



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
POSGRADO EN CIENCIA E INGENIERÍA
DE LA COMPUTACIÓN

“BEHAVIOUR RECOGNITION SYSTEM BASED ON HIDDEN MARKOV MODELS”

TESIS

QUE PARA OPTAR POR EL GRADO DE:
DOCTOR EN CIENCIAS
(COMPUTACIÓN)

PRESENTA:
JOSÉ ISRAEL FIGUEROA ANGULO

DIRECTOR DE TESIS: JESÚS SAVAGE CARMONA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA
UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

CODIRECTOR: ERNESTO BRIBIESCA CORREA
DEPARTAMENTO DE CIENCIAS COMPUTACIONALES
UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

MÉXICO, D. F.

ENERO, 2016



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

Ahora que llego al término de este período de vida como aspirante al Doctorado en Ciencias de la Computación, me doy cuenta de que es el momento de hacer dos cosas: Agradecer y Recordar

Agradecer a todas las personas que a lo largo de estos años estuvieron a mi lado brindándome sus saberes, consejos, regaños, apoyo y amistad, llevándome a enfrentar con éxito los retos y obstáculos que implica el crecer personal y profesionalmente.

Recordar los momentos vividos, unos agradables y satisfactorios que me llenaron de confianza y tranquilidad para seguir adelante. Otros, desagradables, tristes o molestos que me cuestionaron sobre si era o no conveniente seguir en el camino. Ahora al voltear hacia atrás, me doy cuenta que unos y otros se convirtieron en vivencias que me enriquecieron y motivaron para llegar hasta aquí.

Especial agradecimiento y recuerdo:

A mis Padres, José Noé Figueroa Solís, y María Guadalupe Angulo Aguirre.

A mis Asesores, los Doctores Jesús Savage Carmona, y Ernesto Bribiesca Correa.

Mis Compañeros de Laboratorio.

Al CONACyT por haberme concedido la beca del doctorado.

Contents

Contents	v
List of Figures	ix
List of Tables	xi
1 Introduction	1
1.1 Objective and Contribution	1
2 Basis and Background	3
2.1 Human Kinematics	3
2.2 Human Activity Recognition	9
2.2.1 Approaches to Activity Recognition	10
2.2.2 Hidden Markov Models	11
2.2.3 Hidden Markov Model Topologies	17
2.2.4 Activity Recognition with Hidden Markov Models	19
3 Proposed Approach	23
3.1 Skeleton Features	23
3.1.1 Interjoint Distance	23
3.1.2 Interjoint Distance Feature Classification	24
3.2 Compound Hidden Markov Model	25
4 Implementation	27
4.1 Organization of the Behaviour Recognition System	27

4.2	Software Used for Development	28
4.2.1	C++ programming language	28
4.2.2	Eigen library	28
4.2.3	Boost C++ libraries	28
4.2.4	UMDHMM library	30
4.2.5	OpenNI® library	30
5	Tests and Results	33
5.1	Data Source	33
5.2	Training	34
5.2.1	Computing the Codebook	34
5.2.2	Computing the Observations	36
5.2.3	Building the Compound Hidden Markov Model	36
5.3	Testing Activity Labelling.	37
5.3.1	Assessing Labelling Accuracy	37
5.4	Results	39
6	Conclusions and Future Work	45
	Appendix A Orthogonal Direction Change Chain Code	47
	Bibliography	55

Behaviour Recognition System Based on Hidden Markov Models

por

José Israel Figueroa Angulo

Resumen

Este trabajo presenta un método novedoso para representar movimiento humano y etiquetar actividades humanas, a partir del esqueleto generado a partir de los datos RGB-D, en un sistema de captura de movimiento basado en visión. El método usa una representación del esqueleto que es invariable a la translación y a la rotación, basada en la distancia Euclidiana entre ciertas articulaciones del cuerpo. Los vectores de distancias entre articulaciones se usan como patrones de referencia para clasificar cuadros de movimiento. Los cuadros de movimiento clasificados sirven como observaciones para un Modelo Oculto de Markov. El etiquetado de actividades en una secuencia de movimiento se hace con un Modelo Oculto de Markov Compuesto. El Modelo Oculto de Markov Compuesto está formado de varios Modelos Ocultos de Markov Lineales, cada Modelo Oculto de Markov Lineal modela una actividad usando 4 estados, que se conectan a un conjunto de estados comunes. Cuando el Modelo Oculto de Markov Compuesto analiza las observaciones del movimiento de una actividad humana, la secuencia de estados más probables indica las actividades que fueron ejecutadas por una persona en un intervalo de tiempo. El propósito de este trabajo es que un robot de servicio sea capaz de identificar las actividades que ejecuta una persona, para que pueda planear acciones sin esperar a un orden verbal. El Modelo Oculto de Markov Compuesto propuesto en este trabajo etiqueta actividades con mayor precisión que un Modelo Oculto de Markov Ergódico y que un Modelo Oculto Markov donde las actividades son representadas por un estado.

Behaviour Recognition System Based on Hidden Markov Models

by

José Israel Figueroa Angulo

Abstract

This research presents a novel way of representing human motion and labelling human activities from the skeleton output computed from RGB-D data of vision-based motion capture systems. The method uses a representation of the skeleton which is invariant to rotation and translation, based on Euclidean distances between certain joints of the body. The vectors of distances between joints become into reference patterns for observation symbols of a Hidden Markov Model. The Hidden Markov Model used for labelling activities is a Compound Hidden Markov Model. The linking of several Linear Hidden Markov Models, one model per activity, to common states make a Compound Hidden Markov Model. Each separate Linear Hidden Markov Model has motion information of a human activity. When the Compound Hidden Markov Model processes the observations from the motion of a human activity, the sequence of most likely states indicates which activities were performed by a person in an interval of time. The purpose of this research is to provide a service robot with the capability of recognising human activity, which can be used for action planning without waiting for a spoken order. The proposed Compound Hidden Markov Model, made of a Linear Hidden Markov Model per activity to recognise, labels activities with more accuracy than an Ergodic Hidden Markov Model or a Hidden Markov Model where activities are modelled by a single state.

List of Figures

2-1	The anatomical position, with three reference planes and six fundamental directions. Source: Gait Analysis: An Introduction, by Michael Whittle, © 2002.	5
2-2	Movements at the Frontal Plane. Figures based on material from Basic Biomechanics, (6th edition), by Susan J. Hall, ©2011 McGraw-Hill Publishing.	6
2-3	Movements at the Saggital Plane. Figures based on material from Basic Biomechanics, (6th edition), by Susan J. Hall, ©2011 McGraw-Hill Publishing.	7
2-4	Movements at the Transverse Plane. Figures based on material from Basic Biomechanics, (6th edition), by Susan J. Hall, ©2011 McGraw-Hill Publishing.	8
2-5	Hidden Markov Model.	12
2-6	Hidden Markov Model Topologies.	18
3-1	Skeleton Features.	24
3-2	Compound Hidden Markov Model for activity labelling.	26
3-3	An activity has three sections: the motion preceding the activity(Anticipation), the motion of the activity (Action) and the motion after the activity is performed (Reaction). Source: Animal Locomotion, Vol. 1, Plate 154, by Eadweard Muybridge, ©1887.	26
4-1	Overall Workflow of the Behaviour Recognition System.	29
4-2	Three-dimensional Joint Data Hierarchical Structure.	31
5-1	Subset of activities from Microsoft Research Daily Activity 3-D used in this work.	35
5-2	Hidden Markov Models for Activity Labelling.	38

A-1	Orthogonal Direction Change Chain Elements.	48
A-2	Mirror of a ODC Chain Code sequence.	50
A-3	Example of ODC Chain Code Sequence.	51
A-4	Three-dimensional Joint Data Hierarchical Structure.	52
A-5	Joint Order for a Skeleton Chain Code Signature.	53

List of Tables

5-1	Criteria for Sequence Accuracy per Activity.	39
5-2	Average labelling accuracy for the Hidden Markov Models with highest accuracy.	41
5-3	Results on Activity Labelling Accuracy for Inter-joint Distance Features (Training Set).	42
5-4	Results on Activity Labelling Accuracy for Inter-joint Distance Features (Testing Set).	43

Chapter 1

Introduction

The journey of a thousand miles begins
with a single step.

Lao Tzu

In daily life, human beings perform activities to accomplish diverse tasks at different times throughout the day. These activities are by one or several simpler actions which are performed at different times, and these simple actions have a chronological relationship to each other.

1.1 Objective and Contribution

The objective of this work is to research and develop a software system based Computer Vision and Machine Learning for identifying the behaviour of a person from his/her movements. The system will use a Hidden Markov Model (HMM) as a model for analyse and identify actions from motion data, which can be used by a domestic service robot as another source of data for action planning. The contribution of this project is to provide a mobile domestic service robot with the ability of interacting with people in a more natural way, by identifying the behaviour of a human being from his/her motion. For such objective, the mobile domestic service robot must be able of learning and identifying the motion which is performed by a human being, in addition to associating those motions to human actions, and assign to those human actions the adequate responses for robot action planning. The proposed method for activity recognition consists of three stages: (1) a capture stage, where the three-dimensional joint data, which

computes the OpenNI®[OpenNI, 2009] library from depth map captured by the Kinect™ sensor, is converted to skeleton features, to provide a simpler and view-invariant representation of the joint data. The skeleton feature data is used as input for the next two stages: (2) a training stage, where the a reference set of skeleton features is arranged in a codebook of key frames, which is used to get the observations for training a set of Hidden Markov Models which recognise simple activities, which are merged into a single Discrete Compound Hidden Markov Model for continuous activity recognition; and (3) a recognition stage, where the sequences of skeleton features are classified against the codebook of key frames, the indices of the classified features are appended to a sequence of observations, which is passed to the single Discrete Compound Hidden Markov Model, which is computed in the training stage, to label actions by computing the Viterbi Path from a sequence of observations.

The thesis is organized as follows: in Chapter 2 the topics are Anatomical Theory on Human Kinematics, Theory on Human Activity Recognition, Introduction to Hidden Markov Models, and applications of Hidden Markov Models to Human Activity Recognition; the Chapter 3 describes the features for clustering and classifying the skeletons, as well as the Compound Hidden Markov Model for Human Activity Recognition and Labelling; the Chapter 4 presents the practical implementation of the skeleton features, the skeleton classification algorithm, and the Compound Hidden Markov Model; the Chapter 5 presents the methodology and the results of the tests to assess the quality of the implementation, and, the Chapter 6 presents the conclusions of the research, and an outline for the future work.

Chapter 2

Basis and Background

Never confuse Motion with Action.

Benjamin Franklin

2.1 Human Kinematics

In order to understand human activity, the first step consists in understanding how the human body moves when it performs an activity.

The anatomical terms which describe the relationships between parts of the body are based on the anatomical position, in which a person is standing upright with feet together and the arms by the side of the body (Figure 2-1) [Whittle, 2002]. Six terms are used to describe directions with relation to the center of the body. These are defined by example [Whittle, 2002]:

1. The umbilicus is *anterior*.
2. The buttocks are *posterior*.
3. The head is *superior*.
4. The feet are *inferior*.
5. *Left* is self-evident.

6. So is *right*.

The motion of each limb is described using the reference planes [Whittle, 2002]:

1. A *sagittal* plane divides a part of the body into right and left portions; the *median* plane is the mid-line sagittal plane which divides the full body into right and left halves.
2. A *frontal* plane divides a body part into front and back portions, this plane may also be called the *coronal* plane.
3. A *tranverse* plane divides a body part into upper and lower portions, this plane may also be called the *horizontal* plane, although this definition applies when in the standing position.

Most joints can only move in one or two of these planes. The possible movements are as follows [Whittle, 2002]:

1. Abduction and adduction take place in the frontal plane (Figure 2-2).
2. Flexion, extension and hyper-extension take place in the sagittal plane (Figure 2-3).
3. *Internal* and *external* rotation take place in the transverse plane; they are called *medial* and *lateral* rotation respectively (Figure 2-4).

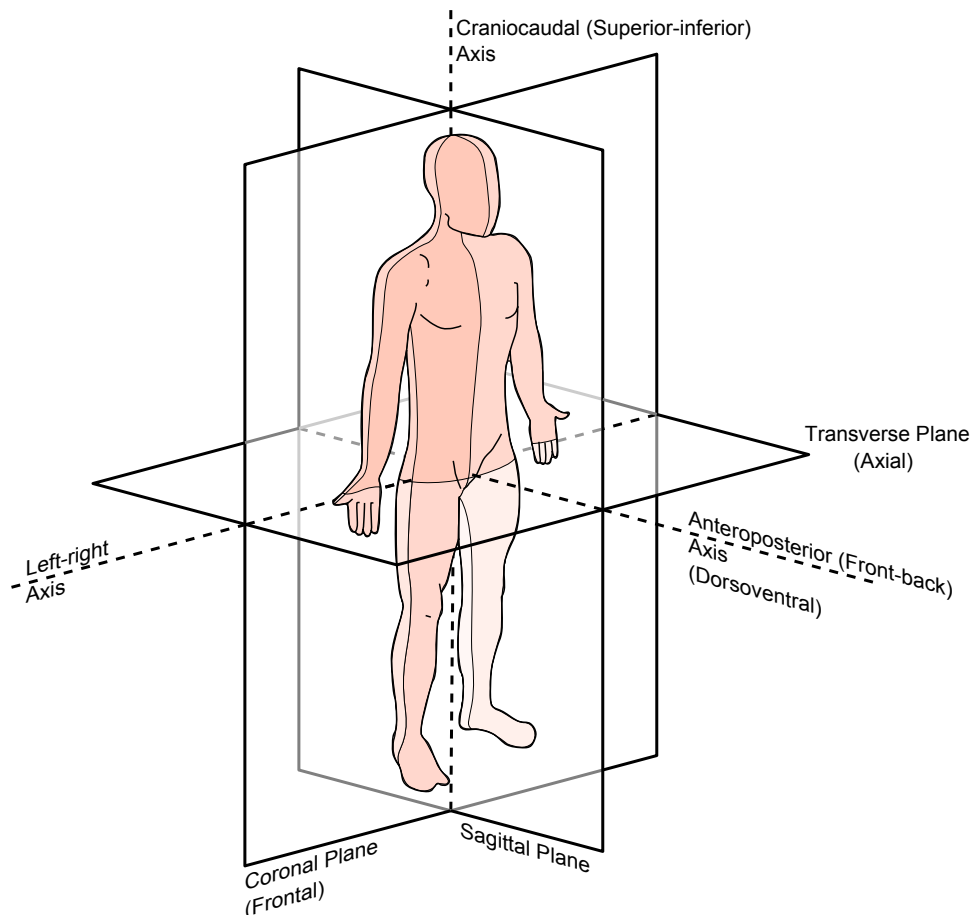
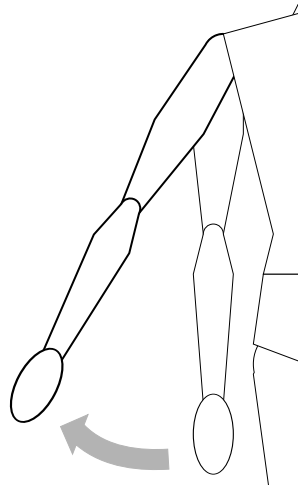
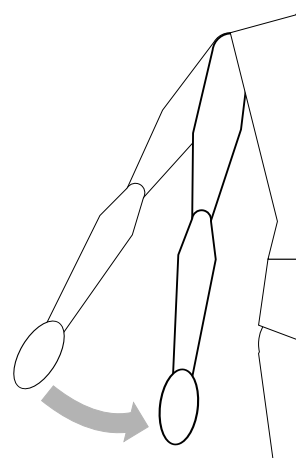


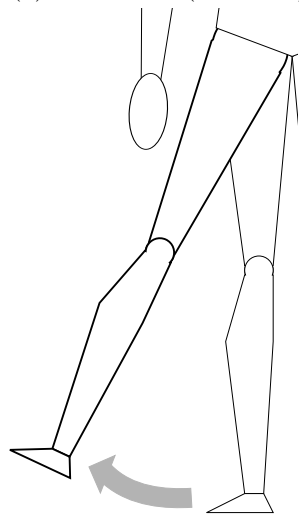
Figure 2-1: The anatomical position, with three reference planes and six fundamental directions. Source: Gait Analysis: An Introduction, by Michael Whittle, © 2002.



