Machine Learning
for Gesture Recognition
from Videos

Proefschrift

ter verkrijging van de graad van doctor
aan de Radboud Universiteit Nijmegen
op gezag van de rector magnificus prof. dr. Th.L.M. Engelen,
volgens besluit van het college van decanen
in het openbaar te verdedigen op maandag 9 februari 2015
om 12.30 uur precies

door

Binyam Gebrekidan Gebre

geboren op 1 april 1983
te Mekelle, Ethiopië

Promotoren:

Prof. dr. Tom Heskes
Prof. dr. Stephen C. Levinson

Copromotor:

Peter Wittenburg (MPI)

Manuscriptcommissie:

Prof. dr. A.P.J. van den Bosch
Prof. dr. E.O. Postma (Tilburg University)
Dr. X. Anguera (Telefónica Research, Madrid, Spanje)

The research presented in this thesis has received funding from the European Commission's 7th Framework Program under grant agreement no 238405, project CLARA, Marie Curie ITN.

ISBN 978-90-76203-62-1
Gedrukt door Ipskamp Drukkers

Machine Learning for Gesture Recognition from Videos

Doctoral Thesis

to obtain the degree of doctor
from Radboud University Nijmegen
on the authority of the Rector Magnificus prof. dr. Th.L.M. Engelen,
according to the decision of the Council of Deans
to be defended in public on Monday, 9 February, 2015
at 12.30 hours

by

Binyam Gebrekidan Gebre

born on 1 April, 1983
in Mekelle, Ethiopia

Supervisors:

Prof. dr. Tom Heskes
Prof. dr. Stephen C. Levinson

Co-supervisor:

Peter Wittenburg (MPI)

Doctoral Thesis Committee:

Prof. dr. A.P.J. van den Bosch
Prof. dr. E.O. Postma (Tilburg University)
Dr. X. Anguera (Telefónica Research, Madrid, Spain)

The research presented in this thesis has received funding from the European Commission's 7th Framework Program under grant agreement no 238405, project CLARA, Marie Curie ITN.

ISBN 978-90-76203-62-1
Printed by Ipskamp Drukkers

Contents

Title page i

Table of Contents v

Acknowledgments ix

1 Introduction **1**
- 1.1 Motivation 3
- 1.2 Problem statement 4
- 1.3 Research approach 5
  - 1.3.1 Divide and conquer 5
  - 1.3.2 Attack subproblems 5
  - 1.3.3 Evaluate solutions 7
- 1.4 Summary of contributions 8
- 1.5 Structure of this thesis 10

2 Speaker diarization: the gesturer is the speaker **11**
- 2.1 Introduction 13
- 2.2 Gesture-speech relationship 13
  - 2.2.1 Gestures occur mainly during speech 14
  - 2.2.2 DAF does not interrupt speech-gesture synchrony 14
  - 2.2.3 The congenitally blind also gesture 15
  - 2.2.4 Fluency affects gesturing 15
- 2.3 Our diarization algorithm 15
- 2.4 Experiments 17
  - 2.4.1 Dataset 17
  - 2.4.2 Evaluation metrics 19
- 2.5 Results and discussion 20
- 2.6 Conclusions and future work 22
- 2.7 Related work 22
  - 2.7.1 Preprocessing 24
  - 2.7.2 Segmentation and clustering 24

3 Signer diarization:
the gesturer is the signer 27

3.1 Introduction 29

3.2 Motivation 29

3.3 Signer diarization complexity 30

3.4 Our signer diarization algorithm 30

3.4.1 Algorithm 31

3.4.2 Implementation 32

3.5 Experiments 32

3.5.1 Datasets 32

3.5.2 Evaluation metrics 33

3.6 Results and discussion 34

3.7 Conclusions and future work 36

4 Motion History Images for online diarization 37

4.1 Introduction 39

4.2 Gesture representation 40

4.2.1 Motion Energy Image 40

4.2.2 Motion History Image 41

4.3 The online diarization system 42

4.3.1 Conversation dynamics 42

4.3.2 Gesture model 42

4.4 Experiments 43

4.4.1 Datasets 43

4.4.2 Evaluation metrics 44

4.5 Results and discussion 44

4.5.1 Speaker diarization 44

4.5.2 Signer diarization 46

4.6 Conclusions and future work 47

4.7 Relation to prior work 47

5 Speaker diarization using gesture and speech 49

5.1 Introduction 51

5.2 Speech-gesture representation 51

5.2.1 Speech representation 52

5.2.2 Gesture representation 52

5.3 Our diarization system 53

5.3.1 Diarization using gestures 53

5.3.2 Diarization using speaker models 54

5.4 Experiments 56

5.4.1 Datasets 56

5.4.2 Evaluation metrics 57

5.5 Results and discussion 57

5.6 Conclusions and future work 59

6 Automatic sign language identification 61

6.1 Introduction 63

6.2 Sign language phonemes 63

6.3 Our sign language identification method 64

6.3.1 Skin detection 64

6.3.2 Feature extraction 65

6.3.3 Learning using random forest 66

6.3.4 Identification 67

6.4 Experiment 67

6.5 Results and discussion 68

6.6 Conclusions and future work 71

7 Feature learning for sign language identification 73

7.1 Introduction 75

7.2 The challenges in sign language identification 76

7.2.1 Iconicity in sign languages 76

7.2.2 Differences between signers 76

7.2.3 Diverse environments 76

7.3 Feature and classifier learning 77

7.3.1 Unsupervised feature learning 77

7.3.2 Classifier learning 79

7.4 Experiments 80

7.4.1 Datasets 80

7.4.2 Data preprocessing 81

7.4.3 Evaluation 81

7.5 Results and discussion 81

7.6 Conclusions and future work 85

8 Gesture stroke detection 87

8.1 Introduction 89

8.2 Gesture stroke 90

8.3 Our stroke detection method 92

8.3.1 Face and hand detection 92

8.3.2 Feature extraction 92

8.3.3 Classification 94

8.4 Experiments 94

8.4.1 Datasets 94

8.4.2 Evaluation 95

8.5 Results and discussion 95

8.6 Conclusions and future work 98

9 Conclusions **99**

9.1 Introduction 101

9.2 Summary: speaker diarization 103

9.3 Summary: signer diarization 105

9.4 Summary: sign language identification 106

9.5 Summary: gesture stroke detection 107

9.6 Putting it all together 108

Bibliography **109**

Summary **121**

Samenvatting **125**

Publications **129**

Curriculum Vitae **131**

Acknowledgments

Peter Wittenburg

Behind every completed thesis, there is a strong support system. The root of the support system, in my case, is Peter Wittenburg. Peter, I cannot thank you enough for the opportunity and the support! Your energy and enthusiasm for technologies and for people are contagious. Not only did I benefit from them, I also learned from them. This PhD thesis is a child of the AVATecH project, an ambitious and creative project you envisioned. I very much enjoyed it, thank you!

Special thanks also go to Jacquelijn Ringersma, who together with Peter interviewed me in London and offered me the position that resulted in this thesis.

Tom Heskes

Professor Tom Heskes, expert in machine learning and intelligent systems, played a critical role in the success of this PhD thesis and the publications in it. I was fortunate to have him as the main promotor of my thesis. We had biweekly meetings to discuss progress and exchanged many emails during paper deadlines. Tom, I cannot thank you enough for the timely discussions and critical feedback!

Stephen C. Levinson

Professor Stephen C. Levinson has been helpful in guiding and supporting the PhD thesis with ideas and administrative support. He has contributed important ideas to the thesis. The funding for my travel to the 2014 Interspeech conference was made possible by him. Thank you Steve!

Collaborators

I benefited a lot from collaborators who shared their data, tools or knowledge. I thank Onno Crasborn, Marijn Huijbregts, Asli Ozyurek, Connie de Vos, Marcos Zampieri, Mingyuan Chu and Emanuela Campisi. I thank Marcos for keeping me interested in natural language processing (text processing). We worked together on native language identification and language variety identification.

MPI Community

I would like to thank the following people for their support in administrative, technical and social matters. *TLA and TG members*: Sebastian, Daan, Gunter, Paul, Han, Menzo, Ad, Tobias, Albert, Reiner, Aarthy, Herman, Huib, Lari, Przemek and Anna, André, Florian, Alex (my paranymph), Guilherme (my paranymph), Eric, Willem, Olaf, Olha, Twan, Kees Jan, Sander and many others. *Administration*: Nanjo, Edith, Angela, Marie-Luise, Uschi and Jan. *Library*: Karin, Meggie and Annemieke for excellent library services. *Canteen*: Thea and Pim for excellent canteen services. *Friends*: Rebecca, Salomi, Sylvia, Ewelina, Julija, Gabriela, Elizabeth, Annemarie, Jeremy, Sean, Mark, Tyko, Rick, Suzanne, Rósa, Sho, Amaia and many others (enjoyed talking with you all). *Football*: Peter, Guilherme, Francisco, Joost, Alastair, Florian, Paul, Harald, Marisa, Giovanni, Matthias, Varun, Alessandro and many others. Thank you guys, it was fun to play football with all of you.

Family and friends

My successes in school are due to the love and encouragement of my family and friends. *My family*: Nigisti Abraha, Ghebremedhin Belay, Gebrekidan Gebre, Kiduse Gebreyohannes, my grandparents and many cousins, uncles and aunts. *My friends*: Nesredin, Asfaw, Mizan, and many other MITians and Kellaminoers.

Saskia van Putten

On a personal side, I would like to thank my lovely girlfriend, Saskia. We have shared the joys and pains of doing a PhD. I thank her for proofreading my thesis and for translating the summary into Dutch. Thanks to her, I found the motivation to take a Dutch course (Ja, ik kan een beetje Nederlands spreken!). I would also like to thank her family for the warm welcome and good times.

Chapter 1

Introduction

Content

This chapter presents context to the work presented in the thesis. It highlights the challenges of annotating videos manually and indicates how a machine with a capacity to learn can help. The chapter also presents summaries of the contributions made in the areas of speaker diarization, signer diarization, sign language identification and gesture stroke detection.

Keywords

Big data, motivation, problem statement, gestures, research approach, machine learning, summary of contributions, structure of the thesis

1.1 Motivation

Video data is growing bigger and bigger. What should we do to make sense of it?

With advances in device technology, it has become much easier for virtually anyone to record, collect and store data. This ease has resulted in data volumes of a scale never seen before, hence called big data. This big data offers new opportunities, because we can raise new questions that we would not have raised otherwise. However, these new questions cannot be answered without parallel advances in technologies that are capable of analyzing non-structured data such as audio and video recordings. The goal of this thesis is to advance technologies used in audio-video content analysis.

The machines we have today are fast but not intelligent yet; they cannot yet understand audio-video content. For this reason, currently, the common practice is that human expertise is required in understanding and annotating the content of audio-video for purposes of, for example, empirical research in the humanities and social sciences. But the use of human expertise in understanding audio-video content has its own problems.

The problems are that *a*) it is expensive – human time is more expensive than machine time; *b*) it is a very slow process – unlikely to ever match the increasing scale of big data. We will illustrate the problems with a concrete question: *which speakers of language gesture the most?* To answer this question, the current common practice is to perform three tasks. First, *gesture the most* is defined as precisely as possible – is *gesture the most* with respect to gesture *size* or the *number* of gestures or both? Second, video recordings of gestures of speakers are made or collected for as many languages as possible. Third, the video recordings are annotated for gesture units; humans go through the video recordings frame by frame and mark carefully the start and end of gesture units for each speaker (and repeat the process for all speakers and languages). After all videos are annotated, a script is written to count and compare the number (or size) of gestures across groups of interest (e.g. languages, professions, cultures).

The above workflow with humans in the cycle is time-consuming. A one-hour video with 25 frames per second may take as long as 25 hours with the assumption that it takes a total of one second to watch, analyze and decide whether a given frame is part of a gesture unit. Marking the start and end of gesture units is not the hardest type of annotation; annotation can be much more complex and the more complex it is, the more time it takes to identify and annotate it.

To summarize, manual annotation takes orders of magnitude longer than the video length. For this reason, empirical research that relies on analysis of audio-video content has been limited in two ways. First, in a given time, only a small fraction of the audio-video data could be annotated and made available for research. Second, the creative mind of the researcher has been divided between doing research and doing manual annotation (or waiting for it to be completed by others).

Given the limitations of manual annotation, can we develop technologies to perform automatic video annotations for some applications?

This thesis answers *yes* by presenting innovative solutions to four gesture-related annotation problems: *1*) **speaker diarization** – the problem of determining *who spoke when*, *2*) **signer diarization** – the problem of determining *who signed when*, *3*) **sign language identification** – the problem of determining the identity of a sign language, *4*) **gesture stroke detection** – the problem of segmenting gestures into meaningful units. These four problems are studied in the realm of the AVATecH project[1], a joint effort of two Fraunhofer and two Max Planck Institutes. The objective of the project is to investigate and develop technologies for semi-automatic annotation of audio and video recordings.

1.2 Problem statement

How can a machine solve gesture-related problems?

Gestures are body, hand and facial movements, which humans use to communicate. Enabling machines to recognize them has applications in video analytics and human-computer interaction. This thesis studies gesture recognition with the objective of solving four important problems: speaker diarization, signer diarization, sign language identification and gesture stroke detection. The fundamental challenges of gesture recognition arise from two sources: *1*) where humans see gestures, a machine sees only time-varying pixels, and *2*) the time-varying gesture pixels occur in diverse environments. The two challenges give rise to the following research question.

Research question 1:

How can a machine recognize gestures in diverse environments?

Whatever the answer to this research question, it has a high chance of success if it involves a machine that can learn from examples. A machine that can learn from data can deal with diverse environments better than a machine that is preprogrammed (if preprogramming is possible at all). For this reason, this thesis takes machine learning as the key to the problems studied. In machine learning, a learning algorithm has to be trained with as many examples as available. The fewer examples needed, the better. But with fewer training examples, machine learning has a severe generalization problem. The more examples available, the better the generalization. But producing more examples, which is usually done by humans, is expensive and non-scalable. The fact that we want good generalization with small examples leads us to raise the following research question.

Research question 2:

How can a machine effectively use data to learn to recognize gestures?

[1] `https://tla.mpi.nl/projects_info/avatech/`

The answer to the second question has to balance two goals: achieve high recognition accuracy and use as few training examples as possible. This can be done by learning to adapt to new situations using small adaptation data.

1.3 Research approach

We study the four problems mentioned in the previous subsection (speaker diarization, signer diarization, sign language identification and gesture stroke detection) using a common research method that we detail as follows:

1. **Divide and conquer**: break each problem into many smaller subproblems
2. **Attack subproblems**: propose a solution to the subproblems
3. **Evaluate solutions**: evaluate solutions quantitatively and qualitatively

1.3.1 Divide and conquer

To solve each of the problems presented in this thesis, we take a *divide and conquer* approach. We divide the problems into several subproblems such that each subproblem can be solved independently (i.e. with very little coupling with the rest of the subproblems). To illustrate this, the following are the subproblems we came up with for speaker diarization:

1. How many people are there in the video?
2. How can we know where the people are in the video?
3. How can we determine if each person is gesturing at any given time?
4. How can we know which spoken utterance belongs to which person?

At first sight, these subproblems seem irrelevant to solving speaker diarization (after all, speaker diarization is about speech). But when we examine the hypothesis that *the gesturer is the speaker*, then we see that it is exactly those subproblems that we need to solve.

1.3.2 Attack subproblems

We attack the video processing subproblems using two strategies: *1*) we assume that one or more of the subproblems have been solved or will be solved by someone else, *2*) we design and develop a complete machine learning (ML) system that solves the subproblems not solved by the first strategy. For example, in *speaker diarization using gesture*, the subproblems of determining the number of speakers and where they are in the video are assumed to be determined or easily initialized by humans (e.g. human computation [Von Ahn, 2009]). But the subproblems of determining whether a person is gesturing and whether a particular spoken utterance belongs to that person are considered novel and are solved by the second strategy.

The heart of the second strategy is machine learning. In attacking problems using machine learning, three issues are important: data, features and learning algorithms. We outline our views of these issues as follows.

Data

Our input data is mainly video, but we also consider audio whenever it is relevant. A video is a time sequence of digital images, each of which is a sequence of quantized intensity values (pixels) taken at discrete points in 2D space. A complete understanding of the classes of objects in the video requires the analysis of the pixels of each frame, both by itself and in relation to the pixels in the neighboring frames. To go from pixels to semantics (i.e. to some high-level information), two types of challenges must be overcome: within-class variations and between-class similarities.

Within-class variations: Instances of the same class give rise to different pixel values. The variation could be natural or artificial. Natural variation refers to the variation of properties of objects of the same class. For example, many types of dogs exist even though they all belong to the same class of dogs. Artificial variation refers to the variation that result from recording conditions: view-point variation (the angle of view affects the appearance of the object), illumination changes (light intensity affects how objects appear), occlusions (partial parts of objects are hidden from view), scale (a video recorded from a close range is different from that recorded from a far range), background clutter (the object of interest could be found on a clutter as opposed to a clear background).

Between-class similarities: Instances of different classes share similar features. The similarity could be natural or artificial. Natural similarity refers to the similarity of properties of objects of different classes. For example, instances of a dog have common features with instances of a cat. Artificial similarity refers to the similarity that results from recording conditions. For example, illumination (e.g. dark) may make objects appear very similar even though the objects have different natural appearances.

The within-class variations and the between-class similarities also apply to classes of movements. For example, a gesture for "goodbye" and a gesture for "stop" have their own within-class variations both within individuals and across individuals but they also have common features (e.g. both gestures involve the raising of the hand palm out in front of the person).

To summarize, instances of the same class give rise to different pixel values and instances of different classes give rise to the same or similar pixel values.

Given that summary, how can a machine learn to distinguish instances of different classes?

First, we need to have many instances of data that cover the range of variations within each class. Second, we need to go beyond pixels and extract invariant features. What are features?

Features

Features are measurable properties of objects that are used for classification. The more informative the features, the better the classification accuracy. Which features are informative in our problems? We use different features depending on the problem. For gesture detection and gesture stroke segmentation, we use features extracted from interest-point and skin-color detectors. For speaker diarization, we use both video features (Motion History Images) and speech features (MFCC). For sign language identification, we use *a*) handcrafted features based on skin-color detection and *b*) features learned through unsupervised techniques.

Unsupervised feature learning techniques are machine learning techniques that learn a transformation function that converts raw inputs (e.g. pixels) to features that can be used in a supervised learning task [Coates *et al.*, 2011; Lee *et al.*, 2009]. Out of several feature learning algorithms available (e.g. autoencoders, clustering, dictionary learning, restricted Boltzmann machines), we implemented clustering (K-means) and sparse autoencoder algorithms.

