even with a reduced set of tactile sensors (not using the middle and pinkie finger tip, the thumb sides, etc), we get a very high recognition accuracy. This suggests that very rich and detailed tactile information is not necessary. Some regions that

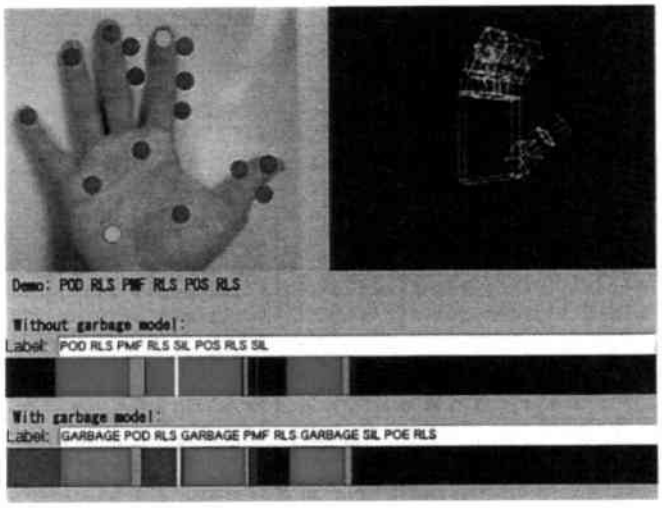

Figure 5.15: The effect of the garbage model on segmentation. At the current time frame, the grasp is not yet made. The system without garbage model failed to recognize this.

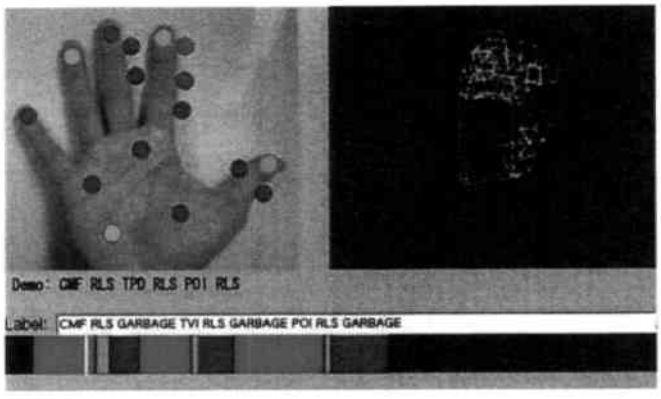

Figure 5.16: The effect of the garbage model: At the current time frame, the user is still holding an object with a Circular Mild Flexion Grip (CMF).

were separated could perhaps be grouped (for Ex. on the palm or the finger sides). The most important thing is to insure contact is registered wherever it occurs. Therefore, large, flat, sensitive sensors, that cover as much surface as possible should be well suited to the task. It can also be seen that the garbage

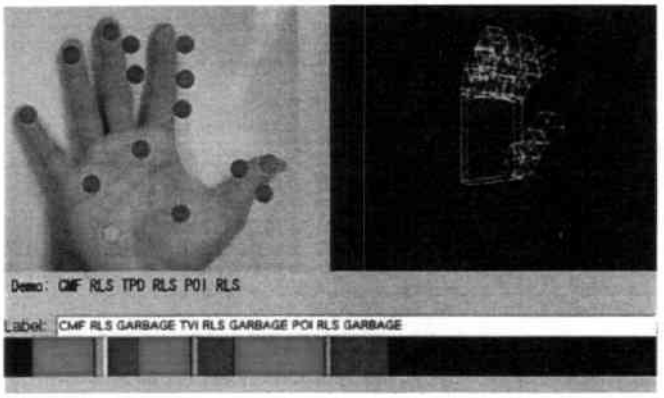

Figure 5.17: The effect of the garbage model: Here, the user released the object. This also shows the usefulness of the tactile sensors. The user made almost no finger movement at the time of release. Only the tactile information allowed to detect it.

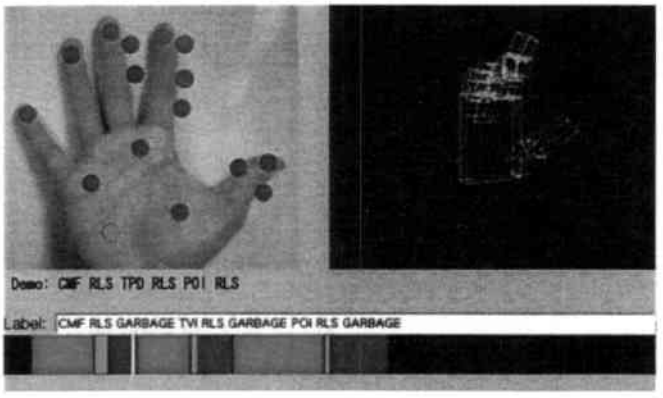

Figure 5.18: The effect of the garbage model: The fingers do not stop in a resting position, but quickly take a shape that resembles a Power Grip Index Type (POI). The system classifies this preshape phase as garbage.

model works well in conjunction with tactile information. If the tactile data is left out, the segmentation quality suffers. The same happens if tactile input features are used but the garbage model is left out. Only a combination of both could bring good results.