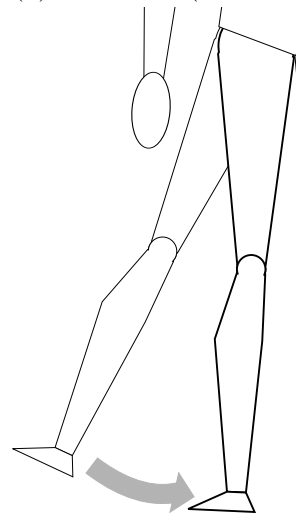
(a) Abduction (Shoulder).



(b) Adduction (Shoulder).



(c) Abduction (Hip).



(d) Adduction (Hip).

Figure 2-2: Movements at the Frontal Plane. Figures based on material from Basic Biomechanics, (6th edition), by Susan J. Hall, ©2011 McGraw-Hill Publishing.

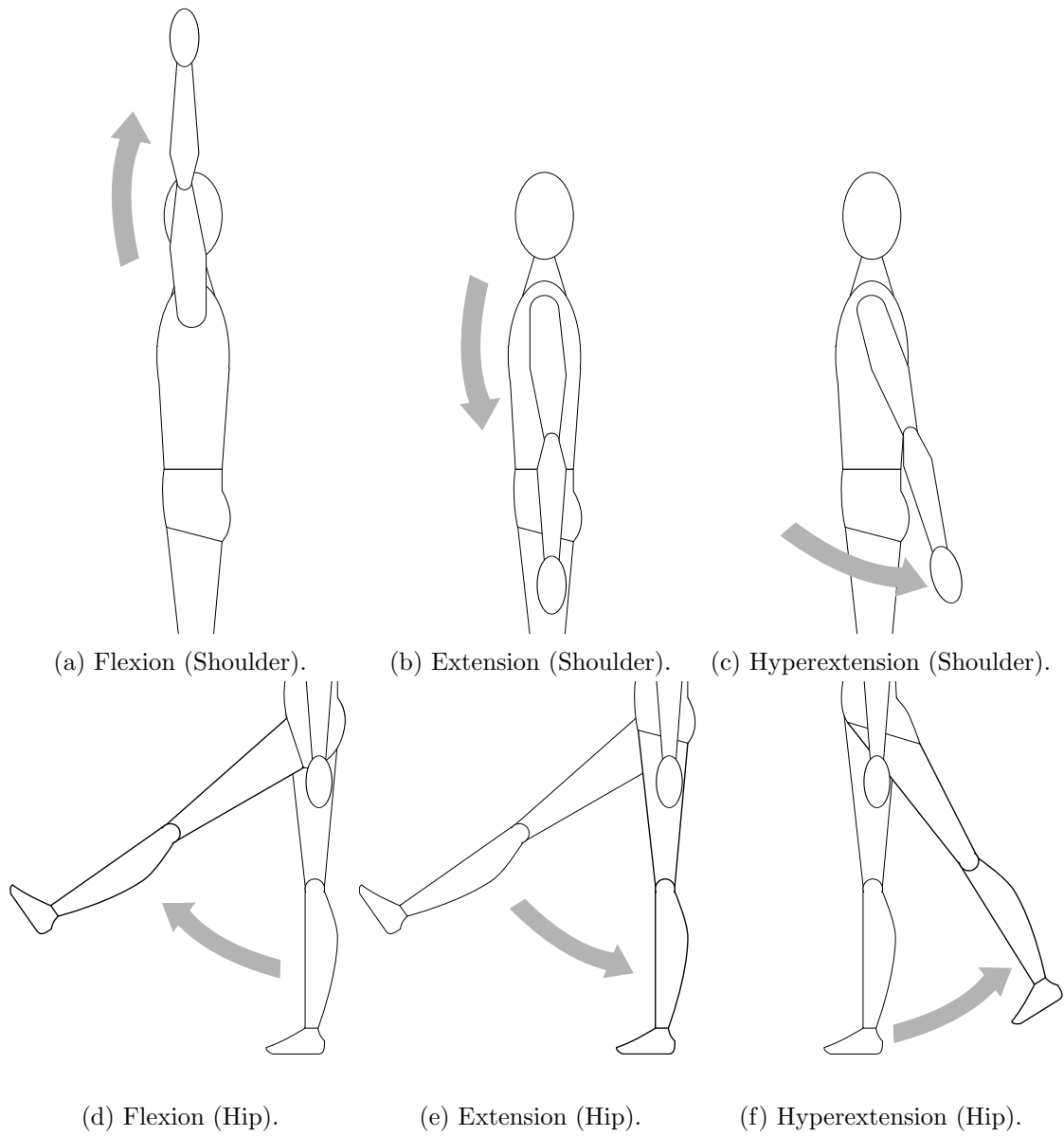


Figure 2-3: Movements at the Saggital Plane.

Figures based on material from Basic Biomechanics, (6th edition), by Susan J. Hall, ©2011 McGraw-Hill Publishing.

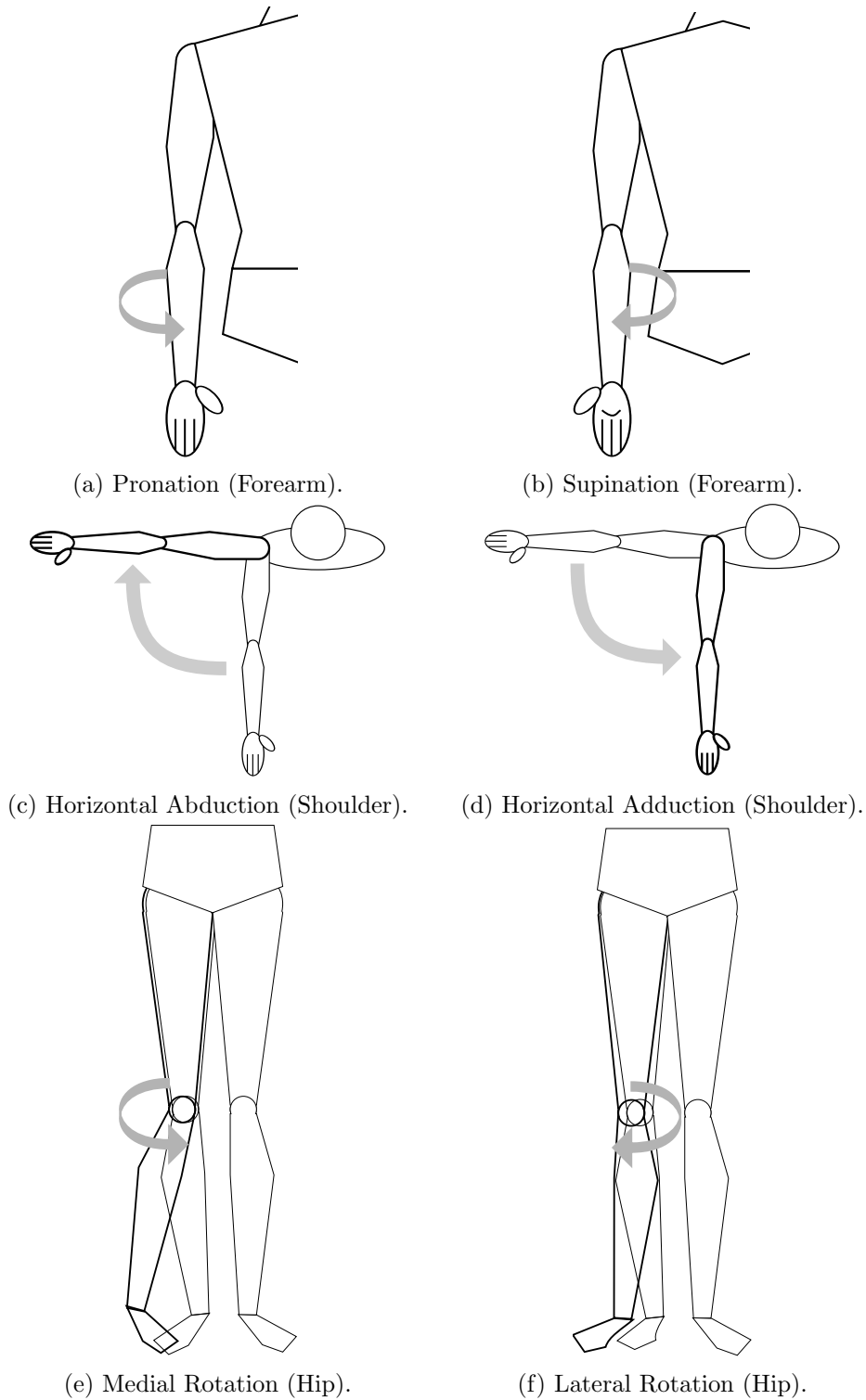


Figure 2-4: Movements at the Transverse Plane.
 Figures based on material from Basic Biomechanics, (6th edition), by Susan J. Hall, ©2011 McGraw-Hill Publishing.

2.2 Human Activity Recognition

The taxonomy of a human activity depends on its complexity [Aggarwal and Ryoo, 2011]. A *gesture* is an elementary movement or presentation of a body part. Some examples of gestures are either “waving an arm” or “flexing a leg”. Gestures are the building blocks for giving a meaningful description to the motion of a person. An *action* is an activity performed by a single person, which is made of several gestures with chronological structure. The actions may involve interactions with objects. Some examples of actions are either “walk” or “drink coffee”. An *interaction* is a human activity involving two or more persons and/or objects. For example, “two persons dance waltz” is an interaction between two persons, or “one person delivers a briefcase to other person” is an interaction between two persons and an object. A *group activity* is an activity performed by conceptual groups, composed of multiple persons and/or objects.

Some applications of the activity recognition are [Aggarwal and Ryoo, 2011]: Domestic Robotics, where can be used for interactions between a robot and a person; in areas of Surveillance, activity recognition is used for detecting suspicious activity, or analysing the activities performed in a room; the Gaming applications aim to achieve interaction with a video game without physical input devices; the Health Care area is a subset of the Surveillance applications, where the activity recognition can be coupled to systems for emergency response.

A particular use case for activity labelling on Domestic Robotics could be: for example, a robot which helps in cooking. A person is preparing food in the kitchen. The vision system of the robot captures motion data of the person. The Activity Recognition System analyses the motion to get the activities performed. The output of the Activity Recognition System provides information to the Action Planning System, which has information of the world and the robot. The Action Planning System picks a plan of action, such as getting closer to the person and ask to help out.

There is a number of challenges on each stage of the activity recognition. When motion data is acquired, we have: noise on the sensor, both from internal and external sources, that alters the captured values of the motion; the occlusion of the sensor, either by other objects or persons, produces inaccurate or incomplete data. There are some issues which are exclusive of the Computer-Vision-based systems: the orientation of the body towards the sensor can

obscure some body parts, generating inaccurate or incomplete data; bad lighting conditions, if they are not compensated, reduce the accuracy of the capture. The challenges on classifying motion data are: the raw motion data can be high-dimensional, so picking the features which provide the best description is necessary; the position of the person in the motion data is not absolute, that is solved by making the motion data relative to a reference frame. Another challenge is that activities can involve interaction with other persons or objects, this is solved by segmenting the data into separate entities and tracking them; and, another challenge is that several activities can have the similar motion data, which is solved by segmenting the motion data before training the recognition model.

2.2.1 Approaches to Activity Recognition

There are two approaches for activity recognition, according to how the motion data is represented and recognised [Aggarwal and Ryoo, 2011]. The *single-layered approach* represents and recognises human activities directly from sequences of images. This approach is suitable for gesture recognition and actions with sequential characteristics. In contrast, the *hierarchical approach* represents high-level human activities with a description in terms of simpler activities. This approach is suitable for analysing complex activities, such as interactions, and group activities.

The taxonomy of the single-layered approach has two ways of modelling human activities: space-time approach and sequential approach [Aggarwal and Ryoo, 2011].

The *space-time approach* views an input video as a 3-D volume XYT . This approach can be categorized further depending on the features used for the XYT volume: volumes of images [Bobick and Davis, 2001; Ke *et al.*, 2007; Shechtman and Irani, 2005], volumes of trajectories [Campbell and Bobick, 1995; Rao and Shah, 2001; Sheikh *et al.*, 2005], or volumes of local interest point descriptors [Ryoo and Aggarwal, 2009; Yilma and Shah, 2005; Wong *et al.*, 2007]

The *sequential approach* uses sequences of features from a human motion source. An activity has occurred if a sequence of features matches any of the sequences of features which had been trained from a set of activities. There are two classifications for sequential approaches: exemplary-based and state model-based [Aggarwal and Ryoo, 2011]. **This work uses the state model-based approach to human activity recognition.**

In the *exemplary-based approach*, human activities are defined as sequences of features which have been trained directly. A human activity is recognised by computing the similarity of a new sequence of features against a set of reference sequences of features; if similarity is high enough, the system deduces that the new sequence belongs to a certain activity. Human beings do not perform the same activity at the same rate or style, and the similarity measuring algorithm must account for those details. An approach to account for those changes is *Dynamic Time Warping* [Aggarwal and Ryoo, 2011; Vintsyuk, 1968], a dynamic programming algorithm which stretches a pattern of motion over the time, to align and match it against a reference pattern of motion. The algorithm returns the similarity between two patterns of motion. When comparing a pattern of motion against a set of reference patterns of motion, the reference pattern which has the highest similarity indicates the most likely activity [Darrell and Pentland, 1993; Gavrilala and Davis, 1996; Yacoob and Black, 1998]. In the *state model-based approach*, human activities are defined as statistical models with a set of states which generate corresponding sequences of feature vectors with a certain probability. This approach accounts for rate and style changes. One of the most used mathematical models for recognising activities is the Hidden Markov Model.