Learning algorithms

The four problems addressed in the thesis require the prediction of a class label for *a*) each frame in an unsegmented video sequence (speaker diarization, signer diarization, gesture stroke detection) or *b*) all frames in the video (sign language identification). The former can be seen as a sequence labeling problem (classification at every time instant t) and the latter as a classification problem that treats the whole video as one entity with a single class label.

A number of machine learning algorithms and models exist to solve both types of problems. We list the ones considered and/or used in the thesis for either classification or feature learning: logistic regression, SVMs, random forest, K-means, Gaussian Mixture models, Hidden Markov models, conditional random fields, probabilistic Bayesian models and neural networks (deep learning). We also design our own deterministic algorithms based on heuristics, when applicable.

1.3.3 Evaluate solutions

We evaluate the performance of our solutions both quantitatively and qualitatively.

Our quantitative evaluations follow different strategies depending on the class label distribution and the type of problem. For speaker diarization, we report results in diarization error rate, which is a standard metric in the speaker diarization research community. For classification problems (sign language identification and gesture stroke detection), we report results in terms of different metrics: accuracy, precision, recall and Area Under the Curve (AUC) of Receiver Operating Characteristic (ROC).

Our qualitative evaluations concern one or more of the following: *a*) time and space complexity *b*) error analysis *c*) visualization of the most informative features. For example, for speaker diarization using gesture and speech, we emphasize how our solution offers advantages of efficiency over diarization techniques that are based on hierarchical agglomerative clustering. For sign language identification, we visualize the learned features and show how they are activated for each sign language. Visualization can help us to understand the learned features better.

1.4 Summary of contributions

This thesis has made contributions to four topics: speaker diarization, signer diarization, sign language identification and gesture stroke detection. We present the contributions in the order of their appearance in the thesis.

Chapter 2: Speaker diarization using gesture

[Gebre *et al.*, 2013b]

> Extensive literature exists on speaker diarization, the task of determining *who spoke when*. This study contributes to the literature by justifying and using gesture for speaker diarization. The use of gesture for speaker diarization is motivated by the observation that whenever people speak, they also gesture. This observation is the basis of the hypothesis: *the gesturer is the speaker*. To justify the hypothesis, this study presents evidence from the gesture literature. After the justification, the study moves on to the design and development of novel vision-based speaker diarization algorithms. Two algorithms are proposed: one based on corner detection/tracking and the other based on motion history images. The latter algorithm is presented in chapter 4.

Chapter 3: Signer diarization using gesture

[Gebre *et al.*, 2013a]

> Signer diarization, the task of determining *who signed when*, has similar motivations and applications as speaker diarization except for the difference in modality. While there is significant literature on speaker diarization, very little exists on signer diarization. This study contributes to the sign language processing literature by identifying signer diarization as an important problem and proposing a solution to it. Given the similarity between sign language and gesturing, the proposed solution is similar to the solution we proposed for *speaker diarization using gesture*.

Chapter 4: Online diarization using Motion History Images

[Gebre *et al.*, 2014c]

A Motion History Image (MHI) is an efficient representation of where and how motion occurred in a single static image. This study demonstrates the use of MHI as a likelihood measure in a probabilistic framework of detecting gestural activity. The study claims with experimental evidence that the efficiency of MHIs makes them usable in online speaker and signer diarization tasks as motion is an integral part of uttering activity.

Chapter 5: Speaker diarization using gesture and speech

[Gebre *et al.*, 2014b]

Speech and gesture can be combined to solve speaker diarization. This study contributes to the speaker diarization literature by approaching speaker diarization as a speaker identification problem after learning speaker models from speech samples co-occurring with gestures (the occurrence of gestures indicates the presence of speech and the location of the gestures indicates the identity of the speaker). This novel approach offers many advantages over other systems: better accuracy, faster computation and more flexibility (controlled trade-off between computation and accuracy). DER score improvements of up to 19% have been achieved over the state-of-the-art technique (the AMI system).

Chapter 6: Automatic sign language identification

[Gebre *et al.*, 2013c]

Extensive literature exists on language identification, but only for written and spoken languages. This work contributes to the literature by identifying sign language identification as an important language identification problem and proposing a solution to it. The solution is based on the hypothesis that sign languages have varying distributions of phonemes (hand shapes, locations and movements). Questions of how to encode and extract hand shapes, locations and movements from video are presented along with classification results on two sign languages, involving video clips of 19 different signers.

Chapter 7: Unsupervised learning for sign language identification

[Gebre *et al.*, 2014a]

What features are different between sign languages? This study contributes to the literature by presenting a sign language identification method based on features learned through unsupervised techniques. It shows how K-means and sparse autoencoder can be used to learn feature maps from videos of sign languages. Through convolution and pooling, it also shows the use of these feature maps for classifier feature extraction. Finally, the study shows the impact on accuracy of varying the number of feature maps with classification experiments on 6 sign languages involving 30 different signers. High accuracy scores are achieved (up to 84%).

Chapter 8: Gesture stroke detection

[Gebre *et al.*, 2012]

Gesture stroke detection is one of the main preprocessing tasks in gesture studies. The task can be likened to speech segmentation or word tokenization. This study contributes to the literature by proposing an adaptive user-controlled solution to gesture stroke detection. The study shows how visual features can be extracted from videos based on interaction with the user (for example, to detect skin colors). The study also considers the role of speech features in gesture stroke detection. Classification results are presented with visual features alone, speech features alone and both visual and speech features.

Summarizing, our main contribution to speaker diarization concerns a novel algorithm for solving an old problem, using a multimodal approach combining gesture and speech. Contributions to the other domains include the formulation, application, extension and implementation of state-of-the-art machine learning techniques, leading to improved adaptive algorithms, among others for sign language identification.

1.5 Structure of this thesis

This thesis is a thesis by publication. It consists of one introduction chapter, seven major chapters, and one conclusion chapter. The major chapters are written to reflect the seven papers that have been published as conference proceedings.

Chapter 2

Speaker diarization: the gesturer is the speaker

Content

This chapter presents a solution to the speaker diarization problem based on a novel hypothesis. The hypothesis is that the gesturer is the speaker and that identifying the gesturer can be taken as identifying the active speaker. After presenting evidence to support the hypothesis, the chapter presents a vision-only diarization algorithm with experimental evaluations on 8.9 hours of the AMI meeting video data.

Based on

B. G. Gebre, P. Wittenburg and T. Heskes (2013). “The gesturer is the speaker”. In *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3751-3755, IEEE.

Keywords

Speaker diarization, gesture, AMI dataset, diarization error rate, optical flow

2.1 Introduction

Speaker diarization is the task of determining *who spoke when* from an audio or video recording. It has applications in document structuring of meetings, news broadcasts, debates, movies and other recordings. Most of its applications come in the form of *speaker indexing* (used for video navigation and retrieval), *speaker model adaptation* (used for enhancing speaker recognition) and *speaker attributed speech-to-text transcription* (used for speech translation and message summarization).

The focus of application for speaker diarization has been shifting over the years. In the past, the focus was on telephone conversations and broadcast news [Rosenberg *et al.*, 2002; Tranter and Reynolds, 2004]. Currently, the focus is on conference meetings [Fiscus *et al.*, 2008; Anguera *et al.*, 2012]. The shift in focus (from telephone conversations to conference meetings) influenced the shift in the signals used in the speaker diarization algorithms: from using audio only [Tranter and Reynolds, 2006] towards using both audio and visual signals [Anguera *et al.*, 2012]. Our work is part of this shift and demonstrates how video signals alone can be used for speaker diarization.

The full attention given to video signals in solving speaker diarization is based on a novel hypothesis: *the gesturer is the speaker*. Our hypothesis arose from the observation that although a speaker may not be gesturing for the whole duration of speech, a gesturer is mostly speaking. Section 2.2 grounds the hypothesis in gesture–speech synchrony studies. Convinced by the evidence for gesture–speech synchrony, we claim *who gestured when* can be used to answer *who spoke when*. This claim leads to questions: *how do we detect gestures?* and *how do we know which person produced them?* In section 2.3, we give answers to these questions and present our proposed diarization algorithm. Our algorithm performs speaker diarization by first detecting optical flows and classifying them based on the location of the speakers in the video. How reliable is this algorithm?

Section 2.4 presents the AMI meeting data and the diarization error rate (DER) metric that we used to validate our algorithm. We used thirteen videos with each having at most four speakers. Section 2.5 discusses achieved results and compares qualitatively our diarization method with previous methods. Section 2.6 summarizes our study and makes suggestions for future work. Section 2.7 summarizes our study and makes suggestions for future work. Finally, section 2.7 presents related work to put context to our approach.

2.2 Gesture-speech relationship

People of any cultural and linguistic background gesture when they speak [Feyereisen and de Lannoy, 1991]. Speakers produce gestures to highlight concepts of length, size, shape, direction, distance and other concepts expressed in their speech. Listeners comprehend by integrating information from speech with information from gestures (of lips, eyes, hands, etc.) [McNeill, 1992a; Özyürek *et al.*, 2007]. What exactly is the relationship between gesture and speech?

Complete agreement does not exist on the exact interpretation of the relationship between gesture and speech. One hypothesis holds that gesture and speech are separate communication systems [Butterworth and Beattie, 1978; Butterworth and Hadar, 1989; Feyereisen and de Lannoy, 1991]. Another hypothesis holds that gesture and speech together form an integrated communication system for the single purpose of linguistic expression; it holds that gesture is linked to the structure, meaning, and timing of spoken language [Kendon, 1980; McNeill, 1985].

Despite differences in the interpretation of the degree of relationship between gesture and speech, both hypotheses agree on the existence of high correlation in the timing of speech and gesture executions (i.e. gesture and speech execution occur within milliseconds of one another) [Levelt *et al.*, 1985; Morrel-Samuels and Krauss, 1992]. The following are selected arguments that show the tight relationship between gesture and speech (for more and detailed arguments, see McNeill [1985]):

- Gestures occur mainly during speech
- Delayed Auditory Feedback (DAF) does not interrupt speech-gesture synchrony
- The congenitally blind also gesture
- Fluency affects gesturing

2.2.1 Gestures occur mainly during speech

Studies of people involved in conversations show that speakers gesture and listeners rarely gesture [McNeill, 1985; Campbell and Suzuki, 2006]. In approximately 100 hours of recording, thousands of gestures were observed for the speaker but only one for the listener [McNeill, 1985]. In a sample of narrations, about 90% of all gestures occurred during active speech [McNeill, 1985]. In a meeting of eight speakers, the occurrence of upper body movement with speech accounted for more than 80% of the total speaking time [Campbell and Suzuki, 2006].

2.2.2 DAF does not interrupt speech-gesture synchrony

Delayed Auditory Feedback (DAF) is the process of hearing one's own speech played over earphones after a short delay (typically, 0.25 seconds). DAF disturbs the flow of speech; it slows it down and subjects it to drawling and metatheses (the transposition of sounds in a word). If speech and gesture were independent, DAF should not affect gesture execution. But because they are not, gesture and speech remain in synchrony despite the interruptions caused by DAF [McNeill, 2005].

2.2.3 The congenitally blind also gesture

The congenitally blind, people who are blind from birth and so have never seen gestures, gesture as frequently as sighted people do [Iverson and Goldin-Meadow, 1997; Iverson *et al.*, 2000]. In Iverson and Goldin-Meadow [1997], four children who are blind from birth were tested in 3 discourse situations (narrative, reasoning, and spatial directions) and compared with groups of sighted and blindfolded sighted children. Their findings indicate that blind children produced gestures and the gestures they produced resembled those of sighted children in both form and content.

2.2.4 Fluency affects gesturing

The relationship between speech fluency and gesture is direct. The number of gestures increases as speech fluency increases and it decreases as speech fluency decreases. For example, stuttering – a speech disorder, characterised by syllable and sound repetitions and prolongations – is rarely accompanied by gesture. During the moment of stuttering, gesturing falls to rest and within milliseconds of resumption of speech fluency, gesturing rises again [Mayherry and Jaques, 2000].

In summary, the aforementioned studies show that speech and gesture are tightly linked, at least in the timing of their executions. This means that the presence of gesture is evidence for the presence of speech. But, how do we recognize gesture from videos and how can we use it to perform speaker diarization? The following section answers these questions.

2.3 Our diarization algorithm

To perform speaker diarization using gesture, three modules need to be designed to determine:

- the number of speakers
- the identity (or signature) of each speaker and
- whether or not each speaker gestured

Each module can be simple or complex depending on the content of the video and recording conditions. For example, if the video content has people appearing and disappearing unpredictably, then a complex model is needed to track speaker numbers and identities. However, because model complexity is neutral to the concept of *the gesturer is the speaker*, this work proposes a simple algorithm that detects and tracks gestures of people in conference meeting videos. Conference meetings usually have fixed number of participants and the participants usually stay in fixed locations. This enables us to fix the number of speakers from the first few video frames either manually or automatically [Dalal and Triggs, 2005]. The fixed locations (territories) of the speakers will serve as their signatures.

Given the (tracked) locations of the speakers, the remaining tasks are to define what a gesture is and to determine its occurrence from frame to frame for each speaker/location. Comparison of any frame with its previous immediate frame shows that there are movements. Any of these movements could either be part of a gesture or be noise. To determine which movements are gestures, we propose a deterministic algorithm using heuristics.

The deterministic algorithm defines gesture to be any movement that lasts longer than a fixed number of frames. Brief and isolated head or hand movements are excluded. The motivation for the exclusion is to remove noise and to avoid confusion between real gestures and the movements that people make for non-communicative reasons (for example, during change of position or orientation).

Our deterministic diarization algorithm is presented in 2.1. The algorithm takes in a video of speakers and returns time segments for which there is at least one person speaking. Initialization of the algorithm includes fixing the number of speakers and their locations at the beginning of the video. From line 3 through 9, the algorithm detects motions. Detecting motion is performed by corner tracking. Corners are unique pixels that can easily be computed and tracked [Tomasi and Shi, 1994].

Given the corner features, tracking is done with the pyramidal implementation of the Lucas-Kanade algorithms [Bouguet, 1999; Bradski, 2000]. The Lucas Kanade algorithm finds the displacement that minimizes the difference of given interest points from two frames in a sequence. It works based on three assumptions: *a*) *brightness constancy* - a point in a given image does not change in appearance as it moves from frame to frame, *b*) *temporal persistence* - the motion of a surface patch changes slowly in time, and *c*) *spatial coherence* - neighboring points in an image belong to the same surface, have similar motion, and project to nearby points on the image plane. These assumptions do not always hold but they are good approximations for, in our case, motion detection.

The tracking of the corners is done within a window of a specified size. A trade-off exists between the choice of the window size and the size of motion detected (aperture problem). A small window cannot capture large motions. A large window violates the spatial coherence assumption. The trade-off is solved by applying the Lucas-Kanade algorithm over a pyramid of images. A pyramid of images is a collection of down-sampled images [Adelson *et al.*, 1984] and, in our case, we use it to detect large motions.

For continuous tracking, the corners need to be present in all frames. But this is rarely the case given that human body motions are non-rigid. This means that the number of corners and their locations are not stable; corners may disappear. The solution is to re-detect corners when tracking fails.

Tracking corners until failure gives motion segments. These motion segments are at the level of corners but what we want are motion segments at the level of hands and face. The motion segments' orientations are binned into three histograms corresponding to motions of the left hand, the right hand and the head.

Algorithm 2.1 Perform speaker diarization using gesture

```
Require: video of people communicating
Ensure: speaker IDs and their times of speech
 1: Initialize the number of speakers
 2: Initialize the location of the speakers
 3: while next frame is available do
 4:     for each speaker do
 5:         //Determine if gesturing activity is observed
 6:         Detect and track corners using Lucas-Kanade algorithm
 7:         Keep only those that move $> x$ pixels in significant directions
 8:     end for
 9: end while
10: Join motions that come from the same locations (smoothing)
11: Remove motions with duration $< y$ frames
12: Join motions that come from the same locations(re-smoothing)
13: Classify motions based on their location
```

Because tracking sometimes fails, the tracks for each speaker will have discontinuities. Line 10 avoids these discontinuities by joining tracks that are not very apart from each other. After smoothing, very short and isolated tracks are removed in line 11. But because this removal introduces discontinuities, re-smoothing is reapplied in line 12. Finally, the resulting segments (or tracks) are the speaking times, which line 13 assigns to speakers based on their locations.

2.4 Experiments

2.4.1 Dataset

The dataset for our experiments comes from the Augmented Multi-Party Interaction (AMI) Corpus [Carletta *et al.*, 2006]. The AMI Meeting Corpus is a multi-modal dataset consisting of 100 hours of meeting recordings. For our experiments, we used a subset of the IDIAP meetings (IN10XX and IS1009x) totalling 8.9 video hours. The selected recordings have four participants engaged in a meeting. Each recording has a separate video for a centre, left and right view of the participants and a separate high resolution video for each participant's face. From these different recordings of the same meeting, we selected the left and right camera recordings, each of which has two speakers with visible hands. An example snapshot of a selected video (IN1016 AMI meeting) is given in figure 2.1. The left and right camera views of the meeting are concatenated.

Speaker diarization can be challenging, depending on the number of speaker and the amount of interaction. Table 2.1 gives details of the interaction of the participants in the selected videos. The details concern the length of videos (in minutes), speech-time percentage (speech-time over video length), speech overlap percentage (overlapped speech time over video length), and speaker turn switches (average number of speaker turn switches per minute).

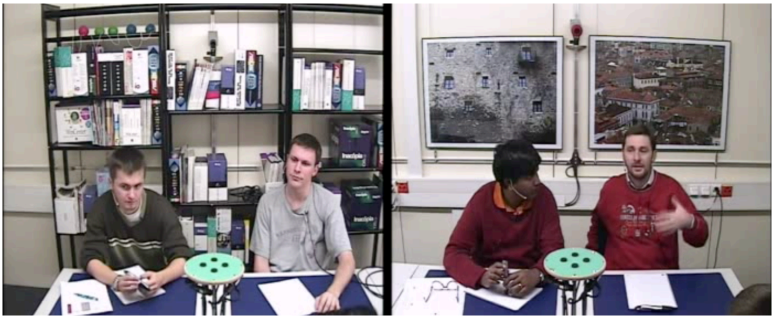


Figure 2.1: The figure represents the expected input to our algorithm. It is an example snapshot of AMI-IN1016 video data. Our algorithm will predict that the person on the right is speaking because, while other participants are motionless, he is gesturing.