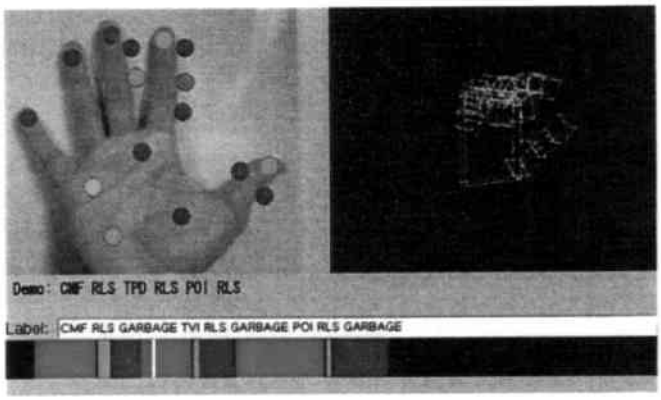

Figure 5.19: The effect of the garbage model: Only now does the user close his fingers to grasp a new object with a Tripod Grip.

However, alone the absence of a garbage model, or of tactile data, is not enough to explain all the segmentation errors made. It should be remembered here that a very simple labeling strategy, one that does not provide segmentation boundaries for training, is used. Instead, only a transcript of the executed demonstration is provided, i.e. a sequence of symbols without attached temporal information. The start and stop frames of grasp and release moves are not given beforehand, they are iteratively estimated during Baum-Welch training of the HMMs. This means that if a grasp is almost always preceded by a phase of slight finger motion in the training data, the system might mistakenly take this for the starting part of the grasp and include it in the training of the corresponding model. This can prevent correct detection of the beginning of even those grasps for which a closing motion of the fingers clearly indicates the start. When very large training databases are available, there should be enough variability to ensure the true bounds of the grasp are found. But this is not the case here. A solution could be to manually segment training labels by replaying the demonstrations offline and carefully identifying grasps, garbage, silence, and their start and stop points. But this is long and painstaking work, and is avoided whenever possible. The use of the garbage model and tactile information allows to obtain good results with only limited training data, and without resorting to manual labeling.

At this point, a few things should be noted about the naturalness of execu-

tion.

First, it is interesting to notice how the users adapted their grasps automatically to the object and its assumed purpose. Instead of asking to perform a specific grasp, it was easier to present the right objects to the user directly. Of course, sometimes, instruction was still necessary: It is not really clear if a cup should be taken with a PoS or a PMF, if no further manipulation is planned. Even when specifically asked to perform a Tripod Grip (TPD), if the user was not presented with a pen, but with a larger cylindrical object of unclear purpose, he or she sometimes inadvertently shifted into a Tripod Variation I (TVI). The two grips are used on very similar objects and when a clear purpose is not given, there is no criterion to choose between them.

The main restriction came from the input devices. The gloves did impair the movements of the users, despite efforts to keep this factor as low as possible. The users had difficulty picking up small objects, such as coins, directly from the table. They also experienced some difficulty making tripod grips on smaller objects. Paticularly the Tripod Variation II was found very hard to execute. Furthermore, some users complained that their dexterity is reduced, and objects tend to slip out of their hands. This seems obvious, as the tactile sensors are placed on the finger contact surfaces. But not all the difficulty comes from this fact. The tactile sensors are attached to the inner side of a thin stretchable glove, and this glove is worn under the Cyberglove, to measure the combined input. This design has a few advantages: The cables for the tactile sensors pass on the back of the user's hand inside the gloves, and cannot get entangled with objects outside. This also insures that the tactile sensors are placed quite precisely at the right points on the hand, regardless of the user. But the disadvantage is that the user has to wear two gloves instead of one. Since the Cyberglove has open fingertips, the problem is not acute at those points, but particularly the palm region feels bulky and uncomfortable, and the finger side sensors are harder to activate. Placing the tactile sensors outside, directly on top of the Cyberglove could increase the user comfort, but would expose them much more to damage from the outside. A custom designed glove, that would incorporate both joint angle measurement and contact point detection sensors inside the glove fabric would be ideal.