2.2.2 Hidden Markov Models

Hidden Markov Models are stochastic processes where the signal or process to model is assumed to be a Markov process with unobserved states [Rabiner, 1989].

A stochastic process is a collection of random variables which represent the evolution of a random values over time, such as the spectra of a sound signal, or the probability of drawing a ball of a certain color from a set of bowls, which have colored balls in varying amounts [Rabiner, 1989].

The states in a stochastic process have different distribution probabilities for the collection of random variables, and the transitions from a state to other depend on probabilities (non-determinism).

The Markov property indicates that the probability distribution of future states depends upon the present state; in other words, it does not keep record of either past time or future time (memoryless).

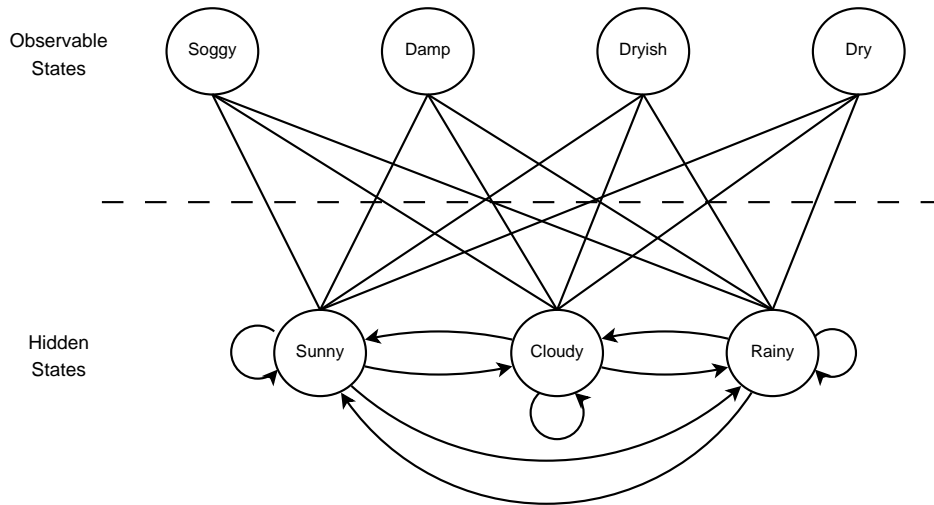


Figure 2-5: Hidden Markov Model.

The unobserved states in a Hidden Markov Model indicate that the states are not visible directly, but output is visible, depending on the state (Figure 2-5).

The most common application of a Hidden Markov Model is temporal pattern recognition, such as speech recognition, handwriting recognition, gesture recognition, speech tagging, following of musical scores, and DNA sequencing.

The output values for the random variables in a Hidden Markov Model can be discrete, originated from a categorical distribution, or continuous, originated from a Gaussian Distribution.

The elements of a Hidden Markov Model (λ) are: $\lambda = \{N, M, A, B, \pi\}$, where N , is the amount of states for the Markov process; M , is the amount of discrete output symbols for the Markov process; A , is the transition probability matrix between states of the Markov process; B , is the emission probability for output symbols per state of the Markov process; and π , is the probability of starting at a certain state of the Markov process.

For a Hidden Markov Model to be useful in real world applications, three basic problems must be solved:

- Evaluation Problem: Given a sequence of observations $O = O_1 O_2 \dots O_T$ and a model $\lambda = (A, B, \pi)$, how to efficiently compute the probability of the sequence of observations, given the model $P(O|\lambda)$?

- **Optimal State Sequence Problem:** Given a sequence of observations $O = O_1 O_2 \cdots O_T$ and a model $\lambda = (A, B, \pi)$, how to choose the most likely sequence of states $Q = Q_1 Q_2 \cdots Q_T$ which explains best the sequence of observations?
- **Training Problem:** How to adjust the parameters of the model $\lambda = (A, B, \pi)$ to maximise $P(O|\lambda)$, the probability of a sequence of observations, $O = O_1 O_2 \cdots O_T$, given the model?

Solution to the Evaluation Problem

The Forward Procedure solves the Evaluation Problem. The forward variable $\alpha_t(i)$ defined as

$$\alpha_t(i) = P(O_1 O_2 \cdots O_t, q_t = S_i | \lambda) \quad (2-1)$$

indicates the probability of the partial observation sequence, $O_1 O_2 \cdots O_t$, (until time t), and state S_i at time t , given the model λ .

The inductive solution of $\alpha_t(i)$ is the following:

1. Initialization:

$$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N \quad (2-2)$$

2. Induction:

$$\alpha_{t+1}(j) = [\sum_i \alpha_t(i) a_{ij}] b_j(O_{t+1}), \quad 1 \leq t \leq T - 1, 1 \leq j \leq N \quad (2-3)$$

3. Termination:

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i) \quad (2-4)$$

The initialization step sets the forward probabilities as the joint probability of state S_i and initial observation O_1 . The induction step computes the partial probability at the state S_j , at time $t + 1$ with the accompanying partial observations. And, the termination step computes the final forward probability by summing all the terminal forward variables $\alpha_T(i)$.

Solution to the Most Likely Sequence of States Problem

The evaluation problem is solved by the Viterbi Algorithm, which computes the most likely sequence of connected states $Q = Q_1 Q_2 \cdots Q_T$ which generates a sequence of observations $O_1 O_2 \cdots O_T$, given a model λ .

The Viterbi Algorithm uses the variable δ , which contains the highest probability of a single path, at the time t .

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P[q_1 q_2 \cdots q_t = i, O_1 O_2 \cdots O_t | \lambda] \quad (2-5)$$

The highest probability along a single path, at time $t + 1$, is computed as:

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] \cdot b_j(O_{t+1}). \quad (2-6)$$

The most likely path is the sequence of these maximised variables, for each time t and each state j . The array $\psi_t(j)$ tracks all the maximised variables $\delta_t(j)$. The most likely state sequence is retrieved by backtracking the variable $\psi_t(j)$.

1. Initialization:

$$\delta_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N \quad (2-7a)$$

$$\psi_1(i) = 0. \quad (2-7b)$$

2. Recursion:

$$\delta_t(j) = \max_i [\delta_{t-1}(i) a_{ij}] b_j(O_t), \quad 2 \leq t \leq T$$

$$1 \leq j \leq N, \quad (2-8a)$$

$$\psi_t(j) = \arg \max_i [\delta_{t-1}(i) a_{ij}], \quad 2 \leq t \leq T$$

$$1 \leq j \leq N. \quad (2-8b)$$

3. Termination:

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)] \quad (2-9a)$$

$$q_t^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)]. \quad (2-9b)$$

4. Backtracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T - 1, T - 2 \dots 1. \quad (2-10)$$

Solution to the Training Problem

An approach for solving the Training Problem is the Viterbi Learning algorithm [Rabiner and Juang, 1993], which uses the Viterbi Algorithm to estimate the parameters of a Hidden Markov Model. The algorithm can estimate the parameters from a set of sequences of observations. That property makes it different of the Baum-Welch algorithm [Rabiner, 1989], which requires all the training observations in a single sequence.

The initialization of the transition matrix is done with random values. A bitmask matrix describing the transitions of a specific graph topology can be used to set the probabilities. The transition probabilities under a bitmask value equal to zero get a very small value, while the transition probabilities under a bit mask value equal to one get a random value. The random values on each row are normalised, so their sum is equal to one.

The initialization step for the emission matrix uses one of these approaches: random values or segmented observation sequences. When initializing with random values, all the values must be larger than zero and each row must be normalised, so their sum is equal to one. In the segmented observations sequences approach, the sequence is split by the number of states of the Hidden Markov Model. If the length of the sequence is not a multiple of the number of states, some of them get less observations. For each state, the emission probability of each symbol is equal to the count of that symbol divided by the total amount of symbols assigned to that state.

The initial probability vector can be initialized either to uniform probabilities or by assigning

the larger probability to an state or a number of states. The probabilities are normalised so their sum is equal to one.

In the induction step, for each sequence of observations for training, the Most Likely State Path is computed with the Viterbi Algorithm on the initial Hidden Markov Model, and the Likelihood Probability is computed either with the Viterbi Algorithm or the Forward Algorithm on the initial Hidden Markov Model. The Most Likely State Path of each sequence is stored for computing the parameters of an updated Hidden Markov Model. The Forward Probability of each sequence is accumulated in the variable $prob_{old}$ for computing the condition of termination.

The values of the updated transition matrix A are computed by counting the transitions from the state Q_t , to the next state Q_{t+1} , on the Most Likely State Paths associated to each sequence of observations for training. At the end, the values of each row on the transition matrix are normalised, so their sum is equal to one.

The values of the updated emission matrix B are the frequencies of each symbol in the observation sequence, O_t , per state in the Most Likely State Path, Q_t , at the time t , i.e, $B(Q_t)(O_t) = B(Q_t)(O_t) + 1$. The values of each row on the emission matrix are normalised, so their sum is equal to one.

The initial probability vector π is updated by counting the states which appear as the first element of each Most Likely State Path Q_1 .

A new Hidden Markov Model is built from the updated model parameters $\lambda = A, B, \pi$. To check if the model maximises $P(O|\lambda)$, the Forward Probability of each sequence of observations for training is computed with the model, and the probability is accumulated in the variable $prob_{new}$.

The conditions for terminating the algorithm are: either the absolute of the difference of $prob_{new}$ and $prob_{old}$ is smaller than a threshold, or a certain number of iterations has been reached. If any of those conditions is false, the updated Hidden Markov Model passes to the next iteration of the induction step, otherwise, the algorithm returns the updated Hidden Markov Model.

Logarithmic Scaling

Both Forward Probability Algorithm and Viterbi Algorithm store the result of floating-point operations in a single variable. The accumulated product of fractional values is a value so small that might fall below the minimum precision of the floating-point variable which stores the result. That value can be represented in logarithmic scale, where multiplication and division operations are represented as addition and subtraction respectively. The range of values in logarithmic scale goes from $-\infty \cdots +\infty$, where negative logarithmic values represent fractional values and positive logarithmic values represent integral values larger than or equal to one.

The logarithmic scale in the Forward Algorithm applies at each iteration in the Induction step, where a scale variable accumulates the value of the forward variable α , for each state at the time t . The forward probability for each state at the time t is divided by the scale variable at the time t . At the end of the Forward Algorithm, the Forward Probability is the sum of the logarithms of the scale variable at the time t from $1 \cdots T$, where T is the length of the sequence of observations.

For the Viterbi Algorithm, the values of the transition probability matrix A , the emission probability B , and the initial transition probability vector π are converted to logarithmic values. In the initialization and recursion steps, the value of the variable δ is updated by addition of the logarithmic values of A , B , and π .

2.2.3 Hidden Markov Model Topologies

Depending on the process that generates a signal, the contents of the signal can have either a stationary structure, or a chronological or linear structure. The structure of the contents of the signal indicates which is the most suitable Hidden Markov Model [Fink, 2007].

The classical case of the set of bowls containing different proportions of colored balls is an example of a stationary process: any ball is drawn from any bowl at any time. For this case, the most suitable topology for the Hidden Markov Model is the ergodic model (Figure 2-6a), where all the states are fully connected [Fink, 2007].

In automated motion recognition and activity recognition applications, the input data to be processed has a chronological or linear structure [Fink, 2007].

The simplest topology for linear processes is the linear model (Figure 2-6b), where each

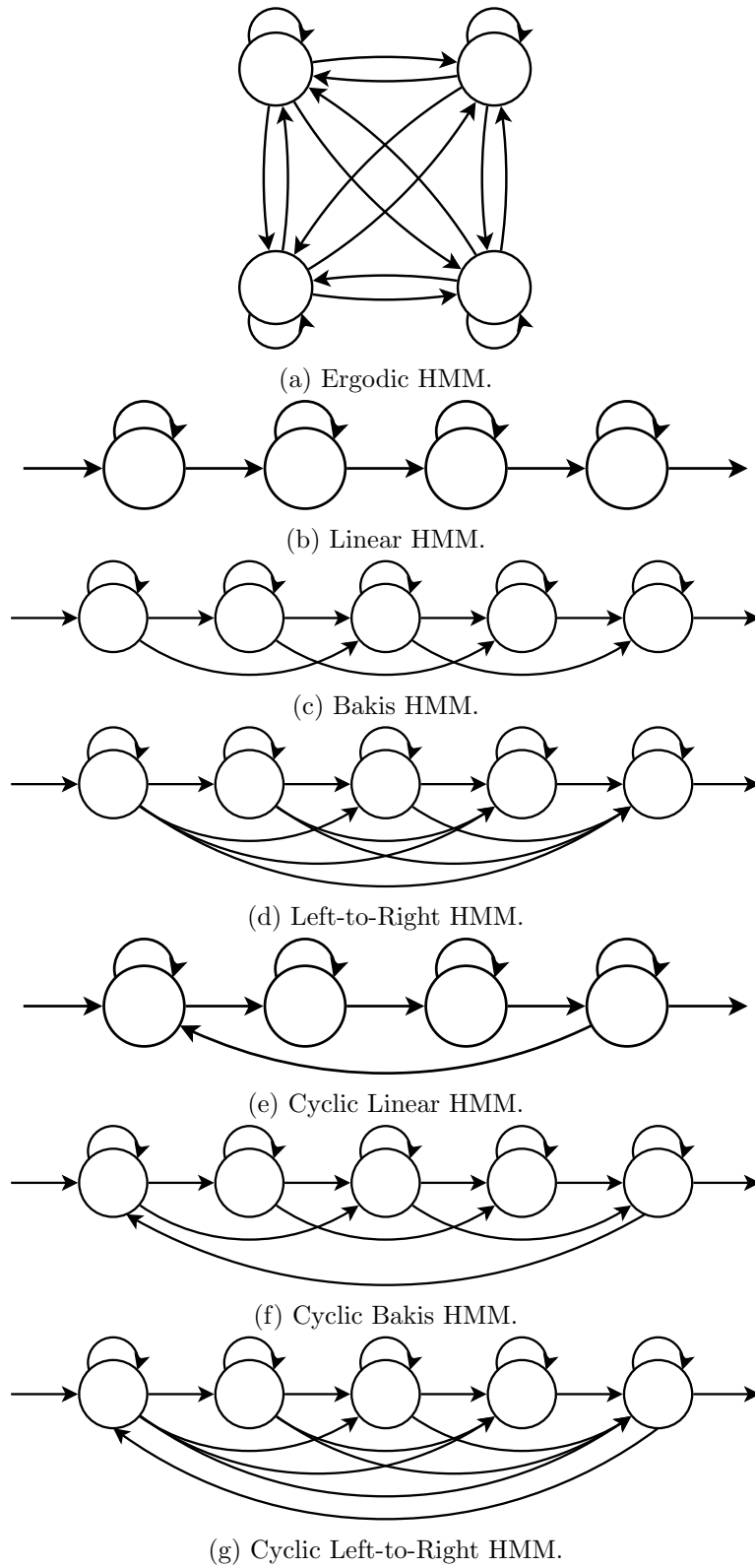


Figure 2-6: Hidden Markov Model Topologies.

state connects to itself (self-transitions) and to the next state. The self-transitions account for variations in the duration of the patterns in a state [Fink, 2007].