Table 2.1: Features of experiment videos: speech-time percentage (speech-time over video length), speech overlap percentage (overlapped speech time over video length), and speaker turn switches (average number of speaker turn switches per minute).

Name	Video length (min)	Speech time (%)	Speech overlap (%)	Turn switches (per min)
IN1005	46	94.90	9.53	7.35
IN1016	59	96.95	18.27	12.30
IS1009b	34	87.88	8.97	6.48
IN1012	51	96.89	28.44	12.82
IN1002*	41	93.15	14.31	10.03
IN1007*	40	96.46	22.57	9.43
IS1009c	30	84.16	4.23	4.85
IN1013	51	96.04	26.64	12.88
IN1009	20	89.67	12.61	4.57
IN1014*	61	90.49	12.21	10.00
IN1008*	56	90.81	9.27	12.40
IS1009d*	32	80.83	8.58	8.45
IS1009a*	13	75.15	10.27	3.25

2.4.2 Evaluation metrics

Diarization Error Rate (DER) is widely used to evaluate speaker diarization systems. Despite its noisiness and sensitivity [Mirghafori and Wooters, 2006], DER is used by NIST[1] to compare different diarization systems. It consists of three types of errors: false alarms (i.e. the system predicted speech that is not in the reference), missed speech (the system failed to predict speech that is in the reference) and speaker error (speech that is attributed to the wrong speaker). Equation 2.1 shows that DER is measured as the fraction of time that is not attributed correctly to a speaker or to non-speech and figure 2.2 illustrates the same information graphically.

$$\text{DER} = \frac{\sum_{s\in S} dur(s)\Big(max\big(N_r(s), N_h(s)\big) - N_c(s)\Big)}{\sum_{s\in S} dur(s)N_r(s)}, \tag{2.1}$$

where
$dur(s)$ = the duration of segment s,
$N_r(s)$ = the # of reference speakers speaking in segment s,
$N_h(s)$ = the # of system speakers speaking in segment s,
$N_c(s)$ = the # of reference speakers speaking in segment s for whom their matching (mapped) system speakers are also speaking in segment s. A segment s is the time range where no reference or system speaker starts or stops speaking.

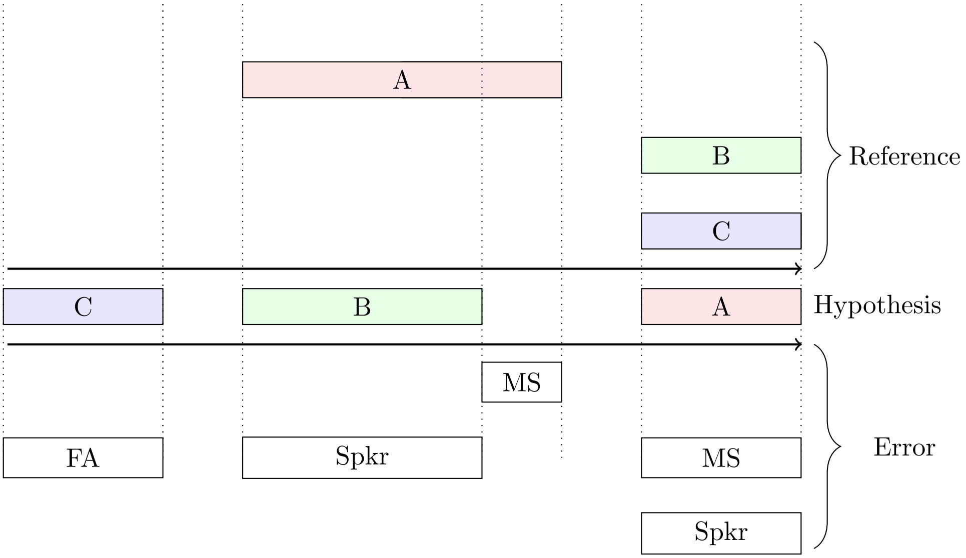


Figure 2.2: Illustration of elements of diarization error rate (DER): DER is the sum of the boxes in the error section. Whenever there is overlapped speech and the system does not predict it, it counts as missed speech and speaker error.

[1] http://www.itl.nist.gov/iad/mig//tests/rt/2006-spring/index.html

2.5 Results and discussion

The output of our diarization system is evaluated for correctness against manually annotated data in terms of Diarization Error Rate (DER). In speaker diarization calculations using DER, the reference segments are only those with speech (see equation 2.1). In our evaluations, the reference segments are those with gestures.

Recall that our diarization algorithm discards movements that are isolated and short. Figure 2.3 shows the impact on performance of this discarding for four videos (achieving the lowest DERs). As movements of short durations (from 0 to 65 frames) are discarded, DER decreases thereby increasing performance. To give a single DER estimate for each video, we considered movements of duration that lasted longer than 2.5 seconds (note that for ICSI-based speaker diarization systems, every speaker is assumed to be speaking for at least 2.5 seconds [Friedland *et al.*, 2012]). Based on the 2.5 seconds cut-off (63 frames) of movement duration, our DER scores for all tested videos are presented in table 2.2. The table also presents percentages for gesture-time, gesture-overlap and the number of gesturer turn switches per minute.

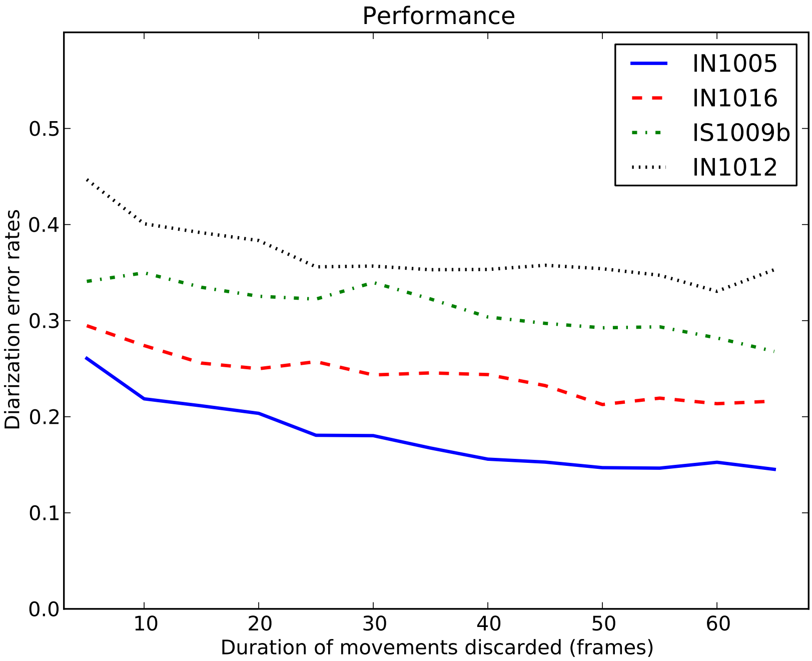


Figure 2.3: Discarding movements of short durations (< 65 frames) decreases DER whereas discarding movements of long durations (> 65 frames) increases DER. Frame rate is 25.

Table 2.2: Diarization Error Rates (DER) for 13 videos characterized by: the gesture-time percentage (gesture-time over video length), the gesture overlap percentage (overlapped gesture time over video length), and the number of gesturer turn switches (average number of gesturer turn switches per minute).

Name	Gesture time (%)	Gesture overlap (%)	Turn switches (per min)	DER (%)
IN1005	62.54	0.03	1.07	14.52
IN1016	72.45	0.00	1.58	21.62
IS1009b	72.23	0.00	0.78	26.80
IN1012	64.00	0.00	1.67	35.30
IN1002*	63.65	0.00	0.95	37.03
IN1007*	67.06	0.04	1.37	40.41
IS1009c	66.40	0.00	0.70	45.22
IN1013	69.47	0.01	1.42	53.73
IN1009	59.50	0.00	0.67	54.92
IN1014*	71.60	0.00	1.15	58.16
IN1008*	57.80	0.00	1.88	62.47
IS1009d*	68.82	0.00	0.58	63.05
IS1009a*	60.84	0.00	0.28	63.98

How do our results compare with previous results? Direct quantitative comparison would be incorrect given the differences in experimental set-up, set of videos used and the sensitivity of the DER [Mirghafori and Wooters, 2006]. But, for rough comparison, we mention previous NIST evaluation results. The official NIST Rich Transcription 2009 evaluation results for various conditions are presented in Friedland *et al.* [2012]. For batch audio, the DER ranges between 17.24% and 31.30%. For online audio, the DER is 39.27% and 44.61%. For audiovisual, it is 32.56%.

We can make qualitative comparison of our diarization method with previous diarization methods. Our diarization method has the advantage of being simpler and of using only video features (making it suitable for noisy environments). Previous speaker diarization systems are based on the ICSI speaker diarization system [Wooters and Huijbregts, 2008] and involve a number of subcomponents [Friedland *et al.*, 2012; Huijbregts *et al.*, 2012] for tasks such as filtering (Wiener), modeling (GMMs and HMMs), parameter estimation (Expectation-Maximization), decoding (HMM-Viterbi), clustering (agglomerative hierarchical clustering (AHC) with Bayesian information criterion (BIC)) and feature extraction (such as MFCC).

Our diarization method does not use any of these subcomponents but uses algorithms for corner detection [Tomasi and Shi, 1994] and tracking [Bouguet, 2001] under the assumption that upper bodies of stationary or tracked speakers are visible in the video. It is this assumption which limits the application of our diarization method. Where an active speaker becomes invisible in the videos (which is the case for video names marked with * in table 2.2), the diarization error becomes higher. Furthermore, in videos where the gestures of a person are picked up by both cameras, which is the case for most videos (because of the camera arrangements), the diarization error becomes higher. This can be seen in figure 2.1, where the head of the left-most person also appears in the bottom-right corner.

There are two criticisms of using gesture for speaker diarization. One is of the form: *speakers do not always gesture*. This is true but gesture is frequent enough that, in some cases, methods can be designed to overcome its absence (e.g. smoothing). In our videos, the diarization algorithm has found that roughly 75% of speech is accompanied by gesture. The other criticism is of the form: *what is a gesture?* This is hard to answer without reference to semantics. In our case, we assumed any movement to be part of a gesture and it seems that this is a reasonable assumption for people in conference meetings. For more complex scenarios, there is a need to differentiate gestural activity from other activities.

2.6 Conclusions and future work

This chapter presented a novel solution to the speaker diarization problem based on the hypothesis that *the gesturer is the speaker* and that gestural activity can be used to determine the active speaker. After giving evidence to support the hypothesis, the chapter presented an algorithm for gestural activity detection based on localization and tracking of corners. The algorithm works based on the assumption that the background of the speakers is static and that the speakers do not switch places. This assumption is reasonable for conference meetings. Further improvements of the algorithm for understanding gestures under more general recording conditions are left for future work. Future work should examine a probabilistic implementation of the diarization algorithm and include other cues including audio, lip movements and visual focus of attention of speakers (listeners tend to look at the active speaker).

2.7 Related work

The work presented here focuses on justifying and using gesture for speaker diarization. To the best of our knowledge, this has not been done before and is therefore a contribution. This work is similar to but more general than the work by Cristani *et al.* [2011], which considers using gesturing as a means to perform Voice Activity Detection (VAD). Their main rationale is different from ours. They see audio as the most natural and reliable channel for VAD. They use gesture when audio is unavailable (e.g. in surveillance conditions). By contrast, this work emphasizes that gesture is synchronous with speech, and wherever applicable, gesturer diarization can reliably be taken as speaker diarization.

The work presented here also includes the presentation of a new vision-based speaker diarization algorithm that is different from the standard ICSI speaker diarization system [Ajmera and Wooters, 2003; Wooters *et al.*, 2004; Wooters and Huijbregts, 2008]. The ICSI system is the most dominant diarization system with state-of-the-art results in several NIST RT evaluations[2]. The system is based on an agglomerative clustering technique. In the context of speaker diarization, this technique has three main stages: preprocessing, segmentation and clustering (see figure 2.4. The preprocessing is done once but the segmentation and clustering are done iteratively until 'optimal' number of clusters is obtained. The optimal number of clusters is meant to represent the actual number of speakers present in the recording.

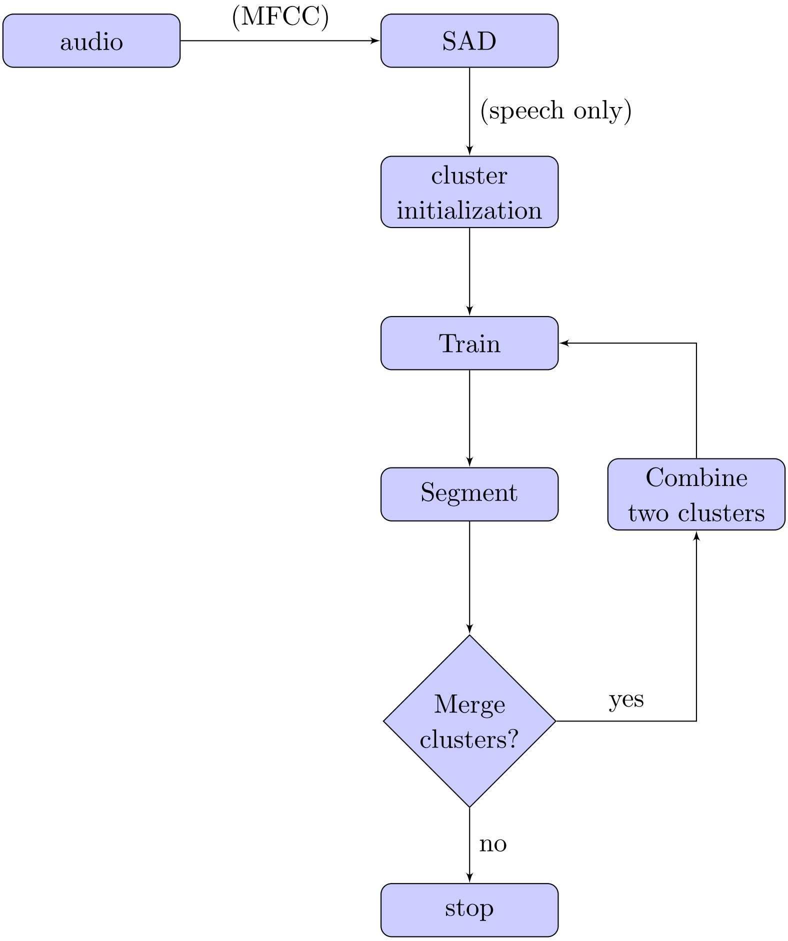


Figure 2.4: Overview of the ICSI speaker diarization system

[2] http://www.itl.nist.gov/iad/mig/tests/rt/

2.7.1 Preprocessing

The purpose of the preprocessing stage is to prepare the speech data. The preparation involves filtering (to reduce noise), speech activity detection (to remove silence parts and non-speech sounds) and feature extraction (to turn speech data into acoustic features such as MFCC, PLP, etc.). At this stage, cluster initialization is also performed – the initial number of clusters is fixed and speech segments are grouped together in the clusters. Systematic approaches to initialization can improve performance and system adaptability [Anguera *et al.*, 2006; Imseng and Friedland, 2009, 2010; Ben-Harush *et al.*, 2012]. The initialization process gives acoustic models in the form of GMMs for each cluster. These GMM models are then used to seed the segmentation process.

2.7.2 Segmentation and clustering

Speaker segmentation is the process of assigning speaker IDs to speech segments. It aims at splitting the speech stream into speaker homogeneous segments or equivalently into speaker turn changes. With the current estimates of the GMM models, Viterbi decoding segments the speech stream. The new segmentation is then used in the clustering stage.

Clustering, aka merging, is the process of identifying and grouping together same-speaker segments from anywhere in the speech stream. This process selects the closest pair of clusters (GMM models) and merges them (a new GMM model). The decision to merge two clusters is made on the basis of BIC scores. The BIC scores of all possible pairs of clusters are compared and the pair that results into the highest BIC score is combined into a new GMM. The segmenting and clustering stages then repeat until there are no remaining pairs that when merged lead to an improved BIC score.

The segmentation and clustering stages do not have tunable parameters but the preprocessing stage has quite a few: the type of speech activity detector (supervised or unsupervised, usually supervised), the initial number of clusters (K, usually chosen to be 16 or 40), the initial number of Gaussian components for the clusters (M, usually chosen to be 5), the type of initialization used to create the clusters (usually, k-means or uniform partitioning), and the set of acoustic features used to represent the signal (usually 19 MFCC features).

Other acoustic features including Linear frequency cepstral coefficients (LFCC), Perceptual Linear Predictive (PLP) and Linear Predictive Coding (LPC) are also used [Anguera, 2007]. And since recently, visual features are receiving more attention; they are being widely used in combination with audio features [Vajaria *et al.*, 2008; Friedland *et al.*, 2009; Hung and Ba, 2010; Garau and Bourlard, 2010; Noulas *et al.*, 2012]. But despite the recognition of their importance, visual features are usually given second level importance. They are rarely used alone for speaker diarization even though tight relationship is known to exist between speech and body gestures.

In summary, our work builds on and extends the speaker diarization literature on two fronts: *a*) emphasis on the use of gesture for speaker diarization, and *b*) a new vision-only diarization method that performs reasonably well with the advantage of being simpler. Both fronts offer opportunities for research in new directions as we will see in chapters 4 and 5.

Chapter 3

Signer diarization: the gesturer is the signer

Content

This chapter presents a vision-based method for signer diarization – the task of automatically determining *who signed when* in a video. This task has similar motivations and applications as speaker diarization but has received little attention in the literature. The chapter motivates the problem and proposes a method for solving it. The method is based on the hypothesis that signers make more movements than their addressees. Experiments on four videos (a total of 1.4 hours and each consisting of two signers) shows the applicability of the method. The best diarization error rate obtained is 0.16.

Based on

B. G. Gebre, P. Wittenburg and T. Heskes (2013). "Automatic signer diarization - the mover is the signer approach". In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 283-287, IEEE.

Keywords

Sign language, diarization, gesture, phonemes, unique features, DER

3.1 Introduction

Speaker diarization, as presented in the previous chapter, is the task of determining *who spoke when* in an audio and/or video recording. It is a dedicated domain of research in the multimedia signal processing community, receiving many publications every year [Tranter and Reynolds, 2006; Anguera *et al.*, 2012]. Most applications and technologies of diarization are driven by spoken languages. But spoken language is only one of the modalities of human communication. Written and signed languages are the other common modalities. In this study, we consider the visual-gestural modality and provide a baseline algorithm for determining *who signed when* from a video recording of multiple signers engaged in a dialogue.