Finally, The type of manipulations the users performed with the objects also reduced the execution naturalness somewhat. It is a known phenomenon

in speech recognition, that users tend to change their communication style somewhat when they know they are interacting with a machine. Here, the users knew that the focus was not on performing real manipulations, but to show grasp examples to the robot system. Objects were simply picked up at one place and put down at another. Some symbolic movement was sometimes made while grasping the object (such as scooping with a spoon), but a key pinched in a Lateral Grip was not, for example, really used to open a lock, a ruler was not used to actually measure, a book was not actually taken and put back on a shelf. This surely resulted in slightly different grasping movement and the overall effect on recognition accuracy has yet to be investigated.

Chapter 6

Summary and future work

6.1 Conclusions

- The main objective stated in chapter 3 has been reached. As shown in chapters 4 and 5, a Hidden Markov Model recognizer for continuous recognition of grasping gestures has been created. The techniques developed for speech recognition and succesfully applied to sign gesture recognition have been adapted to the domain of grasping gestures. By combining finger joint angle data obtained from a Cyberglove device, and information about hand-object contact points obtained from an array of tactile sensors, a high degree of accuracy could be reached.

- The principal mechanism needed to achieve this continuous recognition is the Hidden Markov Model. By using a dynamic programming algorithm, the Viterbi algorithm, whole sequences of grasps can be analyzed at once. The segmentation bounds are placed at the same time classification decisions are being made, while maximizing the probability of the output hypothesis.

- While the global aim is to achieve recognition of any type of naturally executed object manipulation, a restriction to grasp and release sequences has been made here. No manipulation movement while the object is grasped and no grasp transitions are considered. Also, the length of the analyzed sequences has always been limited to three grasps. This latest restriction, however, was made only for convenience of recording the demonstration data and is not enforced by the task

grammar; the same system could be used to recognize grasp sequences of any length. Apart from this, the users are free to make their demonstrations in a natural way, paying no attention to the start or end shape of their hands when grasping, to the speed with which they grasped, and without having to make explicit pauses betwen grasps.

The thickness of the data gloves does reduce dexterity somewhat, and makes picking up and handling of some objects difficult. Still, no restrictions are made on the kinds of objects to be manipulated. A multitude of objects commonly handled in everyday life are considered, the Kamakura classification table providing an appropriate class for every type of grasp used on them. Most of the 14 grasp classes can be correctly recognized, with exception of the Adduction Grip, which is sometimes hard to detect and the Tripod Variation II, which is hard to execute with the data glove and sometimes gets confused with other tripods.

- Recognition has been achieved for multiple users, the results surpassing, on average, those of the single user systems. This shows that the Hidden Markov Model recognizer can robustly handle all the input variability coming from different hand shapes and sizes, grasping styles and strengths, and execution speeds. The system has been so far trained and tested for four users, but the results indicate that a grasp recognition system applicable to any kind of user operating in unstructured environments is feasible.

- Because of the small number of grasps considered and the relatively simple HMM topologies used, the system is quite compact and fast. An exact measurement of the recognizer speed was not made, but the evaluation of the test demonstrations on a 500MHz Pentium III machine was made more than five times faster than real time. Since the purpose is teaching manipulation tasks by showing, online recognition is not a priority. The robot can wait for the end of the demonstration and then analyze it, before repeating the task. But the actual speed of the offline recognizer shows that the development of an online system is feasible, in principle.

6.2 Summary

A system to recognize continuously executed sequences of grasping gestures has been presented. The main difference to most other systems for analysis and classification of gesture sequences is that they are designed only for the recognition of communicative gestures, for which the requirements are different. The few existing systems that consider manipulative gesture sequences either ignore the types of grasps used, or analyze them in a two step fashion, first segmenting the sequences and then classifying grasps. Using Hidden Markov Models, a system was designed for manipulative gestures, that both detects the grasping phases in a user demonstration and classifies them with a single, statistically sound approach. For a total of 14 grasp classes, a recognition accuracy of up to 92.2% for a single user system and 90.9% for a multiple user system was reached.

A glove based approach was used to capture the user demonstration. Both a Cyberglove and an array of pressure sensitive capacitive sensors was used to gain precise information about the shape of the hand and its contact points with grasped objects. The influence of the tactile sensors on the recognition was analyzed and they were found to be particularly useful in improving the quality of segmentation, when used in combination with a well designed garbage model. While a system using no tactile information was still able to produce good classification results, the precise bounds for the grasp segments could not be accurately set. Whereas this may not be crucial for communicative gestures, in a programming by demonstration scenario, the time point where a grasp is made may well be a valuable piece of information for subsequent inference steps.