The flexibility in the modelling of the duration can increase if it is possible to skip individual states in the sequence. A popular topology variation for automated speech and handwriting recognition is the Bakis model (Figure 2-6c). Besides of the self-transitions and the transitions to the next state, the states in the Bakis model have a transition that skips two states ahead the current state, when the state is not the last state or the next-to-last state [Fink, 2007].

The largest variations in the chronological structure are achieved by allowing a state to have transitions to itself and any posterior states in the chronological sequence. The only forbidden transitions are to past states. This model is called Left-to-Right model (Figure 2-6d) [Fink, 2007].

Any of the Hidden Markov Models for signals with chronological structure — Linear (Figure 2-6e), Bakis (Figure 2-6f), Left-to-Right (Figure 2-6g)— can model cyclic signals by adding a transition from the last state to the first state [Magee and Boyle, 2002].

2.2.4 Activity Recognition with Hidden Markov Models

The *Hidden Markov Model* is one of the most commonly used statistical models for Sequential Activity Recognition. There are two approaches for recognising activities with Hidden Markov Models: Maximum Likelihood Probability (MLP) [Chen *et al.*, 2006; Starner and Pentland, 1995; Sung *et al.*, 2012; Xia *et al.*, 2012; Yamato *et al.*, 1992] and Most Likely State Path (MLSP) [Bobick *et al.*, 1998; Nergui *et al.*, 2012; Oh *et al.*, 2010; Yu and Aggarwal, 2006; Zhang *et al.*, 2006].

Maximum Likelihood Probability Activity Recognition

- Features:
 - Each Activity has a Hidden Markov Model.
 - Each Hidden Markov Model computes the Forward Probability of a sequence of observation symbols.

- The Hidden Markov Model with the largest Forward Probability identifies the activity.
- Advantages:
 - New activities can be added easily by training another Hidden Markov Model.
 - The evaluation of a sequence of observation symbols can be performed with multiple tasks.
- Disadvantages:
 - Motion segmentation is required when recognising connected activities.

Most Likely State Path Activity Recognition

- Features:
 - All the activities are embedded in a single large Hidden Markov Model.
 - Each activity is represented by a subset of states.
 - A sequence of observation symbols is processed to obtain the sequence of most likely states which generates it.
- Advantages:
 - The evaluation of connected activities is possible without motion segmentation.
 - Reconstruction of activities from the sequence of most likely states.
- Disadvantages:
 - Adding a new activity is complicated: the Hidden Markov Model for the new activity is trained separately, the Hidden Markov Model is merged with the single large Hidden Markov Model and the single large Hidden Markov Model must be retrained to update the probabilities of emission and transition.
 - The computation of likelihood probability with Viterbi Algorithm is slower than with Forward Algorithm,

- Reconstruction of activities requires an index which associates each activity with a subset of states.

Variants of Hidden Markov Models

A limitation of the Hidden Markov Models is that do not allow for complex activities, interactions between persons and objects, and group interactions. To enhance the probability of recognising activities with Hidden Markov Models, variations to the model have been studied in previous works.

In the Conditioned Hidden Markov Model [Glodek *et al.*, 2012a,b], the selection of states is influenced by an external cause, such as the symbols generated by an external classifier. The probability of those symbols increases the probability of a sequence. This model allows using two streams of different features from the same data.

The Coupled Hidden Markov Model [Brand *et al.*, 1997; Oliver *et al.*, 2000; Starner and Pentland, 1995] is formed by a collection of Hidden Markov Models. Each Hidden Markov Model handles a data stream. The observations cannot be merged because the cartesian product of the amount of the symbols of each data stream would result in a massive amount of symbols for a single Hidden Markov Model. The nodes at the time t are conditioned by the nodes at the time $t - 1$ of all the related Hidden Markov Models. This model is suitable for recognising activities using data from multiple sources.

The states of a Hidden Semi-Markov Model [Duong *et al.*, 2005; Natarajan and Nevatia, 2007; Shi *et al.*, 2008] emit a sequence of observations. The next state is predicted from how long has remained in the past state. This model relaxes the memoryless property of a Markov Chain.

The Maximum Entropy Markov Model [Sung *et al.*, 2011, 2012] models explicitly the dependence between each state and the full observation system. The model completely ignores modelling the probability of the state $P(X)$. The learning objective function is consistent with the predictive function $P(Y|X)$. The observation Y sees all the states X , instead of the observation being dependent on the state.

The Compound Hidden Markov Model [Fink, 2007; Guenterberg *et al.*, 2009; Lowerre, 1976; Ryo and Aggarwal, 2006; Savage, 1995] is formed by the concatenation of Hidden Markov

Models representing sub-word units, which form a lexicon of words. Parallel connections link all the individual sub-word units. The recognised words are subsets of connected states in the Most Likely State Path. The representation of the model can be simplified by the addition of non-emitting states.

The Dynamic Multiple Link Hidden Markov Model [Gong and Xiang, 2003] is built by connecting multiple Hidden Markov Models. Each Hidden Markov Model models the activities of a single entity. The relevant states between multiple Hidden Markov Models are linked. This model is suitable for group activities.

The Two-Stage Linear Hidden Markov Model [Nguyen-Duc-Thanh *et al.*, 2012] is formed by two stages of Linear Hidden Markov Models. The first stage recognises low-level motions or gestures to generate a sequence of gestures. The sequence of gestures becomes the input for the Hidden Markov Model at the second stage, which recognises complex activities from the sequences of gestures.

The Layered Hidden Markov Model [Oliver *et al.*, 2002; Zhang *et al.*, 2006] is a model which organizes Hidden Markov Models in layers of increasing activity levels. The layers at the lowest level recognise simple activities. The layers at the upper levels use the simple activities to recognise complex activities. The Layered Hidden Markov Model has higher accuracy than Hidden Markov Models, it is robust to environment changes, and has higher discriminative power.

Chapter 3

Proposed Approach

If you cannot get rid of the family
skeleton, you may as well make it dance.

George Bernard Shaw, *Immaturity*

The method for activity recognition proposed in this work uses a Compound Hidden Markov Model for activity labelling. Two types of features were evaluated for classifying skeletal data: one based on distances between joints, and other based on invariant Chain Codes.

3.1 Skeleton Features

In order to simplify the recognition of data from motion capture systems, the data must be processed to get a representation with the most relevant information.

3.1.1 Interjoint Distance

The features of the skeleton are a variation of those presented in Glodek et al [Glodek *et al.*, 2012b]. The feature descriptor is a set of Euclidean Distances between joints. The skeleton features in Glodek et al. (Figure 3-1a) represent the upper body with a feature vector of 8 elements. The features presented in this work use a set of Euclidean Distance between joints of the upper body and joints of the lower body (Figure 3-1b), resulting in a feature vector of 16 elements.

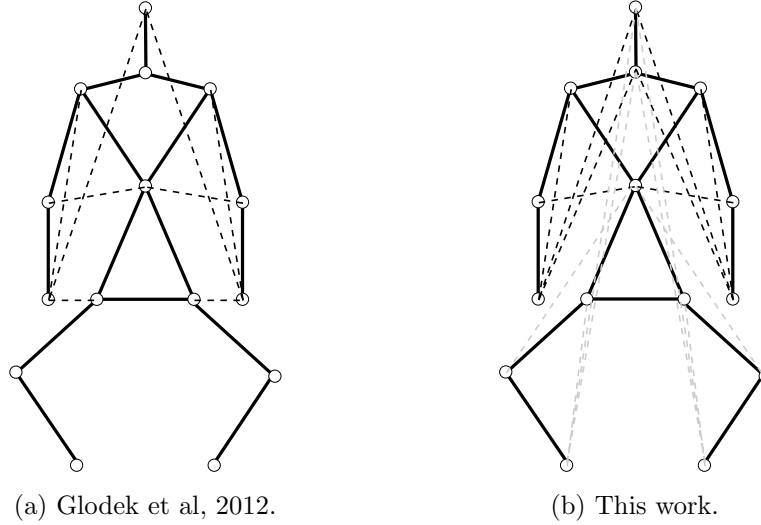


Figure 3-1: Skeleton Features.

The noise of the KinectTM sensor and the arithmetical precision on the algorithm which computes the skeleton from depth data [Shotton *et al.*, 2011] provoke variations on the location of each body part on the skeleton, thus affecting the distance between connected body parts. To correct those variations, the original skeleton is converted to a skeleton made of unit vectors.

3.1.2 Interjoint Distance Feature Classification

The observations for the Hidden Markov Model come from the classification of unknown data against the codebook of key frames. The dissimilarity between two vectors of Interjoint Distance Features is measured by computing the Euclidean Distance.

If a vector of features is very similar to another vector of features, the dissimilarity will be closer to zero. Large values of dissimilarity mean two skeletons, represented as vectors of features, do not have similar shape.

$$D(A, B) = \sqrt{\sum_{i=1}^n (A_i - B_i)^2}, \quad n = 16 \quad (3-1)$$

3.2 Compound Hidden Markov Model

The model proposed for activity labelling is a Compound Hidden Markov Model [Gunterberg *et al.*, 2009; Lowerre, 1976; Ryoo and Aggarwal, 2006; Savage, 1995], which is a Hidden Markov Model where a subset of states represent a pattern and connects to a common initial state and a common final state, and the common final state always connects to the common initial state. The recognised patterns are extracted from the sequence of most likely states, obtained from applying the Viterbi Algorithm to a sequence of observations.

The Compound Hidden Markov Model is formed by several simpler Hidden Markov Models, whose topologies are configured according to the type of activity to model: the stationary activities, like *sit still* and *stand still*, have a single state; the non-periodic activities, like *stand up* and *sit down*, are modelled with Linear Hidden Markov Models; and, the periodic activities, such as *walk*, are modelled by a Cyclical Linear Hidden Markov Model.

The activities are connected using context information. For example, the *sit still* activity connects to the first state of the *stand up* activity, and receives a connection from the last state of the *sit down* activity. The *stand still* activity connects to the first state of the *sit down* activity, and receives a connection from the last state of the *stand up* activity. Also, the *stand still* activity connects to the first state of the *walk* activity and receives a connection from the last state of the *walk* activity (Figure 3-2).

The stationary activities (*sit still*, *stand still*) are modelled with a Hidden Markov Model formed by a single state. The emission probabilities of each Hidden Markov Model are initialized to the averaged frequency of the observations for the corresponding idle activity.

The non-periodic activities (*stand up* , *sit down*) and the periodic activities, (*walk*), are trained using the following procedure: the observations from motion data of each activity are segmented into three sections: the *anticipation* (Figure 3-3a), which contains the poses which indicate that a motion is about to start; the *action* (Figure 3-3b), which contains the poses which describe a motion; and the *reaction* (Figure 3-3c), which contains the poses which indicate the recovery from an action to a neutral position. These three sections, anticipation-action-reaction (AAR), come from the theory of animation [Lasseter, 1987; Williams, 2009].

The Hidden Markov Models for stationary activities, non-periodic activities, and periodic activities are merged in a Compound Hidden Markov Model, as specified in the Section 3.2,

and its parameters are re-estimated using Viterbi Learning with all the elements of the training set.

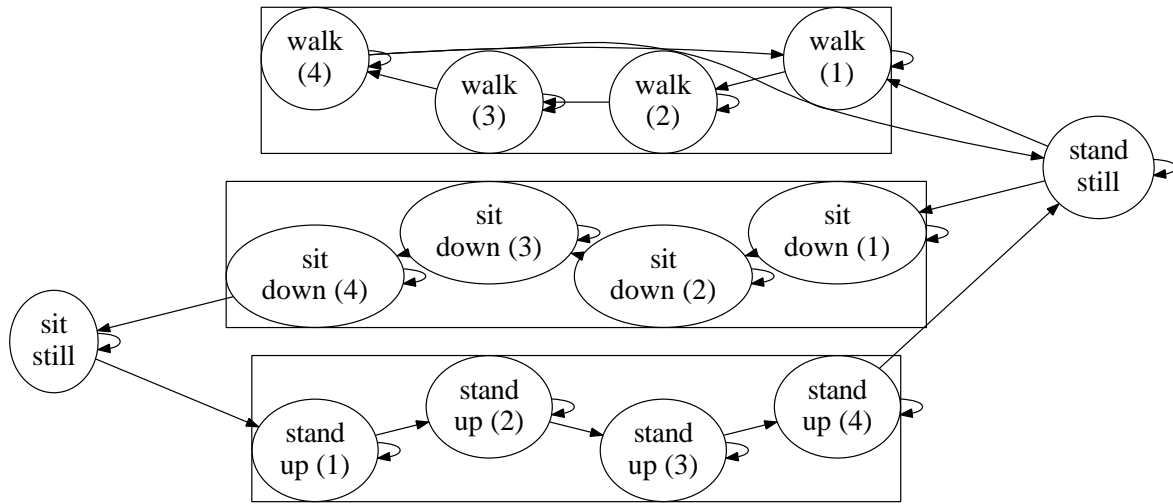


Figure 3-2: Compound Hidden Markov Model for activity labelling.