We call the task of determining *who signed when* a signer diarization problem. This task is similar to the problem of speaker diarization. In the previous chapter, we proposed a speaker diarization algorithm based on gestures. In this chapter, we propose to use the same algorithm to solve signer diarization as signed languages inherently involve gestures. Our hypothesis in the previous chapter is that *the gesturer is the speaker*. In this chapter, we update that hypothesis to: *the gesturer is the signer* as we are dealing with signed languages.

Compared to the previous chapter, the contribution in this chapter is the identification of signer diarization as an important problem and showing that the speaker diarization algorithm that we proposed in the previous chapter is also applicable to signer diarization. In section 3.2, we provide motivations and applications of signer diarization. In section 3.4, we present the signer diarization algorithm. The algorithm uses no more knowledge than signers' movements. In subsequent sections, we discuss the achieved results and give suggestions for future work.

3.2 Motivation

Determining the number of signers and *who signed when* from a video recording of unknown content and unknown signers has a number of applications in different domains that involve sign languages. These include broadcast news, debates, shows, meetings and interviews. The general applications come in the following forms.

Pre-processing module

> Signer diarization output can be used as input for single signer-based sign language processing algorithms such as signer tracking, signer identification and signer verification algorithms. It can also be used to adapt automatic sign language recognition towards a given signer. Currently, signer-dependent sign language recognition systems outperform signer-independent systems [Bauer *et al.*, 2000; Zhang *et al.*, 2004; Zieren and Kraiss, 2005; Cooper *et al.*, 2012b; Akram *et al.*, 2012]. In this context, automatic signer diarization systems can be used as input to signer adaptation methods.

Signer indexing and rich transcription

Indexing video and the linguistic transcripts by signers makes information search and processing more efficient for both humans and machines. Typical uses of such output might be for message summarization, machine translation and linguistic and behavioral analysis (for example, scientific turn-taking studies [Stivers *et al.*, 2009; Coates and Sutton-Spence, 2001]).

The need for some of the applications mentioned above might not be urgent given that sign language recognition is at research stage [Cooper *et al.*, 2012b], but in scientific turn-taking studies [Stivers *et al.*, 2009], humans already perform manual signer diarization. And, like all manual annotations, this process has limitations - it is slow, costly and does not scale with the increasing amount of data. Therefore, there is a need to develop methods for automatic signer diarization.

3.3 Signer diarization complexity

Given a video of signers recorded using a single camera, automatically determining *who signed when* is challenging. The challenge arises from signers themselves and the environment (recording conditions).

Signers

To begin with, the number of signers is unknown and this number may change in time as a participant leaves or joins the conversation. The locations and orientations of signers may change and these changes could take place while signing. Signers may take short signing turns and often sign at the same time (overlap in time). The signing spaces of signers may also be shared (overlap in space).

Environment

The environment includes background and camera noises. The background video may include dynamic objects – increasing the ambiguity of signing activity. The properties and configurations of the camera induce variations of scale, translation, rotation, view, occlusion, etc. These variations coupled with lighting conditions may introduce noises. These are common challenges in many other computer vision problems.

3.4 Our signer diarization algorithm

Sign language is a gestural-visual language. A signer produces a sequence of signs and an interlocutor sees and interprets the sequence. Like spoken languages, sign languages can be described at different levels of analysis such as phonology, morphology, syntax and semantics [Valli and Lucas, 2001]. The phonemes, which are the basic units of sign languages, are made from a set of hand shapes, locations and movements [Stokoe, 2005]. These subunits make up the manual signs of a given sign language. The whole message of an utterance is contained not only in manual signs but also in non-manual signs (facial expressions, head/shoulder motion and body posture) [Liddell, 1978].

In theory, an automatic signer diarization system can exploit some or all of the basic units from both manual and non-manual signs to perform signer diarization. In practice, however, some sub-units are easier to extract and exploit by the machine. This paper proposes a diarization method based on body movements. The hypothesis is that the active signer makes more movements than the other interlocutors.

3.4.1 Algorithm

Our automatic signer diarization algorithm consists of modules that determine: *a*) the number of signers, *b*) their identities (or signatures), and *c*) whether or not they signed. The modules can be simple or complex depending on the content of the video and/or recording conditions. The most general signer diarization system assumes nothing of the number of signers, their signatures and the video recording conditions. Developing such a method, besides being more complex, will be inefficient and is likely to even be less accurate than a system developed and tailored for a specific instance of video recording conditions.

In our diarization system, we make a number of simplifying assumptions about the video recording conditions and provide a mechanism for user involvement using annotation tools like ELAN [Sloetjes and Wittenburg, 2008]. The user of the system makes some simple decisions to initialize the system. The user determines the number of signers from the first frame of the video by creating bounded boxes for each signer. These bounded boxes limit the boundaries of the signing spaces for each signer. The diarization system assumes the signers maintain their location (this is a reasonable assumption for videos of interviews and conference meetings) or are tracked [Darrell *et al.*, 2000]. Given the locations of signers and assured of their stability, the remaining task is to define and determine signing activity detection for each signer/location from frame to frame.

What constitutes signing activity? Based on any consecutive frame pairs, each bounded box (i.e. a signer) may have some movement (arising either from signing activity or noise). Movements that last longer than a fixed number of frames are considered to constitute a signing activity. In other words, isolated and brief head or hand movements are excluded. The motivation for the exclusion of isolated and brief movements is to remove noise and to avoid confusion between real signs and other movements like moving the body to change orientation.

3.4.2 Implementation

What is a hand/face and what is a movement from an implementation perspective? We use corners to detect and track body movements. Corners have the property that they are distinct from their surrounding points, making them good features for tracking [Tomasi and Shi, 1994]. For a given point in a homogeneous image, it is not possible to identify whether or not it has moved in the subsequent frame. Similarly, for a given point along an edge, it is not possible to identify whether or not it has moved along that edge. However, the motion of a corner can conveniently be computed and identified [Tomasi and Shi, 1994].

For a given application, not all corners in a video are equally important. For sign activity detection, the interesting corners are the ones resulting from body movements, mainly from head and hand movements. In order to filter out the corners irrelevant to body movements, we ignore corners that do not move more than a given threshold. For tracking the movement of corners, we apply the pyramidal implementation of the Lucas-Kanade algorithm [Bouguet, 2001; Bradski, 2000].

The following is a pseudo-code for determining the active signer. For detailed description of the algorithm, see the explanation in the previous chapter (2.3).

Algorithm 3.1 Perform signer diarization using movement

```
Require: video of people communicating
Ensure: signer IDs and their times of signing
Initialize the number of signers
Initialize the location of the signers
while next frame is available do
    for each signer do
        //Determine if signing activity is observed
        Detect and track corners using Lucas-Kanade algorithm
        Keep only those that move > $x$ pixels in significant directions
    end for
end while
Join motions that come from the same locations (smoothing)
Remove motions with duration < $Y$ frames
Join motions that come from the same locations (re-smoothing)
Classify motions based on their location
```

3.5 Experiments

3.5.1 Datasets

We ran our signer diarization algorithm on four videos taken from the Language Archive at the Max Planck Institute for Psycholinguistics. Each video has two signers of Kata Kolok [de Vos, 2012] for the whole length of the video but sometimes a child or a passerby appears in the video. Table 3.1 shows the details of the interaction of the signers in the videos. The details are extracted from manually annotated data.

Table 3.1: Experiment dataset details: four videos each with two signers signing in Kata Kolok [de Vos, 2012]

Video	Length	STP	STM	DSS	SO
KN5	≈17	82.89	16.30	62.56	9.61
PiKe	≈18	70.04	15.40	57.62	11.52
ReKe	≈24	81.82	19.10	58.13	9.22
SuJu	≈24	78.13	15.24	66.39	9.68

Length = Video length in minutes
STP = Signing Time Percentage
STM = # of Signing Turns per Minute
DSS = Dominant Signer Share of sign time
SO = % of Signers Overlap (over sign time)

3.5.2 Evaluation metrics

We propose to use Diarization Error Rate (DER) to evaluate signer diarization algorithms. This evaluation metric, which we presented in the previous chapter, is widely used to evaluate speaker diarization systems despite the observation that it can be noisy and sensitive [Mirghafori and Wooters, 2006]. Equation 3.1 is the same formula that we use in the previous chapter to compute DER. In this chapter, we use the same formula but redefine it to give it a new meaning to reflect the fact that we are dealing with signed languages. Accordingly, it is defined as the fraction of signer time that is incorrectly attributed to a signer as shown in equation 3.1.

$$\text{DER} = \frac{\sum_{s\in S} dur(s)\Big(max\big(N_r(s), N_h(s)\big) - N_c(s)\Big)}{\sum_{s\in S} dur(s)N_r(s)}, \tag{3.1}$$

where
$dur(s)$ = the duration of segment s,
$N_r(s)$ = the # of reference signers signing in segment s,
$N_h(s)$ = the # of system signers signing in segment s,
$N_c(s)$ = the # of reference signers signing in segment s for whom their matching (mapped) system signers are also signing in segment s. Note that a segment s is the time range where no reference signer or system signer starts signing or stops signing. Qualitatively speaking, diarization error rate consists of three types of errors: false alarm signer time fraction (i.e. the system predicted signing time that is not in

the reference), missed signer time fraction (the system failed to predict signing time that is in the reference) and signer error time fraction (signer time that is attributed to the wrong signer).

3.6 Results and discussion

The output of our diarization system is evaluated for correctness against manually annotated data using Diarization Error Rate (DER). The reference frames are those frames that have been annotated (70-80% of the video length as shown in table 3.1). Table 3.2 presents the diarization error rate scores for each video. The best DER scores are obtained for SuJu, KN5 and ReKe videos. The worst DER is obtained for PiKe video. The explanation for the latter result has to do with false alarm errors (movements that are detected by the algorithm but that are not annotated as signs in the manually annotated data). Examining the video shows the sources of the false alarms. One source is the movement of a child that comes to her mother for part of the video. The other source is the appearance of signing activity of one signer in the signing space of the other signer.

Table 3.2: Signer diarization evaluation: diarization error rate scores.

Video	Y	**MS**	**FA**	**SE**	**DER**
KN5	13	0.12	0.07	0.05	0.24
PiKe	8	0.11	0.14	0.04	0.29
ReKe	18	0.14	0.05	0.05	0.25
SuJu	10	0.08	0.05	0.03	**0.16**

Y = Minimum signing duration (frames)
MS = fraction of Missed Sign Time
FA = fraction of False Alarm
SE = fraction of Wrong Signer Prediction
DER = **MS** + **FA** + **SE**

From the experiment data statistics and the DER scores, we can make the following observation: the diarization error rate is lower when one signer dominates more and when there is less overlap. For example, the best DER score of 0.16 is achieved for video SuJu, which has the most dominant signer and low signing overlap percentages (66.39% and 9.68%, respectively) and the worst DER score is achieved for PiKe, which has the highest signing overlap percentage (11.52%).

An important parameter of the signer diarization algorithm is the number of frames to remove – parameter Y shown in line 11 of the diarization algorithm (3.1). This parameter controls the minimum duration of body movements to consider as signing activity. It is measured in frames and any motion less than Y is considered noise and discarded. Figure 3.1 shows the impact of varying this parameter on

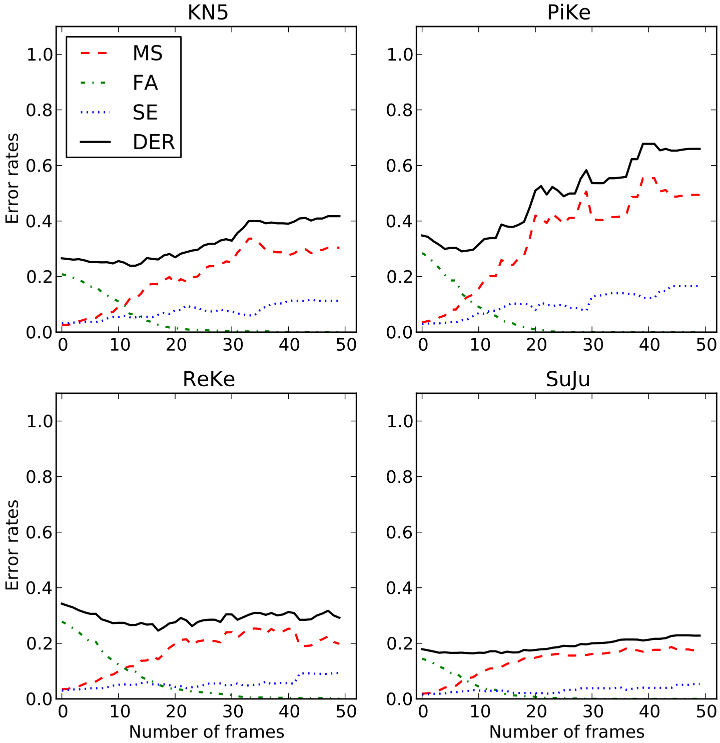


Figure 3.1: Performance variation as body movements of short duration are discarded.

diarization error rates for the four videos. The larger the Y value, the higher the missed signs and the lower false alarms (and vice versa). In other words, the Y value controls the trade-off between false alarms and missed signs. The best Y values that result in the lowest diarization errors are indicated in table 3.2.

Apart from the duration of the movements, our diarization algorithm does not interpret the movements. This makes it applicable independent of sign languages/signers but it also makes it vulnerable to false alarms. But, as our results indicate, movement is one of the most informative indicators of signing activity or uttering activity in general. Movements that speakers make, called gestures, are also used to identify speakers as we showed in the previous chapter.

In standard speaker diarization algorithms, which are based on iterative segmentation and clustering [Wooters and Huijbregts, 2008; Huijbregts *et al.*, 2012], each speaker is modeled by a Gaussian Mixture model (GMM). In our model, each signer is represented by a location. If the location is shared, which is not unlikely, a more powerful model of disambiguating the sources of signing activity is needed.

3.7 Conclusions and future work

This chapter introduced and motivated the signer diarization problem by drawing similarities with the speaker diarization problem. The chapter proposed a signer diarization algorithm based on the hypothesis that signers make more body movements than their interlocutors. The algorithm is implemented using corner detection and tracking algorithms. With a best score of 0.16 DER, our experimental results show the applicability of the algorithm in semi-automatic video annotations. From the results, we can formulate two conclusions. First, body motion is an inexpensive source of information for signer diarization - making it applicable regardless of sign languages and signers. Second, not all body motion is signing activity - making it less effective in noisy environments.

Future study should examine other sources of information than just body motion. Other sources include body posture, head orientations (interlocutors look at the active signer) and audio (signers sometimes make sound while signing). These different sources of information can then be fused in a probabilistic framework to perform signer diarization. In the next chapter, we present a probabilistic diarization algorithm based on a Motion History Image and show its application for online signer and speaker diarization. Note that our study in the previous two chapters focused on off-line speaker/signer diarization.

Chapter 4

Motion History Images for online diarization

Content

The previous two chapters presented a solution to the problems of offline speaker and signer diarization. This chapter presents a solution to the problem of online speaker and signer diarization. The solution is based on the idea that gestural activity is highly correlated with uttering activity; the correlation is necessarily true for sign languages and mostly true for spoken languages. The novel part of our solution is the use of motion history images (MHI) as a likelihood measure for probabilistically detecting gesturing activities and, because of its efficiency, using it to perform online speaker and signer diarization.

Based on

B. G. Gebre, P. Wittenburg, T. Heskes and S. Drude (2014). “Motion history images for online speaker/signer diarization”. In *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1537-1541, IEEE.

Keywords

Motion History Images, Motion Energy Images, gesture, AMI dataset

4.1 Introduction

Conversation can take place in written, spoken and signed languages. In any of these modalities, determining *who said when* is a challenging problem. In written works (e.g. fiction), tracking the number of characters and their utterances is hard because of, for example, anaphora resolution [Mitkov, 2002]. In spoken languages, determining *who spoke when* has also proven hard despite the research dedicated to it [Anguera *et al.*, 2012]. In signed languages, even though there is little research into it, our study presented in chapter 3 shows that it is also a hard problem because of non-communicative body movements.

In this chapter, we propose a solution to the problems of both speaker and signer diarization in online settings. Our work in chapters 2 and 3 focused on offline diarization, where the whole data is assumed to be available before diarization. In this chapter, we consider the problem where diarization has to be performed as soon as a segment of data arrives. We are interested in online diarization because it has applications in human-to-human or human-to-computer interactions (e.g. dialogue systems). For example, in video conferences, we would like to focus automatically on the active speaker. In human-robot interaction, we would like the robot to turn its head to look at the person speaking. Online diarization systems can also be used where offline diarization systems are used. For example, in information retrieval, we would like to index and search information by speakers/signers.

The aforementioned applications and others have led to extensive research into speaker diarization, resulting into many types of solutions and tools [Anguera *et al.*, 2012]. Most of these solutions focus on offline tasks [Tranter and Reynolds, 2006; Anguera *et al.*, 2012; Meignier and Merlin, 2010; Vijayasenan and Valente, 2012; Rouvier *et al.*, 2013]. A few of them focus on online tasks [Noulas and Krose, 2007; Markov and Nakamura, 2007; Friedland and Vinyals, 2008; Vaquero *et al.*, 2010]. Compared to previous work, the novel part of our solution is the application of Motion History Images [Davis and Bobick, 1997] in solving both speaker and signer diarization problems.

Our use of Motion History Images is presented in the context of online diarization tasks although it can also be used for offline diarization tasks. Motion History Image (MHI) is an efficient way of representing arbitrary movements (coming from many frames) in a single static image. This type of representation has been used for various action recognition tasks [Davis and Bobick, 1997; Bradski and Davis, 2002; Ahad, 2013]. The strength of MHI is its descriptiveness and real-time representation. It is descriptive because it can tell us where and how motions occurred. It is real-time because its computational cost is minimal. The rest of the chapter gives more details about MHI and its application in speaker/signer diarization.

4.2 Gesture representation

When people speak, they mostly gesture. When people use sign language, they inherently make movements. In either case, our goal in a diarization system is to determine where motion occurs and to decide if it indicates an uttering activity. Our work assumes that there is only body motion in the video. Motions that result from the camera or distracting objects are assumed to have been separated in a preprocessing step. For conference or meeting data, there is no need for a preprocessing step; we can safely assume that motions come mainly from humans engaged in a conversation. In such cases, how can we detect the foreground motion? We can either apply background subtraction or frame differencing. In our experiments, we applied frame differencing because we obtained results that are qualitatively similar to those coming from a less efficient background subtraction algorithm that uses a Gaussian Mixture Model [KaewTraKulPong and Bowden, 2002].