Kamakura's grasp classification table, containing 14 graps classes, was used. It considers grasping techniques for all objects used in everyday life. The fact that the grasp classes could be well distinguished in tests, with the given input devices, even for a multiple user system, and with such a broad range of objects, shows that the taxonomy is indeed well adapted to the task. In total, 112 user demonstrations were used for the training of the multiple user system. Considering the number of Hidden Markov Model parameters to adjust, this is still a relatively small amount, which shows that the system is able to learn quickly and adapt robustly to noisy data, even with little training.

6.3 Future work

While the current system concentrates on grasp-release sequences, it should be interesting, in the future, to consider also grasp transitions, i.e. changes in the way an object is held, without releasing it. This should be a simple extension of the current Hidden Markov Model based design. Only the task grammar must be adapted, new training demonstrations recorded, and the corresponding labels created. As no release move is made, the time point of transition cannot be detected by the presence or absence of contact, but if the contact point distribution changes in between grasps, the tactile sensors can still be useful for finding the correct segmentation bounds.

It should also be possible, with the current system, to recognize both manipulative and communicative gestures. This could simplify teaching the robot system: the user would first make signs to give instructions to the robot (for example a sign to indicate the beginning of the demonstration), then start manipulating objects, demonstrating the task, and then give additional signs, for example to indicate how strongly or carefully the robot should grasp objects. Although the tactile sensors are not needed when making signs, the same basic system should be usable. The model for the sign gesture would just learn to ignore the tactile inputs and rely only on the finger joint angles. On the other hand, one could recognize non-prehensile grasps, where the object is touched, but not held, and where an analysis based on hand shape alone would not be possible. This happens, for example, when pressing a button, pushing a box, when holding down a lid, etc.

Humans very often use both hands when manipulating objects. Often, an object is first picked up at an easily reachable part with one hand, before it is held tightly at the correct spot with the other hand. Often, actions are performed with two objects at once: holding a nail with one hand and the hammer with the other, holding a bowl firmly while cleaning it with a tissue. If complex manipulations are to be analyzed, the movement of both hands should be considered.

Concerning the design of the Hidden Markov Model recognizer itself, some future extensions should also prove advantageous. At present, only a simple context independent recognizer is used. A context dependent version could increase accuracy, especially for the recognition of grasp transitions. The

beginning and end parts of a grasp vary greatly depending on the moves preceding and following it, but the most important part, the nucleus or stroke, stays mostly the same. Thus, context dependent modeling could yield better results when the grasps are not clearly separated by silence.

One could also, as is done in many speech recognition systems, let the recognizer adapt progressively to a new user while he is performing demonstrations. The recognition results of the first few demonstrations would be used to run a quick Baum-Welch reestimation and adjust the Hidden Markov Model parameters to better fit the new user, resulting in better recognition of subsequent demonstrations. This reduces the negative effects of different user hand shapes, for example, which cause a constant bias in the input features. Also, it allows to adapt to the user's style of grasping (for example, some users systematically extend the free fingers when performing a Tip Grip, some flex them).

Finally, one could incorporate and fuse other input modalities, such as vision. The distance from the hand to the closest object lying on its current path could, for example, be detected by a vision system and used as feature. When the object is grasped, the distance becomes very small, and stays small until it is released again. Such a feature could be useful to recognize that the object has not been released, even if the tactile sensors are temporarily deactivated, for example because the object is shifted to a new position between the fingers. The distance from the hand to the table top or to other objects, while an object is held, could also be used to distinguish, for example, if an object has been simply "released" or more precisely "put down".
Theoretically, any kind of input modality can be added (also speech), as long as it produces a well defined set of features to be used as input for the Hidden Markov Model recognizer. The richer the variety of inputs, the more robust the system becomes. Of course, the more input features are added, the bigger and slower the system also becomes, and the more training examples are needed to adjust its parameters. Thus, it will always be a matter of the expert's skill in designing the recognizer, to distinguish which input features are useful and which are not, which preprocessing, which filtering is needed, etc... until a method can be found to somehow select the input features automatically.

[9] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification, Second Edition.* John Wiley & Sons, Inc., New York, USA, 1999.

[10] M. Ehrenmann, T. Luetticke, and R. Dillmann. Dynamic gestures as an input device for directing a mobile platform. In *Proceedings of the 2001 International Conference on Robotics and Automation (ICRA)*, volume 1, May 2001.