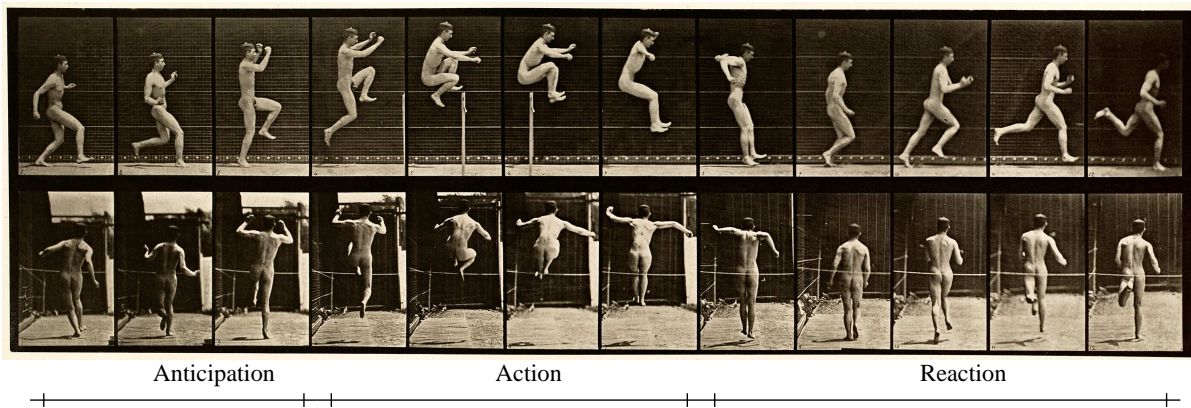


Figure 3-3: An activity has three sections: the motion preceding the activity (Anticipation), the motion of the activity (Action) and the motion after the activity is performed (Reaction).
 Source: Animal Locomotion, Vol. 1, Plate 154, by Eadweard Muybridge, ©1887.

Chapter 4

Implementation

Experience without theory is blind, but
theory without experience is mere
intellectual play.

Immanuel Kant

In order to create software from the algorithms described in the former sections, each algorithm has to be implemented in a high-level programming language to generate a software component which can be connected to other components to build a whole software system.

4.1 Organization of the Behaviour Recognition System

The Behaviour Recognition System is organized according to a workflow (Figure 4-1), where each stage has one or more software components which accomplish a certain task with incoming data and whose output is used as input for the next stage.

On a real world scene, an activity being performed by a person is recorded using the KinectTM sensor, whose data is processed to obtain RGB images, depth images, and the coordinates of the joints of the skeleton (*Recording stage*). The joints of the skeleton are normalised before computing the skeleton features. In order to recognise an activity, the features which are computed from samples of a recording are labelled through classification against the codebook of key frames. Those labels are stored in a sequence of observations, which is used as input for a Compound Hidden Markov Model to compute a Viterbi path (*Activity Recognition stage*).

The Viterbi path is labelled using a list which contains the states which are associated to a certain activity, to obtain the sequence of activities which are being performed by the person (*Activity Labelling stage*).

4.2 Software Used for Development

4.2.1 C++ programming language

The programming language of choice is C++, because it has a large amount of software libraries which are used in scientific and academical environments, the programs written in that language can be deployed both in Microsoft® Windows and GNU/Linux; and, last but not least, it is the programming language which we are more proficient in software development.

While the C programming language has an even larger amount of libraries and it is portable across operating systems, the reasons for not choosing it are: there is no support for Object Oriented Programming, which is helpful for reusing and extending software components; the memory and resource management are explicit, C++ has this issue too but it has operators which simplify those tasks; and some libraries used in this work exist only in C++.

In order to avoid the “Not Invented Here Syndrome” as much as possible, the implementation of the algorithms is done around existing libraries of software, giving preference to those which are available for Microsoft® Windows and GNU/Linux operating systems.

4.2.2 Eigen library

Eigen [Guennebaud *et al.*, 2010] is a high-level C++ library of template headers that implements linear algebra and matrix operations. This library simplifies the implementation of algorithms which operate on vectors and matrices by providing functions and operators which operate across their elements. As it is a library of template headers, there is no binary library to link against.

4.2.3 Boost C++ libraries

Boost [Dawes *et al.*, 2001] is a set of peer-reviewed portable C++ source libraries which provide support for tasks and structures such as linear algebra, pseudo random number generator, multi-

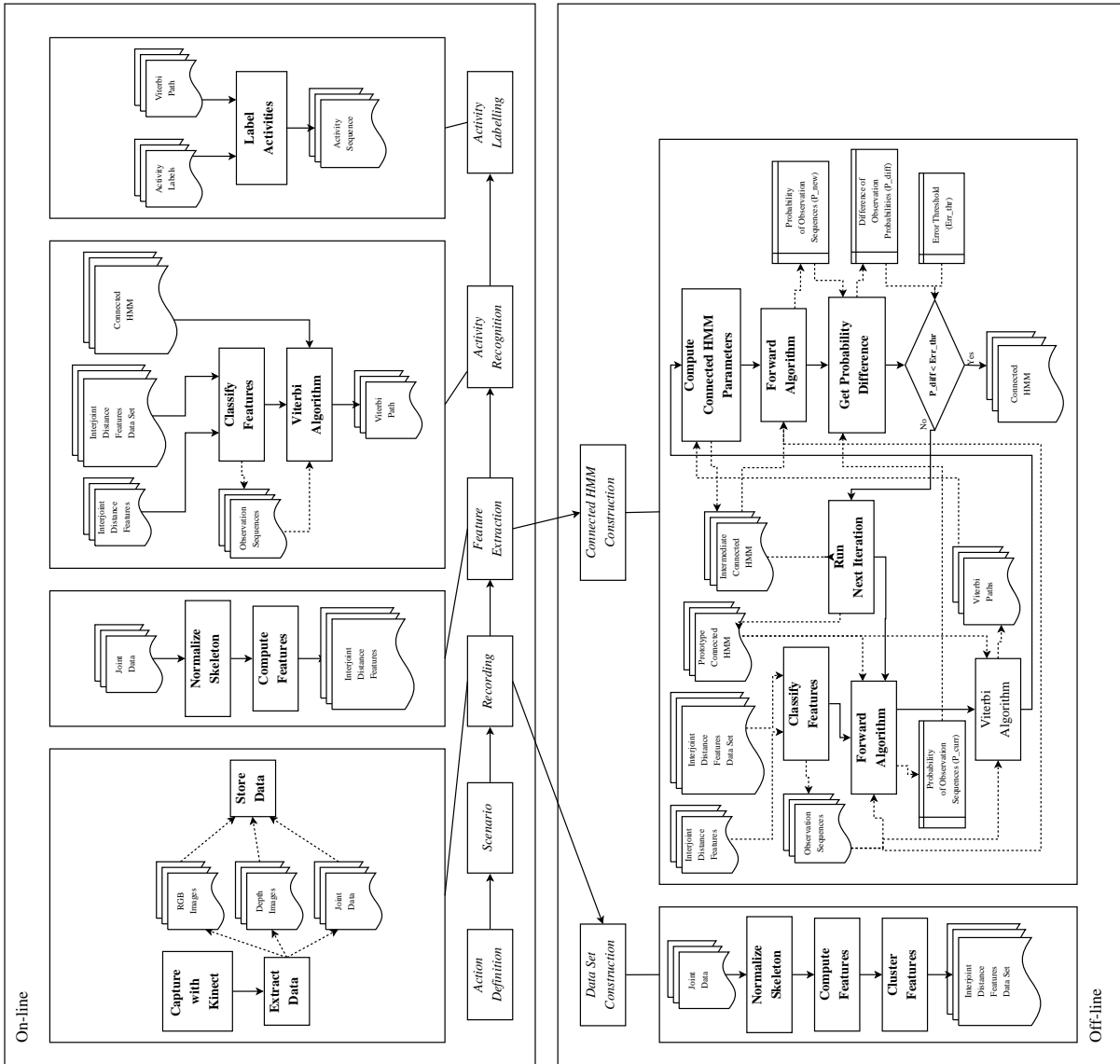


Figure 4-1: Overall Workflow of the Behaviour Recognition System.

threading, image processing, regular expressions and unit testing. The functions contained in these libraries work in almost any modern operating system, thus simplifying the cross-platform development.

4.2.4 UMDHMM library

UMDHMM [Kanungo, 1999] is a library written in C programming language for working with discrete Hidden Markov Models, and implements the Forward-Backward, Viterbi and Baum Welch algorithms.

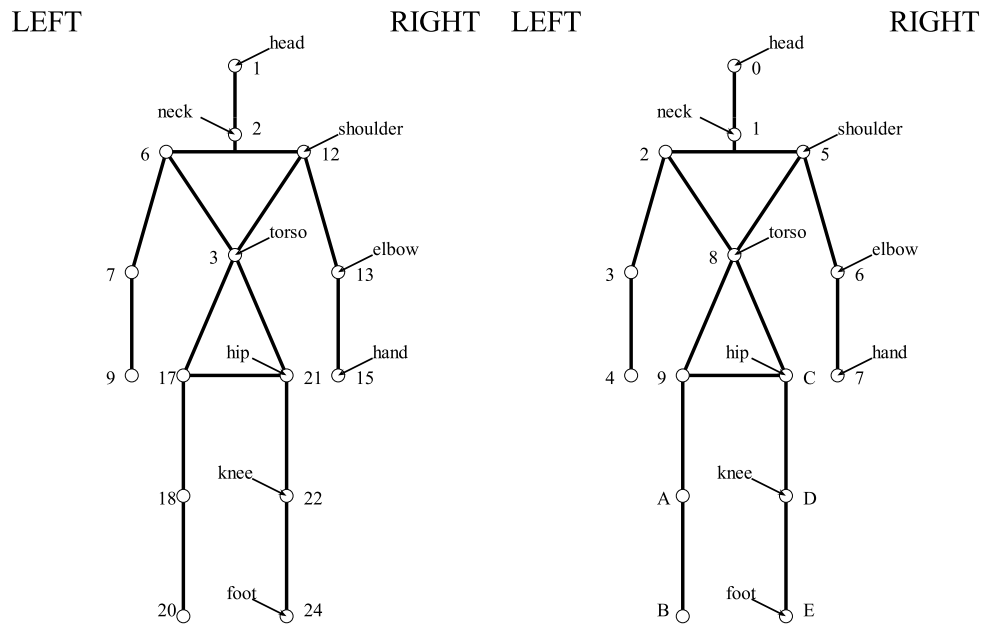
4.2.5 OpenNI® library

OpenNI® [OpenNI, 2009] is a C library, with C++ binding code, which provides functions for the development of natural user interfaces and organic user interfaces for natural interaction devices, such as the Kinect™ sensor.

OpenNI® Skeleton Structure

The skeleton computed by motion capture systems, either by active sensors or passive sensors, is useful for analysing motion, since each limb has 3 degrees of freedom, which accounts for a large number of combinations of motions and angles which can be used to extract relevant information about how a limb is moving.

The skeleton computed by the OpenNI® library has the following joint enumerations (Figure 4-2a), some labels are missing because they belong to joints which are not computed by the current version of the library (1.5.4.0) and they have all their axis values set to zero. For purposes of this research, the joints are numbered in this way (Figure 4-2b): 0) Head (0); 1) Neck (1); 2) Left Shoulder (2); 3) Left Elbow (3); 4) Left Hand (4); 5) Right Shoulder (5); 6) Right Elbow (6); 7) Right Hand (7); 8) Torso (8) 9) Left Hip (9); 10) Left Knee (A); 11) Left Foot (B); 12) Right Hip (C); 13) Right Knee (D); 14) Right Hip (E).



(a) OpenNI® Skeleton Joints Structure. (b) Structure used in this work.

Figure 4-2: Three-dimensional Joint Data Hierarchical Structure.

Chapter 5

Tests and Results

Science isn't about WHY, it's about
WHY NOT!

Cave Johnson

PORTAL 2

5.1 Data Source

In order to assess the labelling accuracy of both the Compound Hidden Markov Model and some reference Hidden Markov Models, they were tested with a data set of human activities.

The tests were performed using the Microsoft Research Daily Activity 3-D Data set (MSRDaily) [Wang *et al.*, 2012], which was captured by using a KinectTM device.

The data set is composed by sixteen activities, *a*) drink; *b*) eat; *c*) read book; *d*) call cellphone; *e*) write on a paper; *f*) use laptop; *g*) use vacuum cleaner; *h*) cheer up; *i*) remain still; *j*) toss paper; *k*) play game; *l*) lay down on sofa; *m*) walk; *n*) play guitar; *o*) stand up; and *p*) sit down which are performed by ten persons, who execute each activity two times, once in standing position, and once in sitting position. There is a sofa in the scene where the persons perform the actions. The dataset has three streams of data: depth maps (.bin), skeleton joint positions (.txt), and RGB video (.avi). There are $16 * 10 * 2 = 320$ files for each stream. The whole set is formed by $320 * 3 = 960$ files. The position of the joints of the skeleton are computed from the depth map [Shotton *et al.*, 2011].

For the purpose of this work, only the skeleton joint positions were used as input for labelling the actions, as well as a subset of activities: *a*) remain still (sitting pose) (Fig. 5-1a); *b*) remain still (standing pose) (Fig. 5-1b); *c*) walk (Fig. 5-1c,5-1d); *d*) stand up (Fig. 5-1e,5-1f); and *e*) sit down (Fig. 5-1g,5-1h) . **Those activities are selected because there is a clear start in the sitting pose or the standing pose, or there are transitions between the sitting pose and the standing pose.**

5.2 Training

At the training step, the Hidden Markov Model for each activity are computed from a training set of motion data. The training set is made of the motion data from the first six subjects of the MSRDaily data set, while the motion data of the last four subjects constitute the testing set.

5.2.1 Computing the Codebook

The KinectTM sensor captures the depth map \vec{D} of a motion sequence of an activity performed by a person. The depth map is processed to extract a skeleton $\vec{S} = \{j_1, j_2 \dots j_{15}\}, j = \{x, y, z\}$ [Shotton *et al.*, 2011]. During the capture, a skeleton represents a single frame of the motion, therefore, a whole motion sequence contains several skeletons. The training set of an activity is formed by motion sequences of the same activity performed by several people.

First of all, the skeletons have their features extracted, using the algorithm described in the Section 3.1. All the features from the skeletons of the training set are clustered with the *k*-means algorithm. The centroids of the clusters become the codebook of key frames. The amount of symbols used in this work is 255, because that is the amount of symbols which provided the best labelling accuracy on the testing set, after performing tests on different amounts of symbols for the codebook, which were 31, 63, 127, 255, 511, 1023, 2047, and 4095 centroids. ¹

¹The reason for those sizes for the codebooks was that the initial amount of symbols for all the experiments were powers of two— 32, 64, 128, 256, 512, 1024, 2048, and 4096— but one of the centroids computed by the *k*-means algorithm had undefined values (NaN values) and had to be removed from the codebook, to avoid arithmetical errors when computing the Euclidean distance between any data object and a centroid with undefined values.