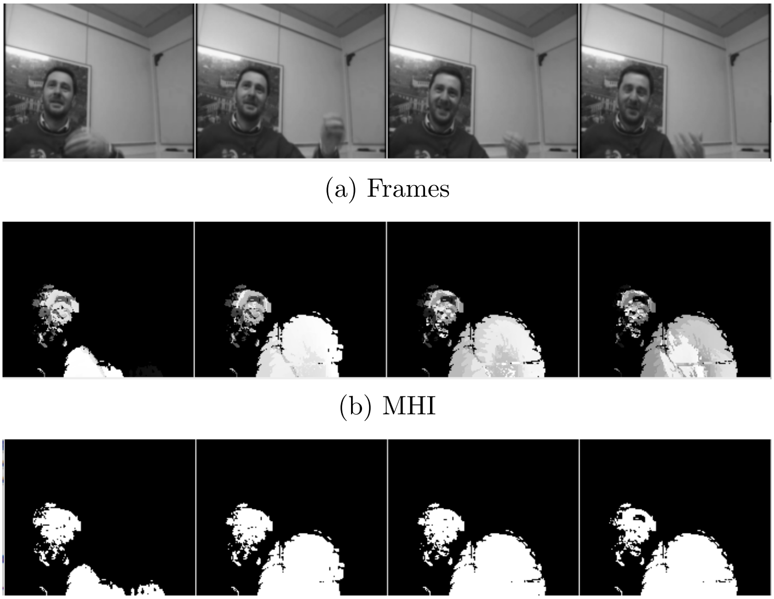
After finding the foreground (moving) objects, how do we efficiently and conveniently represent motion in a way that indicates *a*) where it occurred (space) *b*) when it occurred (time). We use Motion History Image (MHI) [Davis and Bobick, 1997]. A MHI is a single stacked image that encodes motion that occurred between every frame pair for the last τ number of frames. The type of information encoded in the MHI can be binary and, in such a case, it is called Motion Energy Image (MEI). The MEI indicates where the motion has occurred in any of the τ frames. We use this MEI to tell us which person is speaking or signing. MEI does not tell us how the motion occurred. For this information, we need to use the Motion History Image (MHI), which is an image whose intensities are a function of recency of motion. The more recent a motion is, the higher its intensity. More formal definitions of MEI and MHI are given in the following subsections.

4.2.1 Motion Energy Image

To represent where motion occurred, we form a Motion Energy Image and it is constructed as follows. Let $I(x, y, t)$ be an image sequence, and let $D(x, y, t)$ be a binary image sequence indicating regions of motion (for example, generated by frame differencing). Then the binary MEI $E(x, y, t)$ is defined as follows:

$$E_{\delta}(x, y, t) = \bigcup_{i=0}^{\delta-1} D(x, y, t-i), \tag{4.1}$$

where δ is the temporal extent of motion (for example, a fixed number of frames). In words, $E_{\delta}(x, y, t)$ is a single image that is the union of several binary images. The number of binary images depends on the parameter δ. Figure 4.1 (c) shows an image example of a MEI for a speaker who is also gesturing with δ set to 1 second.



(a) Frames

(b) MHI

(c) MEI

Figure 4.1: Examples of visualizations of MHI and MEI images. (a) Selected frames of a video taken from AMI meeting data. (b) The MHI of 25 frames - recent motions are brighter. (c) The MEI of 25 frames - white regions correspond to motion that occurred in any pixel in any one of the last 25 frames.

4.2.2 Motion History Image

To represent how motion occurred, we form a Motion History Image (MHI) as follows:

$$H_\tau(x,y,t) = \begin{cases} \tau & \text{if } D(x,y,t) = 1 \\ 0 & \text{else if } H_\tau(x,y,t) < (\tau - \delta) \end{cases} \tag{4.2}$$

where τ is the current time-stamp and δ is the maximum time duration constant (τ and δ are converted to frame numbers based on frame rate). In words, $H_\tau(x,y,t)$ is an image where current motions are updated to the current timestamp (basically, high values) whereas motions that occurred a little earlier keep their old timestamps (which are smaller than the current timestamp). Motions that are older than δ time are set to zero. Figure 4.1 (b) shows an example of MHIs at four different time instants for a speaker who is gesturing. Note that by thresholding a MHI above zero, a MEI image can be generated.

4.3 The online diarization system

In an online diarization system, we want to determine who at any time is speaking/signing given that we have video observations from 0 to t. Let each person's state be represented by x_t^i (binary values of speaking or not speaking) and let $z_{0:t}^i$ be measurements (of the video frames) for each person i, the objective is then to calculate the probability of x_t^i at time t given the observations $z_{0:t}^i$ up to time t:

$$p(x_t^i|z_{0:t}^i) = \frac{p(z_t^i|x_t^i)p(x_t^i|z_{0:t-1}^i)}{p(z_t^i|z_{0:t-1}^i)}, \tag{4.3}$$

where $p(z_t^i|z_{0:t-1}^i)$ is a normalization constant. In equation 4.3, there are two important probability distributions: one is $p(x_t^i|z_{0:t-1}^i)$, we refer to it as the conversation dynamics model and the other is $p(z_t^i|x_t^i)$ and we refer to it as the gesture model.

4.3.1 Conversation dynamics

Conversation imposes its own dynamics on speakers. A given speaker is more likely to continue to speak in the next frame than stop or be interrupted by others. We encode this type of dynamics as follows:

$$p(x_t^i|z_{0:t-1}^i) = \sum_{x_{t-1}} p(x_t^i|x_{t-1}^i)p(x_{t-1}^i|z_{0:t-1}^i) \tag{4.4}$$

where $p(x_{t-1}^i|z_{0:t-1}^i)$ is the posterior from the previous time and $p(x_t^i|x_{t-1}^i)$ is the conversation dynamics. The dynamics can be learned from training data but, for simplicity, we assume that a speaker is 90% more likely to continue speaking than not. Similarly, a silent person is more likely to continue to be silent. We encode these assumptions in a fixed transition matrix as follows:

$$p(x_t^i|x_{t-1}^i) = \begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix} \tag{4.5}$$

4.3.2 Gesture model

For both speaker and signer diarization systems, we assume that MEI is a strong indicator of an utterance. The higher the energy (the sum of MEI individual values), the higher the probability of an utterance. We model this type of relationship using a gamma distribution with shape parameter $\boldsymbol{k}$ and scale parameter $\boldsymbol{\theta}$.

$$p(z_t^i|x_t^i;\boldsymbol{k},\boldsymbol{\theta}) = \frac{(z_t^i)^{k_x-1}\exp(-\frac{z_t^i}{\theta_x})}{\theta_x^{k_x}\Gamma(k_x)} \quad \text{for } z_t^i,\boldsymbol{k},\boldsymbol{\theta} > 0 \tag{4.6}$$

where $x = x_t^i$, z_t^i is the number of motion pixels in a MEI for speaker/signer i and x_t^i is a binary random variable whose values represent uttering and non-uttering status of each person. Each state of x_t^i has its own gamma distribution whose parameter values are learned from data that has been manually annotated for speaking and non-speaking (similarly, for signing and non-signing). The models for gesture and the conversation dynamics are illustrated in figure 4.2.

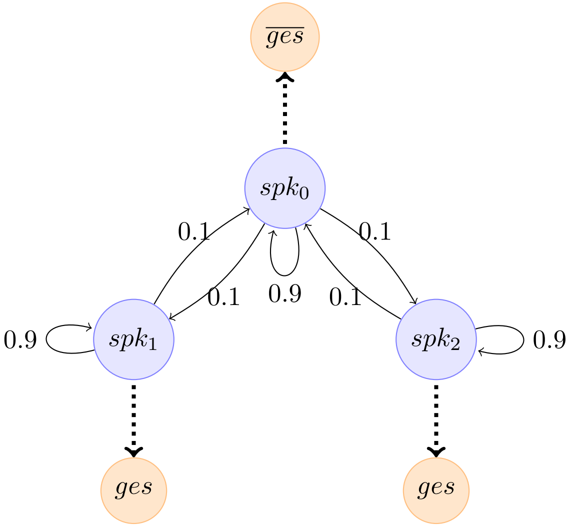


Figure 4.2: A state transition diagram for two speakers (spk_1 and spk_2) and one dummy speaker (spk_0), which represents silence or non-speech. Each speaker is checked for gesturing using the same gesture models (ges and $\overline{ges}$). The speaker that has the highest probability of speaking given observed gestures and the conversation dynamics is predicted to be the active speaker.

4.4 Experiments

4.4.1 Datasets

Spoken language data

Our spoken language experiment data comes from a publicly available corpus called the AMI corpus [Carletta *et al.*, 2006]. The AMI corpus consists of annotated audio-visual data of a number of participants engaged in a meeting. We selected seven meetings (IN10XX and IS1009), which together run for a total of ($\approx$ 4.9) hours. These meetings have four participants and are a subset of the meetings we used in chapter 2. The video recordings we used in chapter 2 were made by two cameras (left and right cameras). In this chapter, we use the video recordings that were made by four cameras, each recording the upper body of one participant. These individual recordings are mostly good but sometimes the hands of a participant are off-screen.

Sign language data

Our signed language experiment data consists of four video recordings (≈ 1.4 hours) of Kata Kolok, a sign language used in northern Bali [de Vos, 2012]. Each video has two participants conversing in sign language and is recorded from a single fixed camera. In these videos, there is no boundary between the signers. In fact, sometimes, the signing space is shared by both signers - making the task of diarization more difficult. Note that these videos are the same videos used in chapter 3 and for more details about the videos, see 3.5.1.

Where is each signer in the video? We answered this question by clustering MEI motion pixels into a prefixed K centers, set equal to the number of signers. We implemented a sequential k-means that updates the centers of clusters (signing space) in an online fashion as follows:

$$\boldsymbol{C}_t^i = \boldsymbol{C}_t^i + \frac{1}{n_{0:t}^i}(\boldsymbol{P}_t^j - \boldsymbol{C}_t^i) \tag{4.7}$$

$\forall j$ with $\boldsymbol{C}_t^i$ closest to P_t^j. $\boldsymbol{C}_t^i$ is the x-y center point for signer i at time t and $n_{0:t}^i$ is the total count of x-y points for signer i for times $0:t$. $\boldsymbol{P}_t$ refers to a location with non-zero value of MEI at time t and P_t^j stands for a point closest to C_t^i.

4.4.2 Evaluation metrics

We use Diarization Error Rate (DER) to evaluate our online diarization systems. This is the same evaluation metric that we presented and described in chapters 2 and 3. It consists of three types of errors: false alarm, missed speaker/signer time and speaker/signer error (see 2.4.2 and 3.5.2).

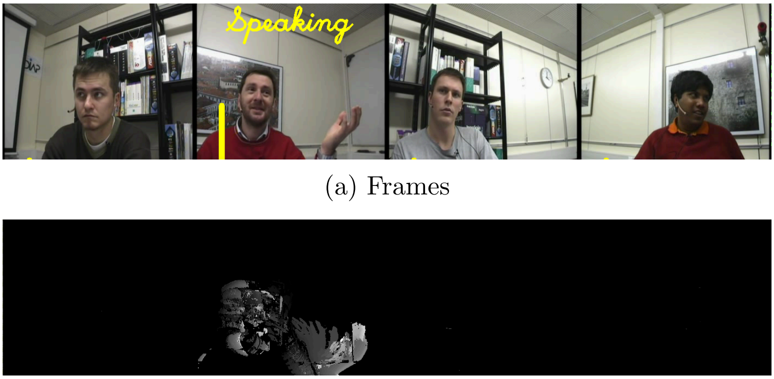
4.5 Results and discussion

4.5.1 Speaker diarization

The output of our speaker diarization system is given by probability values - one for each person per frame. We say that a person is speaking when the probability value for that person is the largest. The assumption is that at any time frame, only one person is speaking (unless more than one person has the same largest probability). Figure 4.3 shows a snapshot example of the output of the diarization system after running it on IN1016-AMI meeting data. In this figure, we can clearly see that the person that is gesturing is the speaker and the MHI clearly reflects this observation. But is that always the case? Table 4.1 shows that a person could be moving without speaking or that they could be speaking without gesturing. For this reason, the DER score is high for a baseline diarization algorithm that predicts the presence of speech whenever it detects motion.

Table 4.1: The proportion of time there is (no) motion when there is speech or no speech.

Speech?	**Motion?**	**Overlap**
Yes	Yes	0.98
	No	0.02
No	Yes	0.77
	No	0.23
Baseline diarization error rate (DER) = 196.75		
Motion for each speaker is defined as sum(MEI) > 0		



(a) Frames

(b) MHI

Figure 4.3: Output of the online diarizer on IN1016 meeting video. (a) Frames of speakers - the predicted active speaker is marked. The vertical bar shows the relative confidence in the prediction of *who is speaking?* (b) The MHI of the active speaker.

Table 4.2 gives performance scores of the diarization system after running it on seven videos. Performance scores range from 31.90% to 59.90% DER. Previous state-of-the-art scores for online diarization using audio range between 39.27% DER (for multiple microphones) and 44.61% DER (for a single microphone) [Friedland *et al.*, 2012]. Our scores, which use only gestures, are close to these previous scores.

Note that in table 4.2, the scores for false alarms (FA) are close to 0. This resulted a) from forcing our system to assume that only one person is speaking at any time and b) from evaluating the performance on speech-only segments. The non-zero FA scores in the table resulted from speakers sharing the same largest probability.

Table 4.2: Online speaker diarization results

Video	Miss	FA	Spkr	DER	DER \{FA}
IN1005	2.90	0.00	38.40	41.24	41.30
IN1009	5.50	0.00	54.40	59.90	59.90
IN1012	11.00	0.00	40.30	51.34	51.30
IN1013	12.80	0.00	36.40	49.23	49.20
IN1016	6.70	0.50	33.50	40.66	40.20
IS1009b	2.60	0.50	29.30	32.46	**31.90**
IS1009c	1.80	0.00	45.30	47.14	47.10
ALL	6.80	0.20	38.80	45.72	**45.60**

MS = Missed Speech
FA = False Alarm
Spkr = Speaker error
DER = **MS** + **FA** + **Spkr**
DER\{FA} = **DER** without **FA**

4.5.2 Signer diarization

Like the speaker diarization output, the output of the signer diarization system is also given by probability values. We say that a person is signing when the probability value for that signer is the largest. The performance scores for signer diarization are given in table 4.3. These error scores are better than those reported in chapter 3, where we used corner detection and tracking (see 3.6).

Table 4.3: Online signer diarization results

Video	Miss	FA	Sgnr	DER	DER\{FA}
KN5	5.80	0.00	9.90	15.67	15.70
PiKe	7.80	0.00	14.80	22.63	22.60
ReKe	6.90	0.00	13.00	19.93	19.90
SuJu	7.10	0.00	15.00	22.18	22.10
ALL	6.90	0.00	13.30	20.17	20.20

One main difference between signer diarization and speaker diarization is that whenever there is signing, there is definitely motion. This fact is confirmed by table 4.4, which also shows that there can be significant motion in the absence of signing. Non-signing motion makes signer diarization a non-trivial problem. If we say there is signing whenever there is motion, then we get a baseline DER score of 121.66. If we apply our online diarization algorithm, then the DER score reduces to 20.20.

Table 4.4: The proportion of time there is (no) motion when there is sign or no sign.

Sign?	**Motion**?	**Overlap**
Yes	Yes	1.00
	No	0.00
No	Yes	0.94
	No	0.06

Baseline diarization error rate (DER) = 121.66
Motion for each signer is defined as sum(MEI) > 0

4.6 Conclusions and future work

This chapter proposed and showed the use of motion history images (MHI) as a representation of gestural activity in an online speaker or signer diarization system. MHIs can efficiently represent where, how and how long motion occurred. The chapter claimed that these properties make MHIs applicable in online speaker and signer diarization systems, where motion is an integral part of uttering activity. Experiments on speaker and signer diarization problems using real data indicate that our solution is applicable in real-world applications (for example, video conferences).

Future work on diarization can extend our work in two ways. One way is by adding in extra information (for example, speech in the case of speaker diarization, or gaze in the case of signer diarization, where interlocutor(s) must be looking at the signer to be part of the conversation). The second way is to modify our model of conversation dynamics. In our conversation model, each person has an independent model of *speaking/signing*. But one can enrich the model by adding in parameters to model the relationship of listening and speaking. Such a model can, for example, encode the idea that a speaker is less likely to continue speaking if another just started speaking.

4.7 Relation to prior work

The work presented here has focused on using MHI for both speaker and signer diarization. To the best of our knowledge, this is our contribution. This work is similar to our work presented in chapter 2, where we first justified and used gestures for speaker diarization. Our work presented in chapter 2 performs speaker diarization by tracking corners, filtering out motionless corners and classifying them based on the location of the speakers. The core of that system depends on corner detection and Lucas-Kanade tracking. These operations are computationally expensive [Tomasi and Shi, 1994; Bouguet, 2001]. By contrast, our current diarization system presented in this chapter is much less computationally intensive because of the use of Motion History Image (MHI) [Davis and Bobick, 1997; Bradski and Davis, 2002; Ahad, 2013].

In terms of the modeling framework, our work is similar to Noulas and Krose [2007], who used a probabilistic framework that utilizes multi-modal information to perform online speaker diarization. The difference is that they use SIFT descriptors [Lowe, 2004] to model the visual aspect of the multimodal information, while we use MHI, a much more efficient technique. Other video features like compressed MPEG-4 features have also been used in the multimodal speaker diarization literature [Vallet *et al.*, 2013; Seichepine *et al.*, 2013; Anguera *et al.*, 2012; Friedland *et al.*, 2009]. We contribute to this literature by drawing attention to the advantages of using motion history images [Davis and Bobick, 1997; Bradski and Davis, 2002; Ahad, 2013] in speaker and signer diarization.

In summary, our work builds on and extends the literature in two ways: *a*) emphasis on the use of MHI for speaker and signer diarization *b*) an online diarization system that works on visual data. The c++ code is publicly available on `https://bitbucket.org/binyam/online-diarizer/src`.

Chapter 5

Speaker diarization using gesture and speech

Content

This chapter demonstrates the use of gesture and speaker parametric models in solving speaker diarization. The novelty of our solution is that speaker diarization is formulated as a speaker recognition problem after learning speaker models from speech samples co-occurring with gestures. This approach offers many advantages: better performance, faster computation and more flexibility. Tests on 4.24 hours of the AMI meeting data show that, compared to the AMI system, our solution makes DER score improvements of 19% on speech-only segments and 4% on all segments including silence.