[11] M. Ehrenmann, O. Rogalla, R. Zoellner, and R. Dillmann. Teaching service robots complex tasks: Programming by demonstration for workshop and household environments. In *Proceedings of the 2001 International Conference on Field and Service Robots (FSR)*, volume 1, pages 397–402, June 2001.

[12] M. Ehrenmann, R. Zoellner, S. Knoop, and R. Dillmann. Multi sensor fusion approaches for programming by demonstration. *International Conference on Multi Sensor Fusion and Integration for Intelligent Systems (MFI)*, pages 227–232, 2001.

[13] H. Friedrich, V. Grossmann, M. Ehrenmann, O. Rogalla, R. Zoellner, and R. Dillmann. Towards cognitive elementary operators: Grasp classification using neural network classifiers. In *Proceedings of the IASTED International Conference on Intelligent Systems and Control (ISC)*, volume 1, October 1999.

[14] R. D. Howe. Tactile sensing and control of robotic manipulation. In *Journal of Advanced Robotics*, volume 8, pages 245–261, 1994.

[15] T. Iberall, G. Bingham, and M.A. Arbib. Opposition space as a structuring concept for the analysis of skilled hand movements. In H. Heuer and C. Fromm, editors, *Experimental Brain Research Series 15, Generation and Modulation of Action Patterns*, pages 158–173. New York: Springer Verlag, 1986.

[16] J.Weissmann and R. Salomon. Gesture recognition for virtual reality applications using data gloves and neural networks. *International Joint Conference on Neural Netwoks*, 1999.

[17] N. Kamakura. *Te no ugoki, Te no katachi (Japanese).* Ishiyaku Publishers, Inc., Tokyo, Japan, 1989.

[18] N. Kamakura, M. Ohmura, H. Ishii, F. Mitsubosi, and Y. Miura. Patterns of static prehension in normal hands. In *Amer. J. Occup. Ther.*,, volume 34, pages 437–445, 1980.

[19] S. B. Kang and K. Ikeuchi. Toward Automatic Robot Instruction from Perception - Recognizing a Grasp from Observation. *IEEE Transactions on Robotics and Automation*, 9(4):432–443, Aug. 1993.

[20] S. B. Kang and K. Ikeuchi. Toward Automatic Robot Instruction from Perception - Temporal Segmentation of Tasks from Human Hand Motion. *IEEE Transactions on Robotics and Automation*, 11(5):670–681, Oct. 1995.

[21] S. B. Kang and K. Ikeuchi. Toward Automatic Robot Instruction from Perception - Mapping Human Grasps to Manipulator Grasps. *IEEE Transactions on Robotics and Automation*, 13(1):81–95, Feb. 1997.

[22] Sing Bing Kang. Robot instruction by human demonstration. *The Robotics Institute, Carnegie Mellon University Pittsburgh, Pennsylvania 15213*, December 1994.

[23] H. Kawasaki, K. Nakayama, and S. Ito. On a better hand model in virtual teaching for multi-fingered robots. *6th International Conference on Virtual Systems and Multimedia*, Oct. 2000.

[24] Y. Kuniyoshi, M. Inaba, and H. Inoue. Learning by watching: extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation*, 10(6):799–822, Dec. 1994.

[25] Yasuo Kuniyoshi and Hirochika Inoue. Qualitative recognition of ongoing human action sequences. *International Joint Conference on Artificial Intelligence*, pages 1600–1609, 1993.

[26] Christopher Lee and Yangsheng Xu. Online, interactive learning of gestures for human-robot interfaces. *IEEE International Conference on Robotics and Automation*, 4:2982–2987, 1996.

[27] H.K. Lee and J. H. Kim. An hmm-based threshold model approach for gesture recognition. In *IEEE Transaction on Pattern Analysis and Machine Intelligence*, volume 21, pages 961–973, 1999.

[48] A.D. Wilson and A.F. Bobick. Realtime online adaptive gesture recognition. *International Conference on Pattern Recognition*, September 2000.

[49] Ying Wu and T. S. Huang. View-independent recognition of hand postures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2000)*, volume 2, pages 88–94, June 2000.

[50] R. Zoellner, O. Rogalla, and R. Dillmann. Integration of tactile sensors in a programming by demonstration system. In *IEEE International Conference on Robotics and Automation*, May 2001.

[51] R. Zoellner, O. Rogalla, R. Dillmann, and J.M. Zoellner. Dynamic grasp recognition within the framework of programming by demonstration. *10. IEEE International Workshop on Robot and Human Interactive Communication*, Sep. 2001.