(a) Idle, sitting position.



(b) Idle, standing position.



(c) Walk in front of a sofa.



(d) Walk behind a sofa.



(e) Stand Up, frontal orientation.



(f) Stand Up, three-quarters orientation.



(g) Sit Down, frontal orientation.



(h) Sit Down, three-quarters orientation.

Figure 5-1: Subset of activities from Microsoft Research Daily Activity 3-D used in this work.

5.2.2 Computing the Observations

The features are computed for each frame of the skeleton data, using the algorithm described in the Section 3.1. The features are measured against the codebook of key frames for dissimilarity using Euclidean Distance. The element with the smallest dissimilarity becomes the observation of each frame.

5.2.3 Building the Compound Hidden Markov Model

The Hidden Markov Model for a non-stationary activity has the following structure for its states: the amount of states is N , where $N \geq 3$, so the states can contain all the states of a motion; the state S_1 is for the random variables of the anticipation of the motion (Anticipation State), the state S_N is for the random variables of the reaction of the motion (Reaction State), and the states $S_2 \cdots S_{N-1}$ are for the random variables of the action of the motion (Action States).

The transition probabilities from the Anticipation State to the Action States are initialized to uniform values. There are no transitions from the Action States to the Anticipation State. The transition probabilities from the Action States to the Reaction State are initialized to uniform values. And, the transition probabilities from the Reaction State to the Anticipation State are set to uniform values.

The observations from the anticipation section are used for initialize the emission probabilities of the Anticipation State. The observations from the reaction section are used to initialize the emission probabilities of the Reaction State. The emission probabilities of the Action States are initialized to random values.

The states for the stationary activities (Sit Still, Stand Still) are connected to the states of the non-stationary according to the specifications stated in the the Section 3.2.

Both the transition probabilities and the emission probabilities for all the States will be refined after applying Viterbi Learning [Rabiner and Juang, 1993] to the Model.

5.3 Testing Activity Labelling.

The assessment of the quality of a labelled activity is performed on the results of computing the Most Likely State Sequence from the observations of an activity.

The joints of the skeleton \vec{S} are converted to vector of features \vec{c} (Section A). The features \vec{c} are classified against a codebook of key frames $F = \{f_1, f_2 \dots f_k\}$, using Euclidean Distance.

The key frame with the minimum distance becomes an observation o , which is appended to a sequence of observations $\vec{O} = \{o_1, o_2 \dots o_t\}$.

5.3.1 Assessing Labelling Accuracy

The first Hidden Markov Model to test is an Ergodic Hidden Markov Model where each state represents a single activity, giving a total of five states (Figure 5-2a).

The second Hidden Markov Model to test is organized like a Finite State Machine. Each activity is represented by a single state, giving a total of five states. The connections between the states of each activity use a language model.

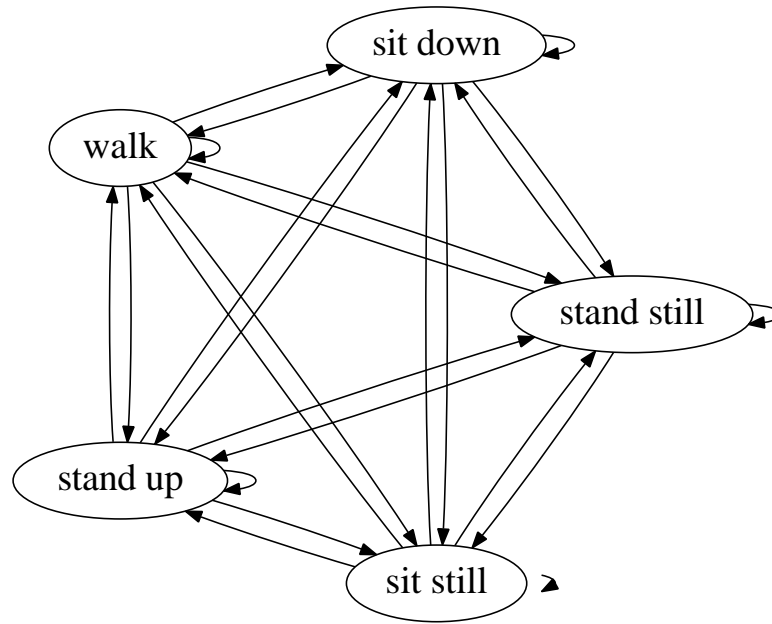
The third Hidden Markov Model is a variation of the former Hidden Markov Model, where the parameters are retrained with Viterbi Learning. The connections between the states of each activity use a language model.

Both the second Hidden Markov Model and the third Hidden Markov Model have a Graph-like structure (Figure 5-2b).

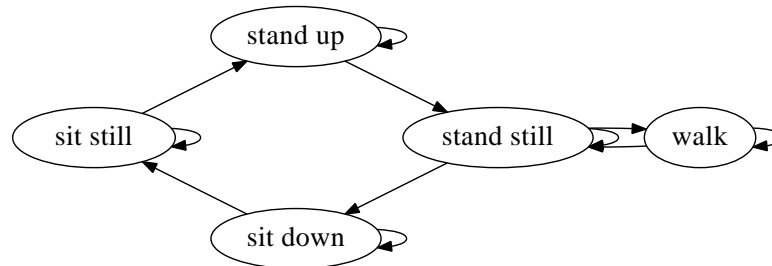
The fourth Hidden Markov Model is the Compound Hidden Markov Model proposed in the Section 3.2 (Figure 5-2c). The connections between the states at the ends of each activity use a language model.

The language model for connecting coherent activities is the following:

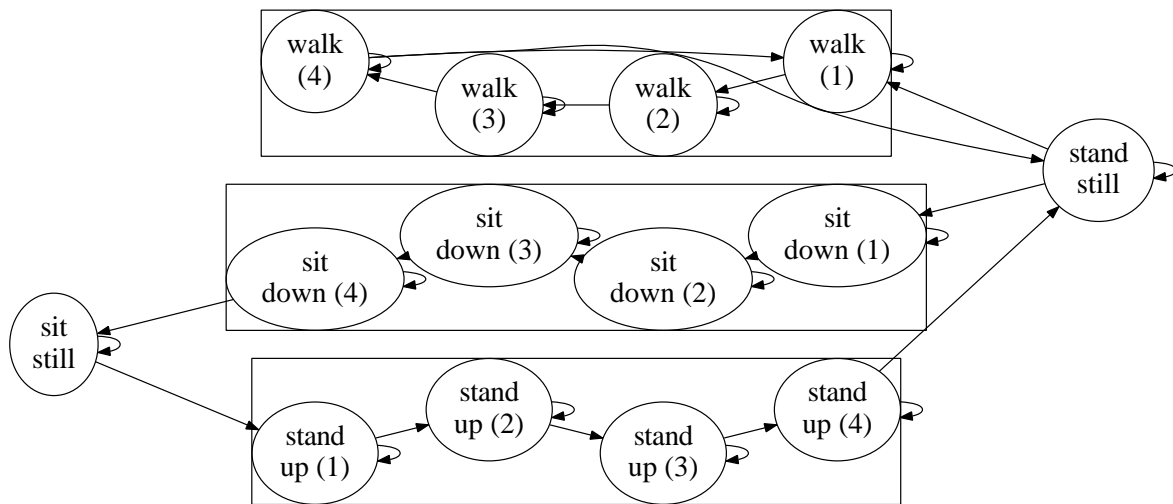
- Sit Still \rightarrow Sit Still.
- Sit Still \rightarrow Stand Up \rightarrow Stand Still.
- Stand Still \rightarrow Stand Still.
- Stand Still \rightarrow Sit Down \rightarrow Sit Still.
- Stand Still \rightarrow Walk \rightarrow Stand Still.



(a) Ergodic Hidden Markov Model.



(b) Graph-like Hidden Markov Model.



(c) Compound Hidden Markov Model.

Figure 5-2: Hidden Markov Models for Activity Labelling.

The sequence of observations \vec{O} is the input for the Compound Hidden Markov Model. The Viterbi algorithm decodes the sequence of observations to a sequence of most likely states \vec{Q} . The states show an activity performed at an instant of time.

The criteria for determining the accuracy of the sequence of most likely states \vec{Q} is *sequence accuracy*. A sequence of states is accurate if the rate between the count of the states which follow the expected sequence of motions and the length of the sequence of states is greater than or equal to a threshold of 90%. The language model for connecting coherent activities specifies the expected sequence of motions for an activity. Repeated states in the sequence are allowed as long as they stay on a expected motion. The sequence accuracy criterion depends on the assessed activity (Table 5-1).

Activity	Expected sequence of motions
Sit Still	{ <i>sit still</i> }
Stand Still	{ <i>stand still</i> }
Stand Up	{ <i>sit still, stand up, stand still</i> }
Sit Down	{ <i>stand still, sit down, sit still</i> }
Walk	{ <i>stand still, walk, stand still(optional)</i> }

Table 5-1: Criteria for Sequence Accuracy per Activity.

5.4 Results

The tests were performed on the four different Hidden Markov Models specified in the section 5.3.1. Each Hidden Markov Model was tested with the following amount of symbols 31, 63, 127, 255, 511, 1023, 2047, and 4095, while keeping the amount of states of each Hidden Markov Model topology. The tables show the Hidden Markov Model with the amount of symbols that provided the highest labelling accuracy.

The assessed data was the sequence of most likely states \vec{Q} computed with the Viterbi Algorithm, on the observations of the motion data. If the percentage of matched states in the sequence is greater than or equal to the labelling threshold, then the activity labelling is correct.

The tables 5-2a and 5-2b show the average labelling accuracy of three codebook sizes for each topology of Hidden Markov Models which gave the highest labelling accuracy for all the activities. The tables 5-3a, 5-3b,5-3c, and 5-3d show the results of the tests on the six subjects

of the training set from the MSRDaily data set. The first column shows the topology of the Hidden Markov Model, the columns 2–4 show the three sizes of codebooks which gave high rates of accuracy for a first activity, and the columns 5–7 show the three sizes of codebooks which gave high rates of accuracy for a second activity.

The tables 5-4a, 5-4b, 5-4c, and 5-4d show the results of the tests on the four subjects of the testing set from the MSRDaily data set. The first column shows the topology of the Hidden Markov Model, the columns 2–4 show the three sizes of codebooks which gave high rates of accuracy for a first activity, and the columns 5–7 show three sizes of codebooks high rates of accuracy for a second activity.

The results for both the training set and the testing set show that the Compound Hidden Markov Model labels correctly a sequence of motion more times than an Ergodic Hidden Markov Model or the Graph-like Hidden Markov Models, when the amount of symbols is lesser than 2047 (Tables 5-2a, 5-2b). The Compound Hidden Markov Model which had the highest labelling accuracy on the testing set has a codebook of 255 symbols.

In the Hidden Markov Models whose codebooks are of 2047 symbols, the Retrained Graph-like Hidden Markov Model had a labelling accuracy similar to the Compound Hidden Markov Model (Tables 5-2a, 5-2b).

It must be noted that the “Walk Occluded” activity is labelled incorrectly by all the Hidden Markov Models. The reason for such failure is that the skeleton data is incorrect or noisy because a sofa occludes the person who is walking. The algorithm which computes the skeleton [Shotton *et al.*, 2011] works correctly when the body is completely visible.

Model (#states)	Codebook size		
	255 symbols	511 symbols	2047 symbols
Ergodic (5)	25.00%	37.50%	58.33%
Graph-like (5)	41.66%	43.75%	54.16%
Graph-like Retrained (5)	41.66%	43.75%	68.75%
Compound (14)	54.16%	54.16%	77.08%

(a) Training Set, Inter-joint Distance Features

Model (#states)	Codebook size		
	255 symbols	511 symbols	2047 symbols
Ergodic (5)	6.25%	15.62%	18.75%
Graph-like (5)	18.75%	12.50%	6.25%
Graph-like Retrained (5)	18.75%	18.75%	53.12%
Compound (14)	59.37%	56.25%	53.12%

(b) Testing Set, Inter-joint Distance Features

Table 5-2: Average labelling accuracy for the Hidden Markov Models with highest accuracy.

Subjects tested	6					
Activity	Sit			Stand		
Model (#states,#symbols)	255	511	2047	255	511	2047
Ergodic (5)	4	6	6	2	3	6
Graph-like (5)	6	6	6	0	0	0
Graph-like Retrained (5)	6	6	6	0	0	0
Compound (14)	6	6	6	0	0	0

(a) Number of subjects, out of 6, with correct labelling on “Sit” and “Stand”.

Subjects tested	6					
Activity	Walk			Walk Occluded		
Model (#states,#symbols)	255	511	2047	255	511	2047
Ergodic (5)	1	4	6	0	0	0
Graph-like (5)	6	6	5	0	0	0
Graph-like Retrained (5)	6	6	6	0	0	2
Compound (14)	6	6	6	4	4	3

(b) Number of subjects, out of 6, with correct labelling on “Walk” and “Walk Occluded”.

Subjects tested	6					
Activity	Stand Up 1			Stand Up 2		
Model (#states,#symbols)	255	511	2047	255	511	2047
Ergodic (5)	3	4	6	0	0	0
Graph-like (5)	5	5	5	0	0	0
Graph-like Retrained (5)	5	5	6	0	0	4
Compound (14)	5	6	6	1	0	4

(c) Number of subjects, out of 6, with correct labelling on “Stand Up 1” and “Stand Up 2”.

Subjects tested	6					
Activity	Sit Down 1			Sit Down 2		
Model (#states,#symbols)	255	511	2047	255	511	2047
Ergodic (5)	2	1	4	0	0	0
Graph-like (5)	3	3	6	0	1	4
Graph-like Retrained (5)	3	3	5	0	1	4
Compound (14)	3	3	6	1	1	6

(d) Number of subjects, out of 6, with correct labelling on “Sit Down 1” and “Sit Down 2”.

Table 5-3: Results on Activity Labelling Accuracy for Inter-joint Distance Features (Training Set).

Subjects tested	4					
Activity	Sit			Stand		
Model (#states,#symbols)	255	511	2047	255	511	2047
Ergodic (5)	0	0	0	1	2	2
Graph-like (5)	0	0	0	1	1	0
Graph-like Retrained (5)	0	1	2	1	1	1
Compound (14)	2	2	2	3	2	2

(a) Number of subjects, out of 4, with correct labelling on “Sit” and “Stand”.

Subjects tested	4					
Activity	Walk			Walk Occluded		
Model (#states,#symbols)	255	511	2047	255	511	2047
Ergodic (5)	1	2	4	0	0	0
Graph-like (5)	4	1	1	0	1	0
Graph-like Retrained (5)	4	2	4	1	1	3
Compound (14)	4	4	4	2	2	2

(b) Number of subjects, out of 4, with correct labelling on “Walk” and “Walk Occluded”.