Based on

Gebre, B. G., Wittenburg, P., Drude, S., Huijbregts, M., and Heskes, T. Speaker diarization using gesture and speech. In *Proceedings of Interspeech 2014: 15th Annual Conference of the International Speech Communication Association.*

Keywords

Speaker recognition, adaptation, UBM, MHI, MEI, gamma distribution

5.1 Introduction

The standard problem formulation of speaker diarization is as follows: given an audio or audio-video recording, the task is to determine the number of speakers and the segments of speech corresponding to each speaker. In this formulation, the state-of-the-art technique used to solve the problem is based on the ICSI system [Ajmera *et al.*, 2002; Friedland *et al.*, 2009; Anguera *et al.*, 2012; Tranter and Reynolds, 2006; Meignier and Merlin, 2010; Vijayasenan and Valente, 2012; Wooters and Huijbregts, 2008; Friedland *et al.*, 2012; Huijbregts *et al.*, 2012; Rouvier *et al.*, 2013]. The ICSI system performs three main tasks: speech/non-speech detection, speaker segmentation and clustering. The latter two tasks are performed iteratively using an agglomerative clustering technique based on HMMs, GMMs and BIC.

The assumption in the ICSI-based systems is that the number of speakers and speaker models remain unknown (uncertain) all along the length of signals. However, this assumption may not hold for particular scenarios where such information is known a priori, which is the case in our experiments, or can be reliably estimated at initial stages. In videos of meetings, the number of speakers can be determined from a few video frames using standard face detection algorithms [Viola and Jones, 2004]. Furthermore, speaker models, as this chapter will demonstrate, can also be estimated for each person based on speech samples co-occurring with gestures.

In chapters 2 and 4, we performed speaker diarization on meeting videos based on the hypothesis that the person who is gesturing is also the speaker. In theory, this could work well because there is a tight relationship between speech and gesture [McNeill, 1985], but, in practice, the hypothesis has limitations: speakers can speak without gesturing and gesture recognition, by itself, is a challenging problem (e.g. people may appear to be gesturing when they move for non-communicative reasons).

The goal of this chapter is to solve these limitations by using the best of both worlds. Predictions based on gestures are used to develop speaker models with the first pass on the data. With subsequent passes of the data, the learned speaker models are iteratively used to classify the frames of speech and adapt speaker models. With three iterations of classification and adaptation, we achieve a DER score that is better than the baseline (the AMI system).

5.2 Speech-gesture representation

Given that the signals from speech and gesture are different (e.g. audio is 1-dimensional and video is 2-dimensional), how can we represent them such that they can be used for efficient computation and integration? For audio, we use MFCCs and for gestures, we use Motion History Images (MHI) that we proposed and presented in chapter 4.

5.2.1 Speech representation

Speech is a time-varying signal and as such is not suitable for speaker recognition. We, therefore, convert the speech signal to MFCCs (Mel Frequency Cepstral Coefficients) [Davis and Mermelstein, 1980]. MFCCs are widely used features in speaker and speech recognition. We extract MFCC features as follows (the numbers correspond to the parameter values we selected). Our speech signal, which is sampled at 16 kHz, is divided into a number of overlapping frames, each 20 ms long (320 samples) with an overlap of 10 ms (160 samples). After multiplying each frame with a Hamming window, each frame is FFT-transformed (Fast Fourier Transform). The resulting power spectrum is then warped according to Mel-scale using 26 overlapping triangular filters producing filterbank outputs. The amplitudes of the DCT (Discrete Cosine Transform) of the logarithms of the filterbank outputs make the MFCC features. In our experiments, we take the first 20 MFCC coefficients (including the energy coefficient C_0) plus their first and second order derivatives for a total of 60-dimensional MFCC feature vector per speech frame. The HTK toolkit is used to compute the coefficients [Young *et al.*, 2006, 1997].

5.2.2 Gesture representation

To represent gestures, we use Motion History Images (MHI) that we presented in chapter 4, which we repeat in this chapter for the sake of clarity and completeness. MHI is a single stacked image that encodes motion that occurred between every frame pair for the last δ number of frames (where δ is a number we can fix ourselves). The type of information encoded in the MHI can be binary and, in which case, it is called Motion Energy Image (MEI); or it can be scalar, in which case, it is called Motion History Image.

Motion Energy Image

To represent where motion occurred, we form a Motion Energy Image. This is constructed as follows. Let $I(x, y, t)$ be an image sequence, and let $D(x, y, t)$ be a binary image sequence indicating regions of motion (we perform frame differencing). Then the binary MEI $E(x, y, t)$ is defined as follows:

$$E_\delta(x, y, t) = \bigcup_{i=0}^{\delta-1} D(x, y, t-i), \tag{5.1}$$

where δ is the temporal extent of motion (for example, a fixed number of frames). Figure 4.1(c) shows an image example of an MEI for a speaker who is also gesturing.

Motion History Image

To represent how motion occurred, we form a Motion History Image (MHI) as follows:

$$H_\tau(x,y,t) = \begin{cases} \tau & \text{if } D(x,y,t) = 1 \\ 0 & \text{else if } H_\tau(x,y,t) < (\tau - \delta), \end{cases} \tag{5.2}$$

where τ is the current time-stamp and δ is the maximum time duration constant (τ and δ are converted to frame numbers based on frame rate). Figure 4.1 (b) shows an example of an MHI for a speaker who is also gesturing. Note that an MEI image can be generated by thresholding an MHI above zero.

5.3 Our diarization system

At a high-level, our diarization system performs the following steps:

1. Train a Universal Background Model (UBM) on all audio data of the given recording.
2. Based on the location of gestures in the video, determine which speech sample belongs to which person (i.e. perform speaker diarization using gestures).
3. Adapt the UBM to create speaker models based on current predictions.
4. Use the current speaker models to identify to which speaker the next speech sample belongs (i.e. perform speaker diarization based on speaker models).
5. Repeat steps 3 and 4 N times, each time using the latest diarization predictions and speaker models. In our experiments, $N = 3$.

5.3.1 Diarization using gestures

Given a video and the number of speakers, we wish to infer, based on gestures, which person is speaking at time t. The inference is made using probabilistic models presented in chapter 4, which repeat here with changes in variable names to make distinction between audio and video features. Let each person's state (speaking or non-speaking) be represented by z_t^i and let $v_{0:t}^i$ be video measurements (i.e. gestures) for person i, the objective is then to calculate the probability of z_t^i given $v_{0:t}^i$:

$$p(z_t^i|v_{0:t}^i) = \frac{p(v_t^i|z_t^i)p(z_t^i|v_{0:t-1}^i)}{p(v_t^i|v_{0:t-1}^i)}, \tag{5.3}$$

where $p(v_t^i|v_{0:t-1}^i)$ is a normalization constant, $p(z_t^i|v_{0:t-1}^i)$ is referred to as a conversation dynamics model and $p(v_t^i|z_t^i)$ is referred to as the gesture model. The person with the highest probability, $p(z_t^i|v_{0:t}^i)$, is the gesturer and hence, the speaker. The gesture and conversation dynamics models are described below.

Gesture model

We use gamma distributions to model gestural and non-gestural activities. The assumption is that MEI is a strong indicator of gestural activity. The higher the energy (the sum of MEI values), the higher the probability of gestural activity. A gamma distribution has a shape parameter $\boldsymbol{k}$ and scale parameter $\boldsymbol{\theta}$:

$$p(v_t^i|z_t^i;\boldsymbol{k},\boldsymbol{\theta}) = \frac{(v_t^i)^{k_z-1}\exp(-\frac{v_t^i}{\theta_z})}{\theta_z^{k_z}\Gamma(k_z)} \quad \text{for } v_t^i, k_z, \theta_z > 0, \tag{5.4}$$

where $z = z_t^i$, v_t^i is the count of motion pixels in a MEI of speaker i and $z_t^i \in \{0, 1\}$ represents the probability of gestures for speaking and non-speaking person. The gamma distributions for speaking and non-speaking are the same for all speakers and their parameter values are learned from annotated development data.

Conversation dynamics

In a conversation, the act of speaking has its own dynamics. The current speaker is more likely to have been speaking for a longer time than just the current frame. We encode this type of dynamics as follows:

$$p(z_t^i|v_{0:t-1}^i) = \sum_{z_{t-1}} p(z_t^i|z_{t-1}^i)p(z_{t-1}^i|v_{0:t-1}^i), \tag{5.5}$$

where $p(z_{t-1}^i|v_{0:t-1}^i)$ is the posterior from the previous time and $p(z_t^i|z_{t-1}^i)$ is the conversation dynamics. For simplicity, we set the conversation dynamics to a fixed matrix based on heuristics: a speaker is 90% more likely to remain in the same state (speaking or non-speaking) as shown below:

$$p(z_t^i|z_{t-1}^i) = \begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix}. \tag{5.6}$$

5.3.2 Diarization using speaker models

The diarization based on gestures comes at the rate of video frame rate (40 ms). The MFCC features we get from audio come at the rate of 10ms. To make the two streams compatible, we take four MFCC feature vectors and replace them with their average vector. Given the average MFCC feature vectors, we determine which person is speaking at time t using maximum likelihood:

$$\hat{i}(t) = \arg\max_i \sum_{t'=t-\Delta}^{t+\Delta} \log p(\boldsymbol{a}_{t'}|\boldsymbol{\lambda}^i), \tag{5.7}$$

where delta, Δ, is a window of frames included for making predictions at time t and $\boldsymbol{\lambda}^i = \left\{\boldsymbol{w}^i, \boldsymbol{\mu}^i, \boldsymbol{\Sigma}\right\}$ is a speaker model for speaker i. In our experiments, Δ is set to 50 (2 seconds). The speaker models are derived from a UBM as described below.

Universal Background Model

A Universal Background Model (UBM) is a Gaussian Mixture Model (GMM) model. A GMM model is a weighted sum of M component densities:

$$p(\boldsymbol{a}_t | \{w_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}_{j=1}^{M}) = \sum_{j=1}^{M} w_j \mathcal{N}(\boldsymbol{a}_t, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j), \tag{5.8}$$

where w_j are the mixture weights satisfying $\sum_{j=1}^{M} w_j = 1$ and $\mathcal{N}(\boldsymbol{a}_t, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ are the individual component densities. Each density component j a D-variate Gaussian of the form:

$$\mathcal{N}(\boldsymbol{a}_t, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) = \frac{\exp\left\{-0.5(\boldsymbol{a}_t - \boldsymbol{\mu}_j)^T (\boldsymbol{\Sigma}_j)^{-1} (\boldsymbol{a}_t - \boldsymbol{\mu}_j)\right\}}{(2\pi)^{D/2} |\boldsymbol{\Sigma}_j|^{1/2}}, \tag{5.9}$$

where $\boldsymbol{\mu}_j$ is the mean vector and $\boldsymbol{\Sigma}_j$ is the covariance matrix.

In our system, the UBM is trained on audio features (MFCC features) from all speakers of a recording (including the silences). The UBM serves two purposes: first, it is used to derive speaker-dependent GMM models. Second, it is used to serve as a background or negative speaker model, against which each particular speaker model is compared to determine if they are speaking. Our UBM model consists of 64 60-variate Gaussian components. The covariance type is diagonal. The minimum variance value of the covariance matrix is limited to 0.01 to avoid spurious singularities [Reynolds and Rose, 1995]. Parameters of the UBM are estimated using EM algorithm [Dempster *et al.*, 1977; Pedregosa *et al.*, 2011].

Adaptation of Speaker Models

The UBM, represented by $\boldsymbol{\lambda} = \{\boldsymbol{w}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}_{\text{ubm}}$, is trained on all audio samples of a given recording. To make it model a particular speaker i, we need speech samples from speaker i and an adaptation technique. Initially, speech samples are collected for each speaker based on the occurrence of their gestures but later speech samples are collected based on speaker models. In either case, the adaptation technique is the same; we use a type of Bayesian parameter adaptation [Gauvain and Lee, 1994; Reynolds *et al.*, 2000]. Given $\boldsymbol{\lambda}$ and training speech samples for speaker i, $A^i = \{\boldsymbol{a}_1^i, \boldsymbol{a}_2^i, \ldots, \boldsymbol{a}_T^i\}$, we compute the responsibilities of each mixture component m^i in the UBM as follows:

$$p(m^i | \boldsymbol{a}_t, \boldsymbol{\lambda}) = \frac{w_m \mathcal{N}(\boldsymbol{a}_t^i, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\sum_{j=1}^{M} w_j \mathcal{N}(\boldsymbol{a}_t^i, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \tag{5.10}$$

$p(m^i|\boldsymbol{a}_t, \boldsymbol{\lambda})$ and $\boldsymbol{a}_t$ are then used to compute sufficient statistics for the weight and mean of speaker i as follows[1]:

$$n^i_m = \sum_{t=1}^{T} p(m^i|\boldsymbol{a}_t, \boldsymbol{\lambda}). \tag{5.11}$$

$$E^i_m(\boldsymbol{a}) = \frac{1}{n^i_m} \sum_{t=1}^{T} p(m^i|\boldsymbol{a}_t, \boldsymbol{\lambda})\boldsymbol{a}^i_t. \tag{5.12}$$

Using $E^i_m(\boldsymbol{a})$ and n^i_m, we can now adapt the UBM sufficient statistics for mixture m for speaker i as follows:

$$\hat{w}^i_m = [\alpha^i_m n^i_m / T + (1 - \alpha^i_m) w_m]\gamma^i. \tag{5.13}$$

$$\hat{\boldsymbol{\mu}}^i_m = \alpha^i_m E^i_m(\boldsymbol{a}) + (1 - \alpha^i_m)\boldsymbol{\mu}_m. \tag{5.14}$$

γ^i is a normalisation factor to ensure that the adapted mixture weights, $\hat{w}^i_m$, sum to unity:

$$\gamma^i = \frac{1}{\sum_{j=1}^{M} \hat{w}^i_j}. \tag{5.15}$$

α^i_m is an adaptation coefficient used to control the balance between old and new estimates for the weights and means. For each mixture m^i, a data-dependent adaptation coefficient is fixed as:

$$\alpha^i_m = \frac{n^i_m}{n^i_m + r}, \tag{5.16}$$

where r is a relevance parameter and is set to 16. For more details on these parameters, see Reynolds *et al.* [2000].

5.4 Experiments

5.4.1 Datasets

We validate our proposed solution on test data of seven video recordings ($\approx$ 4.24 hours), taken from a publicly available corpus called the AMI corpus [Carletta *et al.*, 2006]. The AMI corpus consists of annotated audio-visual data of a number of participants engaged in a meeting. The selected videos (IB4XXX) have four participants. The upper body of each participant is recorded using a separate

[1] The covariance parameter is kept the same for all speakers; adapting it with new data decreased performance.

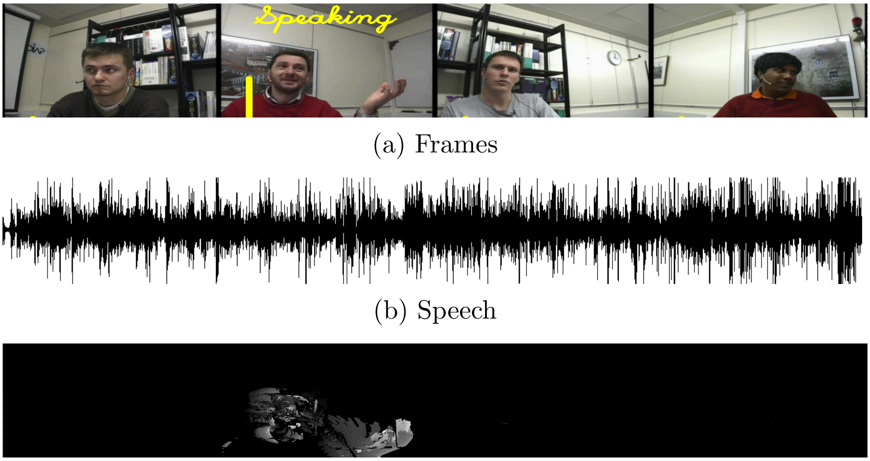
camera and we put them together before diarization. For audio, we use the mixed-headset single wave file per video. Our development data consists of 4.9 hours of videos coming from IN10XX and IS1009x. The development data are used to learn parameter values when necessary.

5.4.2 Evaluation metrics

We report our scores using Diarization Error Rate (DER) (see 2.4.2). DER consists of false alarm, missed speech and speaker errors [Anguera, 2007]. DER is known to be noisy and sensitive [Mirghafori and Wooters, 2006] but it is still widely used in many evaluations [Wooters and Huijbregts, 2008; Anguera *et al.*, 2012]. A perfect diarization system scores 0% DER, but a very bad system (e.g. a system that predicts every speaker is speaking all the time) can go over 100%.

5.5 Results and discussion

Figure 5.1 illustrates how training speech samples are collected for adapting speaker models based on predictions using gestures. The figure clearly shows that the person that is gesturing is the speaker and the MHI visualization clearly reflects it. As table 5.1 shows, this is not always true (i.e. a person could be moving without speaking or that they could be speaking without gesturing). Hence, the need to pass through the data iteratively (adapting speaker models and making predictions).



(a) Frames

(b) Speech

(c) MHI

Figure 5.1: A snapshot of IN1016-AMI meeting data: (a) Video frames with four individuals engaged in a conversation (the bar indicates probability of speaking calculated using gestures). (b) The speech waveform of the speaker. (c) The MHI of the gesturing person, which is indirectly used to adapt a speaker model for that person. The adapted speaker model is then used to identify the speaker on subsequent passes of the speech data.

Table 5.1: The proportion of time there is (no) motion when there is speech or no speech.

Speech?	**Motion**?	**Overlap**
Yes	Yes	0.96
	No	0.04
No	Yes	0.82
	No	0.18

Baseline diarization error rate (DER) = 72.09
Motion for each speaker is defined as sum(MEI) > 0

After the first diarization using gestures, we adapt the UBM to create speaker models. Based on equation 5.7, we then use the adapted speaker models to score each audio feature vector – a person is said to be speaking at frame t when the likelihood for that person is the largest in a window spanning ± 50 frames (4 seconds). Note that the assumption is that only one person is speaking at any frame. The alternative to this assumption is to set a threshold for likelihood, which may be necessary to handle overlapped speech. The scoring is repeated 3 times: new diarization results are used to adapt speaker models and new adapted speaker models are used to make new diarization. Based on this procedure, DER scores are given in tables 5.2 and 5.3. The best scores of our system come after 3 iterations and are better than the baseline scores (18.79% vs 23.28% and 29.87% vs 31.18%). The baseline system is the AMI system [Van Leeuwen and Huijbregts, 2006; Huijbregts, 2008], which is based on an agglomerative clustering and segmentation technique.

Table 5.2: Speaker diarization scores evaluated on speech-only segments. Each column in the *Speaker models* section is a diarization score based on speaker models that are adapted using diarization results from the previous column.

Diarization Error Rates (%)					
			Speaker models		
Name	**Baseline**	**Gesture**	**1st**	**2nd**	**3rd**
IB4001	19.76	53.81	33.51	27.06	23.76
IB4002	54.40	58.42	52.03	48.12	40.86
IB4003	12.20	44.53	16.13	10.48	10.35
IB4004	39.05	49.68	32.33	27.14	24.79
IB4005	13.56	37.69	17.89	18.70	19.63
IB4010	18.15	50.52	19.34	13.29	12.92
IB4011	14.59	45.76	11.53	10.64	10.37
ALL	23.28	48.04	24.14	20.20	18.79

Table 5.3: Speaker diarization scores evaluated on all segments including silences. Evaluating our system on silence segments increases DER as a result of increase in false alarms.

Diarization Error Rates (%)					
			Speaker models		
Name	**Baseline**	**Gesture**	**1st**	**2nd**	**3rd**
IB4001	38.26	82.50	61.27	54.78	51.48
IB4002	100.20	104.76	97.62	93.71	86.39
IB4003	13.20	48.89	18.13	12.47	12.34
IB4004	41.15	59.44	37.16	31.94	29.61
IB4005	16.16	47.66	23.80	24.61	25.55
IB4010	20.75	56.18	25.42	19.37	19.00
IB4011	17.59	52.57	18.27	17.38	17.09
ALL	31.18	60.99	35.23	31.28	29.87

5.6 Conclusions and future work

This study proposed a solution to the speaker diarization problem based on the exploitation of the best of two worlds: gestures and speech. The use of gestures enables the formulation of the diarization problem in a novel way. A UBM is first trained on all audio feature vectors of a given recording. The UBM is then adapted to different speakers based on the speech samples co-occurring with their gestures. Finally, the adapted speaker models are used to perform diarization (then adaptation, then diarization, then adaptation, and so on). This new approach has better performance and is faster (avoids agglomerative clustering) and offers better flexibility (better trade-off between accuracy and computational complexity).

Future work can extend our work in two directions. First, enriching the gesture model: our current gesture model is quite efficient but may fail to distinguish true gestures from other movements. Second, making an online version of our system: our current system makes multiple passes through the data but this may not be necessary: speaker models do not need much more than 90 seconds of training samples [Reynolds and Rose, 1995] and the UBM, which, in our current system, is trained on the whole audio recording, could be trained on a general population and be adapted online as more gesture and speech samples arrive.

Chapter 6

Automatic sign language identification

Content

This chapter introduces sign language identification as an important pattern recognition problem and presents a solution to it. The solution is based on the hypothesis that sign languages have varying distributions of phonemes (hand shapes, locations and movements). The chapter presents techniques of phoneme extraction from video data with experimental evaluations on two sign languages involving video clips of 19 signers. Achieved average F1 scores range from 78-95%, indicating that sign languages can be identified with high accuracy using only low-level visual features.

Based on

B. G. Gebre, P. W. Wittenburg and T. Heskes (2013). “Automatic sign language identification”. In *Proceedings of the 2013 IEEE International Conference on Image Processing (ICIP)*, pages 2626-2630, IEEE.

Keywords

Sign language, invariant moments, hand shapes, locations, movements

6.1 Introduction

The task of automatic language identification is to quickly and accurately identify a language given any utterance in the language. The correct identification of a language enables efficient deployment of tools and resources in applications that include machine translation, information retrieval and routers of incoming calls to a human switch-board operator fluent in the identified language. All these applications require language identification systems that work with near perfect accuracy.

Language identification is a widely researched area in written and spoken modalities [Dunning, 1994; Muthusamy *et al.*, 1994a; Zissman, 1996; Torres-Carrasquillo *et al.*, 2002; Singer *et al.*, 2012]. The literature shows varying degrees of success depending on the modality. Languages in their written forms can be identified to about 99% accuracy using Markov models [Dunning, 1994]. Languages in their spoken forms can be identified to an accuracy that ranges from 79-98% using different models (GMM, PRLM, parallel PRLM) [Zissman, 1996; Singer *et al.*, 2003]. What is the accuracy for automatic sign language identification?

Even though extensive literature exists on sign language recognition [Starner and Pentland, 1997; Starner *et al.*, 1998; Gavrila, 1999; Cooper *et al.*, 2012a], to the best of our knowledge, no published work existed on automatic sign language identification prior to this work. In this chapter, we propose a system for sign language identification and run experimental tests on two sign languages (British and Greek). The best performance obtained, measured in terms of average F1-score, is 95%. This score is much higher than 50%, the score that we would expect from a random binary classifier. Interestingly, this performance is achieved using low-level visual features. The rest of the chapter gives more details.

6.2 Sign language phonemes

A signer of a given sign language produces a sequence of signs. According to Stokoe [2005], each sign consists of phonemes called *hand shapes*, *locations* and *movements*. The phonemes are made using one hand or both hands. In either case, each active hand assumes a particular *hand shape*, a particular *orientation* in a particular *location* (on or around the body) and with a possible particular *movement*.

The aforementioned phonemes that come from hands make up the *manual signs* of a given sign language. But the whole message of a sign language utterance is contained not only in *manual signs* but also in *non-manual signs*. Non-manual signs include facial expressions, head/shoulder motion and body posture. Note that this work does not attempt to use non-manual signs for language identification.

There are two systems that attempt to formally describe the phonemes of sign languages: the Stokoe system and the Hold-Movement system. The Stokoe system is proposed by Stokoe and the central idea in this model is that signs can be broken down into phonemes corresponding to location, hand shape, and movement (put in that order) [Stokoe, 2005]. An alternative to Stokoe's model is the Move-Hold model [Liddell and Johnson, 1989]. The Move-Hold (M-H) system emphasizes the sequence aspect of segments of signs. Each segment is described by a set of features of *hand shape*, *orientation*, *location* and *movement*. A hold is defined as a period of time during which hand shape, orientation, location, movement, and nonmanuals are held constant. A movement is defined as a transition between holds during which at least one of the four parameters changes.

Which description system do we use for sign language identification? Our work uses the idea than signs can be broken into phonemes, an idea that is common to both the Stokoe and M-H systems; we extract video features to represent locations, hand shapes and movements. But, because we extract the features from a sequence of at most two frames, we think that we are using the Move-Hold (M-H) system.

6.3 Our sign language identification method

An ideal sign language identification (SLID) system should be independent of content, context, and vocabulary and should be robust with regard to signer identity and noise and distortions introduced by cameras. Some of the desirable features of an ideal SLID system are:

1. should be robust with respect to intra- and inter-signer variability.
2. should be insensitive to camera-induced variations (scale, translation, rotation, view, occlusion, etc).
3. increasing the number of target sign languages should not degrade performance (there are at least 300 sign languages[1]).
4. decreasing the duration of the test utterance should not degrade system performance.

Our proposed SLID system has four subcomponents and each subcomponent attempts to address points 1, partly 2 (scale and translation), 3 and 4. The system subcomponents are: *a*) skin detection *b*) feature extraction *c*) modeling *d*) identification. We describe each subcomponent in the following subsections.

6.3.1 Skin detection

We use skin color to detect hands/face [Vezhnevets *et al.*, 2003; Phung *et al.*, 2005]. Skin color has practically useful features. It is invariant to scale and orientation and it is also easy to compute. But it also has two problems: *1*) perfect skin color ranges for one video do not necessarily apply to another *2*) some objects in the video have the same color as the hands/face. To solve the first problem, we did explicit manual selection of the skin color RGB ranges in a way that is comparable to Kovac *et al.* [2003]; other skin detection approaches (i.e. based on parametric

[1] `http://en.wikipedia.org/wiki/List_of_sign_languages`

and non-parametric distributions) did not perform any better on our dataset. To solve the second problem, we applied dilation operations and constraint rules to remove objects that are identified as face or hands but do not have the right sizes.

6.3.2 Feature extraction

Given that the phonemes of sign language are formed from a set of hand shapes (N), in a set of locations (L) and with movement types (M), we encode shapes using Hu-moments, locations using discrete grids (binary patterns) and movements as XORs of two consecutive location grids (binary patterns).

Hand-shapes/Orientations

> To encode hand shapes and orientations of the hands, we use the Hu set of seven invariant moments ($H_1 - H_7$) [Hu, 1962], calculated from the gesture space of the signer. The gesture space is the region bounded by the external lines of the grids shown in figure 6.1. The seven Hu moments capture shapes and arrangements of the foreground objects (in this case, skin blobs). Formed by combining normalized central moments, these moments offer invariance to scale, translation, rotation and skew [Hu, 1962]. They are among the most widely used features in sign language recognition [Cooper *et al.*, 2012a]. Note that an image moment is a weighted average (moment) of the image pixels' intensities.

Locations/Hand-arragements

> To encode hand locations of the signer, we use grids of 10×10 with the center of the face used as a reference. To find the center of the face, we used the Viola Jones face detector [Viola and Jones, 2001]. The position and scale of the detected face is used to calculate the position and scale of the grid. The center of the grid is fixed at the third row and in the middle column (See figure 6.1). Each cell in the grid is a quarter of the height of the detected face [Cooper *et al.*, 2012a]. A cell is assigned 1 if more than 50 percent of the area is covered by skin, otherwise, it will be assigned 0. These cells are changed into a single row vector of size 100 by concatenating the various rows – one after the other.

Movements

> To encode the types of body movements, we compare the locations of hands and face in the current frame with respect to the previous frame. The motion is then captured by XORing (the absolute of pairwise element subtraction of) two frame location vectors. The location vectors are obtained from the cell grids as described above.

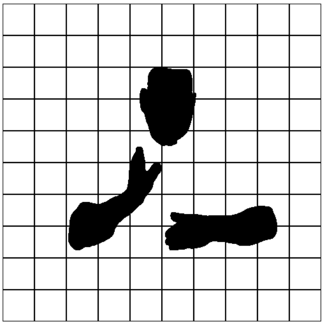


Figure 6.1: Each cell in the grid is a square whose side is a quarter of the height of the face. The size of the face is determined by the Viola Jones algorithm [Viola and Jones, 2001] using the data and implementation from the OpenCV library [Bradski and Kaehler, 2008].

6.3.3 Learning using random forest

We use a random forest algorithm for sign language classification [Breiman, 2001; Pedregosa *et al.*, 2011]. A random forest algorithm generates many decision tree classifiers and aggregates their results [Breiman, 2001]. Its attractive features include high performance [Caruana and Niculescu-Mizil, 2006], greater flexibility (no need for feature normalization and feature selection) and high stability (small parameter changes do not affect performance). Algorithm 6.1 shows how random forest works for classification. The algorithm is first trained on labeled data as shown in algorithm 6.1 and then predictions of new data are made by aggregating the predictions of the N_{trees}.

Algorithm 6.1 Random forest training

```
Require: $\{x, y\}$ pairs of data
Ensure: $N_{\text{trees}}$ predictors (Random forest)
Let $N_{\text{trees}}$ be the number of trees to build
for each of $N_{\text{trees}}$ iterations do
    Select a new bootstrap sample from training set
    //Grow an un-pruned tree on this bootstrap
    for each node do
        randomly sample $m$ of the feature variables
        choose the best split from among those variables using gini impurity measure
    end for
end for
```

The random sampling of features at every node in a tree prevents random forests from overfitting and makes them perform very well compared to many other classifiers [Breiman, 2001]. In our experiments, we fixed N_{trees} to 10 and m to 14 (14 $\approx \sqrt{207}$, the size of our feature vector).

6.3.4 Identification

During identification, an unknown sign language utterance of frame length T is first converted to frame vectors of length T, with each frame vector x_t having features of 207-dimension. These feature vectors are then scored against each language. With the assumption that the observations (feature vectors x_i) are statistically independent of each other, the scoring function is a log-likelihood function and is defined as:

$$L(x/l) = \sum_{t=1}^{T} \log p(x_t/l), \tag{6.1}$$

where T is the number of frames and $p(x_t/l)$ is a probability of x_t for a given language l. The predicted class probabilities of a given feature vector is computed as the mean predicted class probabilities of the trees in the forest [Pedregosa *et al.*, 2011]. The language $\hat{l}$ of the unknown utterance is chosen as follows:

$$\hat{l} = \arg\max_{l}(\sum_{t=1}^{T} \log p(x_t/l) + \log p(l)), \tag{6.2}$$

where $p(l)$ is the prior probability of choosing either sign language, which we fixed to 0.5 (making it irrelevant in our experiments).

6.4 Experiment

We test our sign language modeling and identification system on data that is publicly accessible from the Dicta-Sign Corpus [Efthimiou *et al.*, 2009]. The corpus has recordings for four sign languages with at least 14 signers per language and a session duration of approximately 2 hours using the same elicitation materials across languages. From this collection, we selected 9 signers of British sign language and 10 signers of Greek sign language[2]. The signers have been selected with the criterion that their skin color is clearly distinct from both the background and their clothes. Table 6.1 gives more details of the experiment data.

[2] Only British and Greek sign languages corpora were publicly available for download from the Dicta-Sign Corpus (http://www.dictasign.eu).

Table 6.1: Sign language identification: experiment data

Sign Language	**British**	**Greek**	**Total**
Total length (in hours)	8.9	7.17	16.07
Number of signers	9	10	19
Number of clips	186	209	395
Average clip size (in minutes)	2.86	2.06	2.46

6.5 Results and discussion

We evaluate the performance of our identification system in terms of precision, recall and F1-score. We also evaluate the impact on performance of varying *a*) the number of training clips, and *b*) the length (in seconds) of the test clips. Table 6.2 indicates that high accuracy scores can be obtained by training on one half of the data and testing on the other half. Figure 6.2 shows performance variations as a function of training data size and the length of the test clip; it indicates that 10 seconds of test clip is good enough to achieve about an F1 score of 90% . Ten seconds of utterance correspond to about 25 signs [Klima and Bellugi, 1979].

Table 6.2: Sign language identification results: utterances in the training and the test data are different but they are not signer independent.

Number of training clips = 197 (random 50% of clips)
Number of test clips = 198 (the remaining 50%) of clips
Clip size = 60 seconds

	Precision	**Recall**	**F1-score**	**Support**
BSL	0.94	0.96	0.95	94
GSL	0.96	0.94	0.95	104
Average/total	0.95	0.95	0.95	198

As clips of the same signers occur in both training and test data, can we be sure that we are not identifying people instead of sign languages? In order to answer this, we trained our system on clips of a group of 11 randomly selected signers and tested on clips of the remaining 8 signers. Even though the score is now less (it decreases from 95% to 78%), we can still see that our system is doing more than signer identity classification (see table 6.3 for signer independent scores).

Are we really identifying sign languages and not some other random pattern? In order to answer this question, we assigned random labels to each clip and trained our system on random 50% of the clips and tested on the remaining 50%. Performance on different runs produced F1 scores that averaged to about 50% – indicating

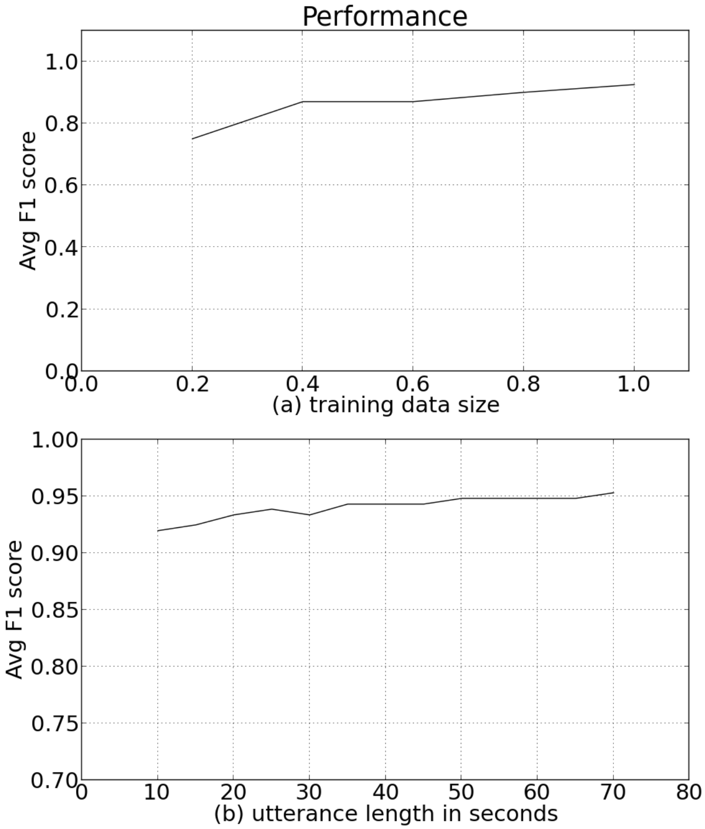


Figure 6.2: (a) The impact of varying the fraction of training data (shown on x-axis) on the average F1 score (shown on y-axis). (b) The impact of varying the test utterance length (shown on the x-axis in seconds) on the average F1 score (shown on the y-axis).

Table 6.3: Signer independent classification results

Number of training clips = 248 (11 signers)
Number of test clips = 147 (from 8 unseen signers)
Clip size = 60 seconds

	Precision	**Recall**	**F1-score**	**Support**
BSL	0.77	0.72	0.74	64
GSL	0.79	0.83	0.81	83
Average/total	0.78	0.78	0.78	147

that our system is not picking upon any random pattern. What about systematic patterns like the characteristics of the video or people that are unique to each language? The video characteristics of the two sign language corpora are similar as they were deliberately designed to be parallel for research purposes. However, the bodily characteristics of the signers of each language could be different.

How can we distinguish bodily characteristics from sign languages? To answer this correctly, further research needs to be done with sign language clips produced by multilingual signers (the same signers producing utterances in two or more sign languages). For now, we can get insight by examining the most important features discovered by the random forest classifier[3].