Subjects tested	4					
Activity	Stand Up 1			Stand Up 2		
Model (#states,#symbols)	255	511	2047	255	511	2047
Ergodic (5)	0	1	0	0	0	0
Graph-like (5)	0	1	1	0	0	0
Graph-like Retrained (5)	0	1	2	0	0	2
Compound (14)	3	2	2	2	2	3

(c) Number of subjects, out of 4, with correct labelling on “Stand Up 1” and “Stand Up 2”.

Subjects tested	4					
Activity	Sit Down 1			Sit Down 2		
Model (#states,#symbols)	255	511	2047	255	511	2047
Ergodic (5)	0	0	0	0	0	0
Graph-like (5)	0	0	0	1	0	0
Graph-like Retrained (5)	0	0	2	0	0	1
Compound (14)	1	1	2	2	3	0

(d) Number of subjects, out of 4, with correct labelling on “Sit Down 1” and “Sit Down 2”.

Table 5-4: Results on Activity Labelling Accuracy for Inter-joint Distance Features (Testing Set).

Chapter 6

Conclusions and Future Work

The cake is a lie.

PORTAL

We present results for labelling human activity from skeleton data of a single KinectTM sensor. We present a novel way of computing features of a skeleton using distances between certain joints of both upper body and lower body. And, we propose a Compound Hidden Markov Model for labelling cyclic and non-cyclic human activities, which perform better than the reference Hidden Markov Models, an Ergodic Hidden Markov Model and a Graph-like Hidden Markov Model. The results for labelling five activities from four non-trained subjects show that the Compound Hidden Markov Model, with a codebook of 255 symbols, labels correctly a sequence of motion with an average accuracy of 59.37%, which is higher than the average labelling accuracy for activities of unknown subjects of an Ergodic Hidden Markov Model (6.25%), and a Compound Hidden Markov Model with activities modelled by a single state (18.75%), both with a codebook of 255 symbols. The contributions of this work are the representation of a full body pose with Euclidean distances between certain pairs of body joints, and the method for training a Compound Hidden Markov Model for activity labelling by segmenting the training data with the Anticipation-Action-Reaction sections from theory of animation. The future work involves using a new representation for the skeleton, based on trees of Orthogonal Direction Change Chain Codes [Bribiesca, 2000, 2008], for both the codebook and the input samples.

Appendix A

Orthogonal Direction Change Chain Code

An approach for computing features of a skeleton is the Orthogonal Direction Change (ODC) Chain Code [Bribiesca, 2000], which digitizes 3-D curves into a set of codes which represent orthogonal direction changes between three constant length segments of a 3-D curve $(\vec{u}, \vec{v}, \vec{w})$ (Equation A-1), which are aligned to the corners of a 3-D grid with constant-sized cells.

As the orthogonal direction changes are relative, these ODC Chain Codes have some interesting properties: invariance to translation, invariance to rotation, invariance to mirroring and invariance to starting point. The invariance to rotation and translation allow to represent a large set of curves generated by absolute direction changes, such as orthogonal 3-D Freeman codes, using only one ODC Chain Code [Bribiesca and Velarde, 2001].

There are five different orthogonal direction changes for representing any 3-D curve (Figure A-1), as explained in the work of Bribiesca [Bribiesca, 2000]:

- The Chain Element “0” represents a direction change which *goes straight* through the contiguous straight-line segments following the direction of the last segment.
- The Chain Element “1” represents a direction change to the *right*.
- The Chain Element “2” represents a direction change *upward* (stair-case fashion).
- The Chain Element “3” represents a direction change to the *left*.

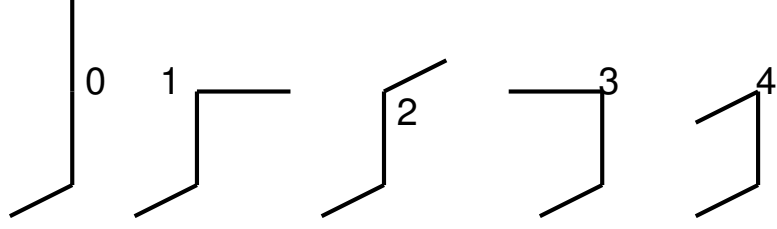


Figure A-1: Orthogonal Direction Change Chain Elements.

- The Chain Element “4” represents a direction change which is *going back*.

$$chain\ element(\vec{u}, \vec{v}, \vec{w}) = \begin{cases} 0, & \text{if } \vec{w} = \vec{v}; \\ 1, & \text{if } \vec{w} = \vec{u} \times \vec{v}; \\ 2, & \text{if } \vec{w} = \vec{u}; \\ 3, & \text{if } \vec{w} = -(\vec{u} \times \vec{v}); \\ 4, & \text{if } \vec{w} = -\vec{u} \end{cases} \quad (\text{A-1})$$

Angular resolution

A 3-D vector has a representation in spherical coordinates. The spherical coordinates have three components: radial distance (r), polar angle (θ) and azimuthal angle (ϕ). For these formulas, the convention for the 3-D space is that the XY plane is parallel to the floor and the z -axis is perpendicular to the XY plane.

$$\begin{aligned} r &= \sqrt{x^2 + y^2 + z^2} \\ \theta &= \arccos\left(\frac{z}{r}\right) \\ \phi &= \arctan\left(\frac{y}{x}\right) \end{aligned}$$

For this work, the convention for the spherical coordinates is adjusted to the coordinate system of the KinectTM sensor, where the XZ plane is parallel to the floor and the y -axis is perpendicular to the XZ plane.

$$\begin{aligned}
r &= \sqrt{x^2 + y^2 + z^2} \\
\theta &= \arccos\left(\frac{y}{r}\right) \\
\phi &= \arctan\left(\frac{z}{x}\right)
\end{aligned}$$

When an unit sphere is subdivided using a 3-D grid, the spherical coordinates of any point in the sphere are rounded to the coordinates at the nearest corner in the grid. The minimum angle which can be represented in the grid α is the inverse tangent of the inverse of the subdivisions D on the 3-D grid.

$$\alpha = \arctan\left(\frac{1}{D}\right), D \in \mathbb{N}$$

Mirroring of an ODC Chain Code

The ODC Chain Codes encode relative direction changes, and there are two ODC Chain Codes which have a mirroring property on the central axis: direction change to the left (1) and direction change to the right (3). These ODC Chain Codes are formed by segments which are set on the three coordinate axes X, Y, Z .

The mirroring property becomes evident when any of the segments the digitized curve is replaced by a segment with the opposite direction, as long as each segment points to an unique coordinate axis.

In any curve converted to a string of ODC Chain Codes, if all the ODCChain Codes representing a direction change to the left are replaced with ODC Chain Codes representing direction changes to the right and viceversa (Figure A-2a), the result is a mirrored version of the string of ODC Chain Codes (Figure A-2b).

Digitization of a Three-Dimensional Curve

In order to convert a set of 3-D lines into a set of line segments of constant length, the first step consists in aligning the vertices, $\vec{p} \in \mathbb{R}^3$ and $\vec{q} \in \mathbb{R}^3$, to the corners of the 3-D grid, by rounding

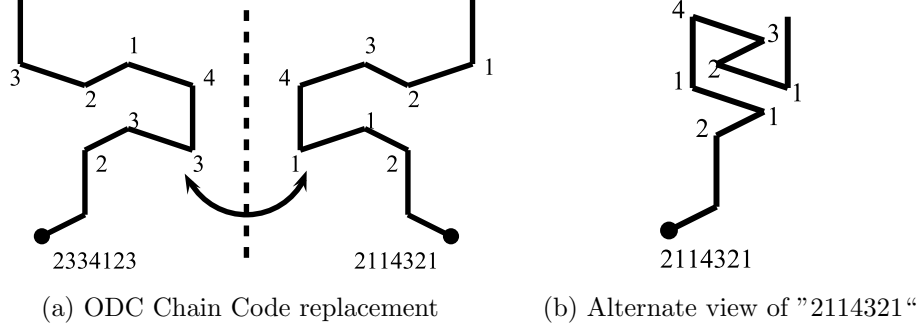


Figure A-2: Mirror of a ODC Chain Code sequence.

the values of \vec{p} and \vec{q} , according to the smallest distance on each axis between the vertex and the neighbour values on the grid, \vec{g}_i and \vec{g}_j , obtaining the vertices \vec{p}' and \vec{q}' (Equation A-2).

$$\vec{v} \in \mathbb{R}^3 \quad (\text{A-2a})$$

$$\vec{g} \in \mathbb{R}^3 \quad (\text{A-2b})$$

$$\vec{g}_i = \{\text{floor}(\vec{v}_x), \text{floor}(\vec{v}_y), \text{floor}(\vec{v}_z)\} \quad (\text{A-2c})$$

$$\vec{g}_j = \{\text{ceil}(\vec{v}_x), \text{ceil}(\vec{v}_y), \text{ceil}(\vec{v}_z)\} \quad (\text{A-2d})$$

$$\vec{v}'_x = \begin{cases} \vec{g}_{i_x}, & \text{if } \vec{g}_{i_x} \leq \vec{v}_x < \frac{\vec{g}_{i_x} + \vec{g}_{j_x}}{2} < \vec{g}_{j_x}; \\ \vec{g}_{j_x}, & \text{otherwise} \end{cases} \quad (\text{A-2e})$$

$$\vec{v}'_y = \begin{cases} \vec{g}_{i_y}, & \text{if } \vec{g}_{i_y} \leq \vec{v}_y < \frac{\vec{g}_{i_y} + \vec{g}_{j_y}}{2} < \vec{g}_{j_y}; \\ \vec{g}_{j_y}, & \text{otherwise} \end{cases} \quad (\text{A-2f})$$

$$\vec{v}'_z = \begin{cases} \vec{g}_{i_z}, & \text{if } \vec{g}_{i_z} \leq \vec{v}_z < \frac{\vec{g}_{i_z} + \vec{g}_{j_z}}{2} < \vec{g}_{j_z}; \\ \vec{g}_{j_z}, & \text{otherwise} \end{cases} \quad (\text{A-2g})$$

When the distance between two vertices on the 3-D grid is longer than the size of the cell, additional vertices on each axis are added to the line from the components of the Manhattan distance of \vec{p}' and \vec{q}' , where $d(\vec{p}', \vec{q}') = |\vec{p}'_x - \vec{q}'_x| + |\vec{p}'_y - \vec{q}'_y| + |\vec{p}'_z - \vec{q}'_z|$ [Krause, 1987], to compute a set of constant-length line segments between two vertices.

Finally, the ODC Chain Codes are computed by taking three consecutive line segments,

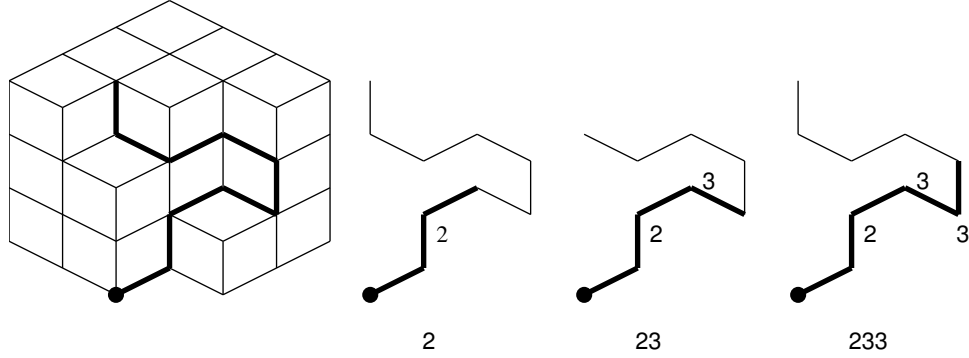


Figure A-3: Example of ODC Chain Code Sequence.

starting from the first line segment, and applying the rules of orthogonal direction changes to compute the corresponding chain element (Figure A-3).

The 3-D joint data is captured by a 3-D vision system, such as the KinectTM, which computes the joint data from the depth map captured by the sensor. The frame of reference for the coordinates of each joint is relative to the plane of the infrared sensor. The axis of coordinates are set as follows: the X axis increases towards the right of the plane of the sensor, the Y axis increases towards the top of the plane of the sensor, and the Z axis increases when moving away from the plane of the sensor. The joint data is organized in a humanoid skeleton hierarchy.

For purposes of this research, the joints are numbered in this way (Figure A-4): *0*) Head (0); *1*) Neck (1); *2*) Left Shoulder (2); *3*) Left Elbow (3); *4*) Left Hand (4); *5*) Right Shoulder (5); *6*) Right Elbow (6); *7*) Right Hand (7); *8*) Torso (8) *9*) Left Hip (9); *10*) Left Knee (A); *11*) Left Foot (B); *12*) Right Hip (C); *13*) Right Knee (D); *14*) Right Hip (E).

There are a couple of factors which have significant impact in the digitization of the skeleton data to ODC Chain Codes: the noise of the sensor and the angle of orientation of the body. The former affects the length of each limb, resulting in ODC Chain Codes of variable length; while the latter affects the proportion of orthogonal segments along a ODC Chain Code, which has negative effects in the algorithms which match ODC Chain Codes.

The noise of the KinectTM sensor and the arithmetical precision on the algorithm which computes the skeleton from depth data [Shotton *et al.*, 2011] are the causes of variation on the location of the body parts on the skeleton. When this noisy data is converted to a set of ODC Chain Codes, the length each element of the set varies according to the length of

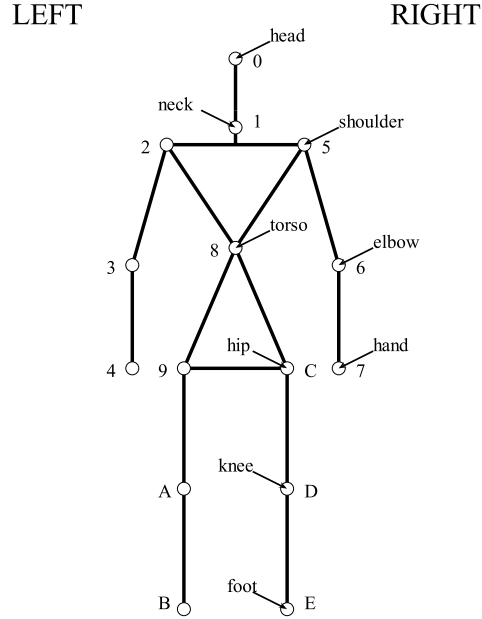


Figure A-4: Three-dimensional Joint Data Hierarchical Structure.

the corresponding body part. To correct those variations, the original skeleton is converted to a skeleton made of unit vectors and scaled by a magnitude value according to the angular resolution of the 3-D grid.

Orientation of the Skeleton

An skeleton has different ODC Chain Code representations when the skeleton is rotated and not aligned against an orthogonal plane.

The body is aligned against the frontal plane (XY) of the field of view of the KinectTM sensor with a projection over three orthogonal unit vectors, u, v, n . The forward vector of the body, n , is the cross product between the vector formed by the joint of the torso and the joint of the left shoulder (S_l), and the vector formed by the joint of the torso and the right shoulder (S_r). The up vector of the body, v , is the sum of S_l and S_r . And, the right vector of the body is the cross product between the forward vector of the body (u) and the up vector of the body (v). Those vectors are set in a projection matrix, W , which transforms the skeleton (S) to project it to the frontal plane (S_f).

The joints of the skeleton are set relative to a reference joint, the torso in this work.

Skeleton Features

The features of the skeleton are represented by a single sequence of ODC Chain Code symbols, or Skeleton Chain Code Signature, by linking consecutive joints of the skeleton. The order of connection between joints is: neck, left shoulder, left elbow, left hand, right shoulder, right elbow, right hand, torso, left hip, left knee, left foot, right hip, right knee and right foot.

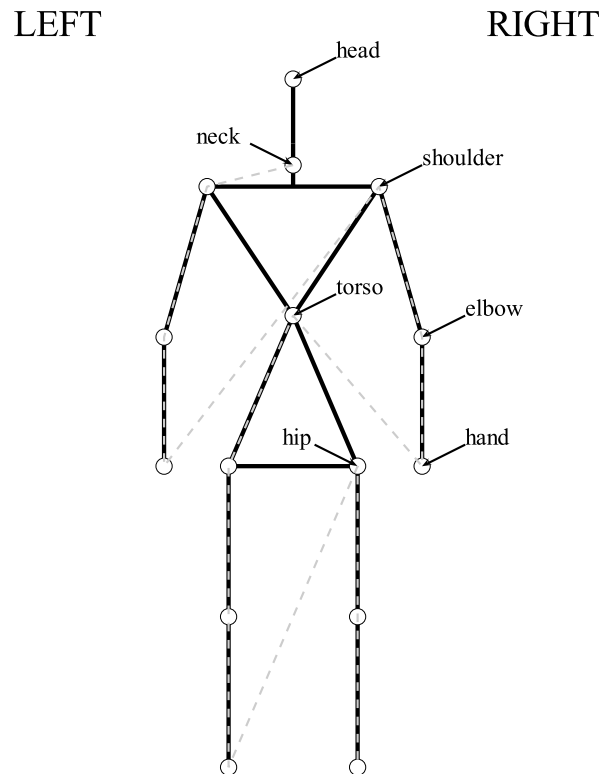


Figure A-5: Joint Order for a Skeleton Chain Code Signature.

The Skeleton Chain Code Signature represents the skeleton as a single three-dimensional line. When the Signature is projected on the XY plane, it has a similar shape to the projection of the original skeleton on the XY plane.

Stretch-Twist Disparity

Since each ODC Chain Code represents a relative change of direction and a sequence of ODC Chain Codes can be considered as a string of characters, the algorithms for measuring similarity

based on Euclidean Distance can not be used directly to figure if two sequences of ODC Chain Codes are similar. Therefore, other approach must be used, such as techniques for text mining.

The algorithm for matching sequences of ODC Chain Codes is based in the Stretch Twist Dissimilarity [Wang *et al.*, 2009], which is based in counting the amount of ODC Chain Codes which are equal (Equality), the amount of insertions of symbols in the shortest sequence of ODC Chain Codes (Stretch), and the amount of substitutions of symbols when the ODC Chain Codes in both sequences are not equal (Twist), between two sequences of ODC Chain Codes P and Q .

The time complexity of this algorithm is linear and the memory requirements are $2(\max(\|S\|, \|T\|))$ symbols and the dissimilarity is bound in the range $[0, 1]$, being zero when both sequences are equal and one when both sequences are totally different (Equation A-3).

$$D(P, Q) = \frac{\varsigma + \vartheta}{S + \varsigma + \vartheta} \tag{A-3}$$

ς = Amount of Stretches

ϑ = Amount of Twists

S = Amount of Equalities

Bibliography

- AGGARWAL, J. AND RYOO, M. Human activity analysis: A review. *ACM Comput. Surv.* **43**(3):16:1–16:43 (2011)
- BOBICK, A., IVANOV, Y., BOBICK, A.F., AND IVANOV, Y.A. Action Recognition using Probabilistic Parsing. In *In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 196–202 (1998)
- BOBICK, A. AND DAVIS, J. The recognition of human movement using temporal templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **23**(3):257–267 (2001)
- BRAND, M., OLIVER, N., AND PENTLAND, A. Coupled hidden Markov models for complex action recognition. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 994 –999 (1997)
- BRIBIESCA, E. A chain code for representing 3D curves. *Pattern Recognition* **33**(5):755 – 765 (2000)
- BRIBIESCA, E. A method for representing 3D tree objects using chain coding. *J. Vis. Commun. Image Represent.* **19**:184–198 (2008)
- BRIBIESCA, E. AND VELARDE, C. A formal language approach for a 3D curve representation. *Computers & Mathematics with Applications* **42**(12):1571 – 1584 (2001)
- CAMPBELL, L. AND BOBICK, A. Recognition of human body motion using phase space constraints. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 624–630 (1995)

- CHEN, H.S., CHEN, H.T., CHEN, Y.W., AND LEE, S.Y. Human action recognition using star skeleton. In *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, VSSN '06, pages 171–178. ACM, New York, NY, USA (2006)
- DARRELL, T. AND PENTLAND, A. Space-time gestures. In *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR '93., 1993 IEEE Computer Society Conference on*, pages 335–340 (1993)
- DAWES, B., ABRAHAMS, D., AND JOSUTTIS, N. BOOST C++ Libraries (2001). <http://www.boost.org>
- DUONG, T.V., BUI, H.H., PHUNG, D.Q., AND VENKATESH, S. Activity recognition and abnormality detection with the switching hidden semi-Markov model. In *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition CVPR 2005*, volume 1, pages 838–845 (2005)
- FINK, G. *Markov Models for Pattern Recognition: From Theory to Applications*. SpringerLink: Springer e-Books. Springer (2007)
- GAVRILA, D. AND DAVIS, L. 3-D model-based tracking of humans in action: a multi-view approach. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on*, pages 73–80 (1996)
- GLODEK, M., LAYHER, G., SCHWENKER, F., AND PALM, G. Recognizing Human Activities Using a Layered Markov Architecture. In A. Villa, W. Duch, P. rdi, F. Masulli, and G. Palm (editors), *Artificial Neural Networks and Machine Learning ICANN 2012, Lecture Notes in Computer Science*, volume 7552, pages 677–684. Springer Berlin Heidelberg (2012a)
- GLODEK, M., SCHWENKER, F., AND PALM, G. Detecting actions by integrating sequential symbolic and sub-symbolic information in human activity recognition. In *Proceedings of the 8th international conference on Machine Learning and Data Mining in Pattern Recognition, MLDM'12*, pages 394–404. Springer-Verlag, Berlin, Heidelberg (2012b)
- GONG, S. AND XIANG, T. Recognition of group activities using dynamic probabilistic networks.

- In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 742–749 vol.2 (2003)
- GUENNEBAUD, G., JACOB, B. *et al.* Eigen v3. <http://eigen.tuxfamily.org> (2010)
- GUENTERBERG, E., GHASEMZADEH, H., LOSEU, V., AND JAFARI, R. Distributed Continuous Action Recognition Using a Hidden Markov Model in Body Sensor Networks. In *Proceedings of the 5th IEEE International Conference on Distributed Computing in Sensor Systems, DCOSS '09*, pages 145–158. Springer-Verlag, Berlin, Heidelberg (2009)
- KANUNGO, T. *Extended Finite State Models of Language*, chapter UMDHMM: Hidden Markov Model Toolkit. Cambridge University Press (1999)
- KE, Y., SUKTHANKAR, R., AND HEBERT, M. Spatio-temporal Shape and Flow Correlation for Action Recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8 (2007)
- KRAUSE, E. *Taxicab Geometry: An Adventure in Non-Euclidean Geometry*. Dover Publ. (1987)
- LASSETER, J. Principles of Traditional Animation Applied to 3D Computer Animation. *SIG-GRAPH Comput. Graph.* **21**(4):35–44 (1987)
- LOWERRE, B.T. *The Harpy Speech Recognition System*. Tesis Doctoral, Carnegie Mellon University, Pittsburgh, PA, USA (1976). AAI7619331
- MAGEE, D.R. AND BOYLE, R.D. Detecting lameness using ‘Re-sampling Condensation’ and ‘multi-stream cyclic hidden Markov models’. *IMAGE AND VISION COMPUTING* **20**:2002 (2002)
- NATARAJAN, P. AND NEVATIA, R. Coupled Hidden Semi Markov Models for Activity Recognition. In *Proceedings of the IEEE Workshop on Motion and Video Computing, WMVC '07*, pages 10–. IEEE Computer Society, Washington, DC, USA (2007)
- NERGUI, M., YOSHIDA, Y., IMAMOGLU, N., GONZALEZ, J., AND YU, W. Human behavior recognition by a bio-monitoring mobile robot. In *Proceedings of the 5th international conference on Intelligent Robotics and Applications - Volume Part II, ICIRA'12*, pages 21–30. Springer-Verlag, Berlin, Heidelberg (2012)

- NGUYEN-DUC-THANH, N., LEE, S., AND KIM, D. Two-stage Hidden Markov Model in gesture recognition for human robot interaction. *International Journal of Advanced Robotic Systems* **9** (2012). Cited By (since 1996) 0
- OH, C.M., ISLAM, M.Z., PARK, J.W., AND LEE, C.W. A gesture recognition interface with upper body model-based pose tracking. In *Proc. 2nd Int Computer Engineering and Technology (ICCET) Conf*, volume 7 (2010)
- OLIVER, N., ROSARIO, B., AND PENTLAND, A. A Bayesian Computer Vision System for Modeling Human Interactions. In *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, volume 22. IEEE (2000)
- OLIVER, N., HORVITZ, E., AND GARG, A. Layered Representations for Human Activity Recognition. In *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces, ICMI '02*, pages 3–. IEEE Computer Society, Washington, DC, USA (2002)
- OPENNI. OpenNI. <http://www.openni.org> (2009)
- RABINER, L. AND JUANG, B.H. *Fundamentals of Speech Recognition*. United states ed edition. Prentice Hall (1993)
- RABINER, L.R. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286 (1989)
- RAO, C. AND SHAH, M. View-invariance in action recognition. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–316–II–322 vol.2 (2001)
- RYOO, M.S. AND AGGARWAL, J.K. Recognition of Composite Human Activities through Context-Free Grammar Based Representation. *2012 IEEE Conference on Computer Vision and Pattern Recognition* **2**:1709–1718 (2006)
- RYOO, M.S. AND AGGARWAL, J. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1593–1600 (2009)

- SAVAGE, J. *A hybrid system with symbolic AI and statistical methods for speech recognition.* Thesis Doctoral, University of Washington, Seattle, WA, USA (1995). UMI Order No. GAX96-09599
- SHECHTMAN, E. AND IRANI, M. Space-time behavior based correlation. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 405–412 vol. 1 (2005)
- SHEIKH, Y., SHEIKH, M., AND SHAH, M. Exploring the space of a human action. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 144–149 Vol. 1 (2005)
- SHI, Q., WANG, L., CHENG, L., AND SMOLA, A. Discriminative human action segmentation and recognition using semi-Markov model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8 (2008)
- SHOTTON, J., FITZGIBBON, A., COOK, M., SHARP, T., FINOCCHIO, M., MOORE, R., KIPMAN, A., AND BLAKE, A. Real-time human pose recognition in parts from single depth images. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 1297–1304. IEEE Computer Society, Washington, DC, USA (2011)
- STARNER, T.E. AND PENTLAND, A. Visual Recognition of American Sign Language Using Hidden Markov Models (1995)
- SUNG, J., PONCE, C., SELMAN, B., AND SAXENA, A. Human Activity Detection from RGBD Images. Technical report, Carnegie Mellon University, Department of Computer Science, Cornell University, Ithaca, NY 14850 (2011)
- SUNG, J., PONCE, C., SELMAN, B., AND SAXENA, A. Unstructured human activity detection from RGBD images. In *ICRA*, pages 842–849. IEEE (2012)
- VINTSYUK, T. Speech discrimination by dynamic programming. *Cybernetics* **4**(1):52–57 (1968)
- WANG, J., LIU, Z., WU, Y., AND YUAN, J. Mining actionlet ensemble for action recognition with depth cameras. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1290–1297 (2012)

- WANG, T., CHENG, I., LOPEZ, V., BRIBIESCA, E., AND BASU, A. Valence Normalized Spatial Median for skeletonization and matching. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 55–62 (2009)
- WHITTLE, M. *Gait Analysis: An Introduction*. Gait Analysis. Butterworth-Heinemann (2002)
- WILLIAMS, R. *The Animator's Survival Kit*. 2nd edition. Faber & Faber (2009)
- WONG, K.Y.K., KIM, T.K., AND CIPOLLA, R. Learning Motion Categories using both Semantic and Structural Information. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–6 (2007)
- XIA, L., CHEN, C.C., AND AGGARWAL, J. View Invariant Human Action Recognition Using Histograms of 3D Joints. In *The 2nd International Workshop on Human Activity Understanding from 3D Data (HAU3D)* (2012)
- YACOOB, Y. AND BLACK, M. Parameterized modeling and recognition of activities. In *Computer Vision, 1998. Sixth International Conference on*, pages 120–127 (1998)
- YAMATO, J., OHYA, J., AND ISHII, K. Recognizing human action in time-sequential images using hidden Markov model. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, pages 379–385 (1992)
- YILMA, A. AND SHAH, M. Recognizing human actions in videos acquired by uncalibrated moving cameras. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 150–157 Vol. 1 (2005)
- YU, E. AND AGGARWAL, J.K. Detection of Fence Climbing from Monocular Video. In *Proceedings of the 18th International Conference on Pattern Recognition - Volume 01, ICPR '06*, pages 375–378. IEEE Computer Society, Washington, DC, USA (2006)
- ZHANG, D., GATICA-PEREZ, D., BENGIO, S., AND MCCOWAN, I. Modeling individual and group actions in meetings with layered HMMs. *Multimedia, IEEE Transactions on* **8**(3):509 – 520 (2006)