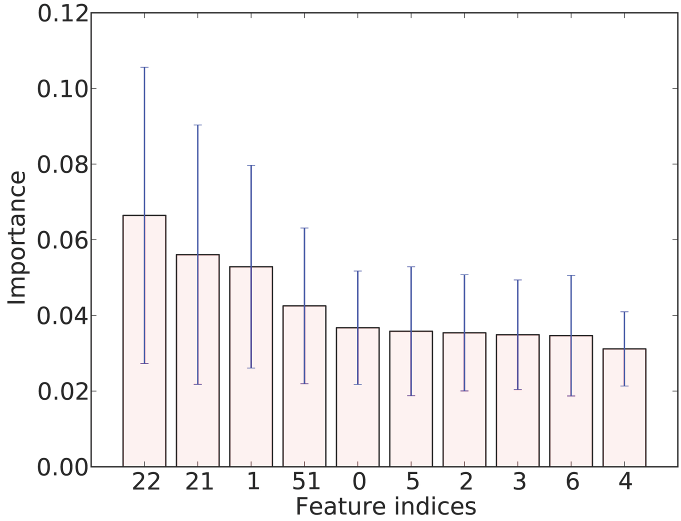


Figure 6.3: The importance of the ten most informative features out of 207 features (7 for shapes, 100 for locations and another 100 for movements, indexed in that order). The error bars are standard deviations of the feature importances for the ten trees.

Figure 6.3 shows the relative importance of the ten most important features indexed by their position in the feature vector. The figure indicates that feature indices 22 and 21 are the most important. Interestingly, these refer to locations above the head slightly to the left. Most of the shape features (the Hu-moments, indexed by numbers 0 through 6) are also among the most important. No movement feature ended up among the top ten.

[3] The relative rank (i.e. depth) of a feature used as a decision node in a tree is used to evaluate its relative importance. A feature used at the top of a tree contributes to the final prediction decision of a larger fraction of the input samples. The expected fraction of the samples it contributes to is used as an estimate of the relative importance of the features.

6.6 Conclusions and future work

The work in this chapter makes a contribution to the existing literature on automatic language identification by *a*) drawing attention to sign languages, and *b*) proposing a method for identifying them. The proposed sign language identification system has the attractive features of simplicity (it uses low-level visual features without any reference to phonetic transcription) and high performance (it uses a random forest algorithm). The system performs with an accuracy ranging from 78-95% (F1-score). From this performance, we can draw one important conclusion: sign languages, like written and spoken languages, can be identified using low level features.

Future work should extend this work to identify several sign languages. Other possible sign language identification methods should also be explored (language identification methods that perform best in written and spoken languages are phonotactic – Ngram language models). Future work should also examine automatic phoneme extraction and clustering algorithms with the view to developing sign language typology (families of sign languages). In the next chapter, we address sign identification using unsupervised feature learning techniques and conduct experiment on 6 sign languages.

Chapter 7

Unsupervised feature learning for sign language identification

Content

This chapter presents a method for identifying sign languages solely from short video samples. The method uses K-means and sparse autoencoder to learn 2D and 3D feature maps from unlabelled video data. Using these feature maps and by the process of convolution and pooling, classifier features are extracted and trained to discriminate between six sign languages. Experimental evaluation, involving 30 signers, shows an average best accuracy of 84%.

Based on

B. G. Gebre, O. Crasborn, P. Wittenburg, S. Drude and T. Heskes (2014). “Unsupervised feature learning for visual sign language identification”. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 370-376. Association for Computational Linguistics.

Keywords

Unsupervised features, k-means, sparse autoencoder, convolution, pooling

7.1 Introduction

As presented in the previous chapter, the task of automatic language identification is to quickly identify a language given any utterance in the language. Performing this task accurately is key in applications involving multiple languages such as machine translation and cross-lingual information retrieval. In machine translation, we would like to know the source language before we load the resources and tools involved in the translation. In information retrieval, we would like to index and search information within or across languages.

Previous research on language identification is heavily biased towards written and spoken languages [Dunning, 1994; Zissman, 1996; Li *et al.*, 2007; Singer *et al.*, 2012; Jiang *et al.*, 2014]. Written languages can be identified to about 99% accuracy using Markov models [Dunning, 1994]. This accuracy is so high that current research has shifted to related more challenging problems: language variety identification [Zampieri and Gebre, 2012], native language identification [Tetreault *et al.*, 2013] and identification at the extremes of scales: many more languages, smaller training data and shorter document lengths [Baldwin and Lui, 2010].

Spoken languages can be identified to accuracies that range from 79-98% using different models (GMM, PRLM, parallel PRLM) [Zissman, 1996; Singer *et al.*, 2003]. The methods used in spoken language identification have also been extended to a related class of problems: native accent identification [Chen *et al.*, 2001; Choueiter *et al.*, 2008; Wu *et al.*, 2010] and foreign accent identification [Teixeira *et al.*, 1996].

While some work exists on sign language recognition [Starner and Pentland, 1997; Starner *et al.*, 1998; Gavrila, 1999; Cooper *et al.*, 2012a], very little research exists on sign language identification. In chapter 6, we showed that sign language identification can be done using linguistically motivated features (i.e. features encoding hand shape, location and movement). We reported accuracies of 78% and 95% on signer independent and signer dependent identification of two sign languages (British and Greek). In the current chapter, we extend this research in the following two ways. First, we present a method to identify sign languages using features learned by unsupervised techniques [Hinton and Salakhutdinov, 2006; Coates *et al.*, 2011]. Second, we evaluate the method on six sign languages under different conditions involving 30 signers (5 different signers per language).

In this chapter, we make two main contributions. First, we show that unsupervised feature learning techniques, currently popular in many pattern recognition problems, also work for visual sign languages. More specifically, we show how K-means and sparse autoencoder can be used to learn features for sign language identification. Second, we demonstrate the impact on performance of varying the number of features (aka feature maps or filter sizes), the patch dimensions (from 2D to 3D) and the number of frames (video length).

7.2 The challenges in sign language identification

The challenges in sign language identification arise from three sources: *1*) iconicity in sign languages *2*) differences between signers *3*) diverse environments.

7.2.1 Iconicity in sign languages

The relationship between forms and meanings in language is not totally arbitrary [Perniss *et al.*, 2010]. Both signed and spoken languages manifest iconicity, that is forms of words or signs are motivated by the meaning of the word or sign. While sign languages show a lot of iconicity in the lexicon [Taub, 2001], this has not led to a universal sign language. The same concept can be iconically realised by the manual articulators in a way that conforms to the phonological regularities of the languages, but still lead to very different sign forms.

Iconicity is also used in the morphosyntax and discourse structure of all sign languages and there we see many similarities between sign languages. Both real-world and imaginary objects and locations are visualised in the space in front of the signer, and can have an impact on the articulation of signs in various ways. Also, the use of constructed action appears to be used in many sign languages in similar ways. The same holds for the rich use of non-manual articulators in sentences and the limited role of facial expressions in the lexicon: these too make sign languages across the world very similar in appearance, even though the meaning of specific articulations may differ [Crasborn, 2006].

7.2.2 Differences between signers

Just as speakers have different voices unique to each individual, signers also have different signing styles that are likely unique to each individual. Signers' uniqueness results from how they articulate the shapes and movements that are specified by the linguistic structure of the language. The variability between signers either in terms of physical properties (hand sizes, skin color, etc) or in terms of articulation (movements) is such that it does not affect the understanding of the sign language by humans, but that it may be difficult for machines to generalize over multiple individuals. At present we do not know whether the differences between signers using the same language are of a similar or different nature than the differences between different languages. At the level of phonology, there are few differences between sign languages, but the differences in the phonetic realization of words (their articulation) may be much larger.

7.2.3 Diverse environments

The visual activity of signing comes in the context of a specific environment. This environment can include the visual background and camera noises. The background objects of the video may also include dynamic objects – increasing the ambiguity of signing activity. The properties and configurations of the camera induce variations of scale, translation, rotation, view, occlusion, etc. These variations, coupled with lighting conditions, may introduce noise. These challenges are by no means specific to sign interaction, and are found in many other computer vision tasks.

7.3 Feature and classifier learning

Our system performs two important tasks. First, it learns a feature representation from patches of unlabelled raw video data using sparse autoencoders and K-means unsupervised learning techniques [Hinton and Salakhutdinov, 2006; Coates *et al.*, 2011]. Second, it looks for activations of the learned representation (by convolution) and uses these activations to learn a classifier to discriminate between sign languages.

7.3.1 Unsupervised feature learning

Given samples of sign language videos (unknown sign language with one signer per video), our system performs the following steps to learn a feature representation (note that these video samples are separate from the video samples that are later used for classifier learning or testing):

1. **Extract patches**

   Extract small videos (hereafter called patches) randomly from anywhere in the video samples. We fix the size of the patches such that they all have r rows, c columns and f frames and we extract patches m times. This gives us $\boldsymbol{X} = \{x^{(1)}, x^{(1)}, \ldots, x^{(m)}\}$, where $x^{(i)} \in R^N$ and $N = r * c * f$ (the size of a patch). For our experiments, we extract 100,000 patches of size $15 * 15 * 1$ (2D) and $15 * 15 * 2$ (3D).

2. **Normalize and whiten the patches**

   There is evidence that normalization and whitening [Hyvärinen and Oja, 2000] improve performance in unsupervised feature learning [Coates *et al.*, 2011]. We therefore normalize every patch $x^{(i)}$ by subtracting the mean and dividing by the standard deviation of its elements. We added a small value to the variance before division to avoid division by zero (for example, 10 when the values are pixel intensities [Coates *et al.*, 2011]). Note that, for visual data, normalization corresponds to local brightness and contrast normalization.

   After normalizing, we perform ZCA whitening on the patches. This is done by rescaling each feature by $1/\sqrt{\lambda_i + \epsilon}$, where λ_i are eigenvalues and ϵ is a small amount of regularization (in our study, set to 0.1). The purpose of whitening is to make sure that the features in the training data *a*) are less correlated with each other, and *b*) have the same variance. This is important because the raw input of videos is redundant (i.e. adjacent pixel values are highly correlated).

3. **Learn a feature-mapping**

   Our unsupervised algorithm takes in the normalized and whitened dataset $\boldsymbol{X} = \{x^{(1)}, x^{(1)}, \ldots, x^{(m)}\}$ and maps each input vector $x^{(i)}$ to a new feature vector of K features ($f : R^N \rightarrow R^K$). We use two unsupervised learning algorithms: K-means, and sparse autoencoders.

   (a) **K-means clustering**: we train K-means to learn K $c^{(k)}$ centroids that minimize the distance between data points and their nearest centroids [Coates and Ng, 2012]. Given the learned centroids $c^{(k)}$, we measure the distance of each data point (patch) to the centroids. Naturally, the data points are at different distances to each centroid. We keep the distances that are below the average of the distances and we set the others to zero:

   $$f_k(x) = \max\{0, \mu(z) - z_k\} \tag{7.1}$$

   where $z_k = ||x - c^{(k)}||^2$ and $\mu(z)$ is the mean of the elements of z.

   (b) **Sparse autoencoder**: we train a single layer autoencoder with K hidden nodes using backpropagation to minimize the squared reconstruction error. Figure 7.1 shows a single layer sparse autoencoder, representative of the autoencoder implemented in our study. To make the sparse autoencoder learn a more interesting function than a trivial identity function, we impose a constraint on the structure at the hidden layer. We do this by either limiting the number of hidden nodes to a number (K) that is less than the input size or by imposing sparsity constraint on the activation of each hidden node. For the latter case, we set the average activation of each hidden node $\hat{\rho}_j$ to some constant ρ (in our case, ρ is set to 0.01). To satisfy the constraint, we add a penalty term to our autoencoder objective function. The penalty parameter uses Kullback-Leibler (KL) divergence and penalizes $\hat{\rho}_j$ deviating significantly from ρ.

   At the hidden layer, the features are mapped using a rectified linear (ReL) function [Maas *et al.*, 2013] as follows:

   $$f(x) = g(Wx + b) \tag{7.2}$$

   where $g(z) = \max(z, 0)$. Note that ReL nodes have advantages over sigmoid or tanh functions; they create sparse representations and are suitable for naturally sparse data [Glorot *et al.*, 2011].

From K-means, we get K R^N centroids and from the sparse autoencoder, we get $W \in R^{KxN}$ and $b \in R^K$ filters. We call both the centroids and filters as the learned features (or feature maps).

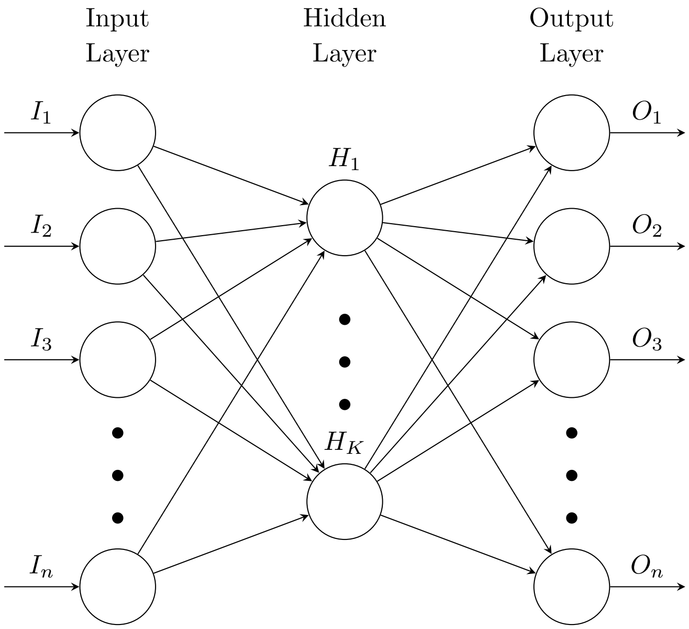


Figure 7.1: Sparse autoencoder: a single layer sparse autoencoder is a neural network with three layers, where the output is set the same as the input. By making the number of hidden nodes smaller than the number of input nodes or by imposing a sparsity constraint on the activation of each hidden node (overcomplete sparse representations), sparse autoencoder is able to discover structure in the input.

7.3.2 Classifier learning

Given the learned features, the feature mapping functions and a set of labeled training videos, we extract features as follows:

1. **Convolutional extraction**

   Extract features from equally spaced sub-patches covering the video sample. This is done by sliding a window that moves by 1 pixel row-wise and column-wise for the 2D case. For the 3D case, it is a sliding box that moves by 1 pixel row-wise, column-wise and time-wise. Convolution takes a long time – $\mathcal{O}(Kmn^2t)$, where K refers to the number of feature maps, m the number of videos, n^2 the resolution of videos and t the video length. Note that we have not included the size of the feature maps in the computational complexity.

2. **Pooling**

   Pool features together over four non-overlapping regions of the input video to reduce the number of features. We perform max pooling for K-means and mean pooling for the sparse autoencoder over 2D regions (per frame) and over 3D regions (per all sequence of frames).

3. **Learning**

   Learn a linear classifier to predict the labels given the feature vectors. This is a standard supervised learning setup. We use a logistic regression classifier and support vector machines [Pedregosa *et al.*, 2011].

The extraction of classifier features through convolution and pooling is illustrated in figure 7.2.

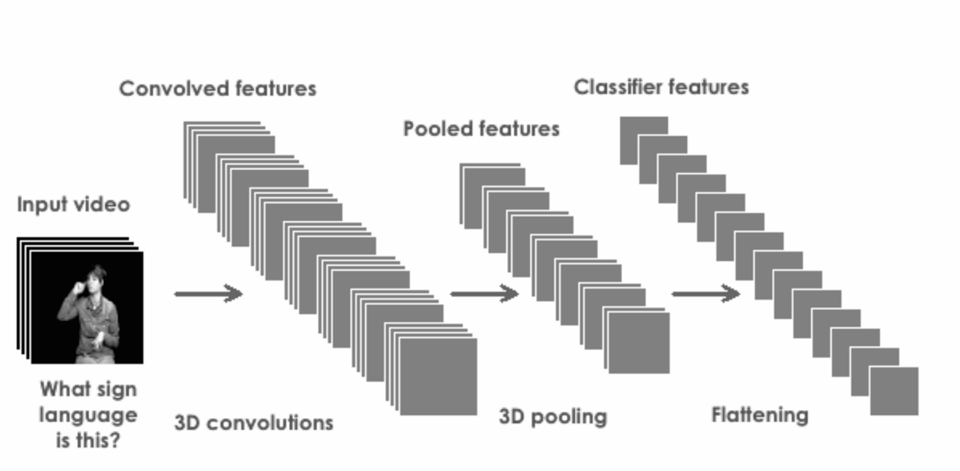


Figure 7.2: Illustration of feature extraction based on convolution and pooling using 7 filters: each 3D block in the convolution features is the result of convolution between a filter (feature map) and the video. Each block in the convolved features then goes through the process of pooling, where values in 8 non-overlapping regions are pooled over.

7.4 Experiments

7.4.1 Datasets

Our experimental data consist of videos of 30 signers equally divided between six sign languages: British (BSL), Danish (DSL), French Belgian (FBSL), Flemish (FSL), Greek (GSL), and Dutch (NGT). The data for the unsupervised feature learning comes from half of the BSL and GSL videos in the Dicta-Sign corpus[1] (16 signers). Part of the other half, involving 5 signers, is used along with the other sign language videos for learning and testing classifiers. Videos of the other sign languages came from different sources.

For the unsupervised feature learning, two types of patches are created: 2D $(15 * 15)$ and 3D $(15 * 15 * 2)$. Each type consists of 100,000 randomly selected patches and involves 16 different signers. For the supervised learning, 200 videos

[1] `http://www.dictasign.eu/`

(consisting of 1 through 4 frames taken at a step of 2) are randomly sampled per sign language per signer (for a total of 6,000 samples).

7.4.2 Data preprocessing

The data preprocessing stage has two goals.

First, to remove any non-signing signals that remain constant within videos of a single sign language but that are different across sign languages. For example, if the background of the videos is different across sign languages, then classifying the sign languages could be done with perfection by using signals from the background. To avoid this problem, we removed the background by using background subtraction techniques and manually selected thresholds. The background is formed from a small patch from the top left corner of the first frame of the video and rescaled to the resolution of the video. Treating the top-left corner patch as background works because the videos have a more or less uniform background.

The second reason for data preprocessing is to make the input size smaller and uniform. The videos are colored and their resolutions vary from $320*180$ to $720*576$. We converted the videos to grayscale and resized their heights to 144 and cropped out the central $144 * 144$ patches.

7.4.3 Evaluation

We evaluate our system in terms of average accuracies. We train and test our system in leave-one-signer-out cross-validation, where videos from four signers are used for training and videos of the remaining signer are used for testing. We repeat this as many times as the number of signers. Classification algorithms are used with their default settings and the classification strategy is one-vs.-rest.

7.5 Results and discussion

Average classification accuracies using different classifiers, video lengths, and K features are presented in table 7.1 for 2D feature maps and table 7.2 for 3D feature maps. Our best average accuracy (84.03%) is obtained using 500 K-means features which are extracted over four frames (taken at a step of 2). This accuracy obtained for six languages is much higher than the 78% accuracy obtained for two sign languages presented in chapter 6. In chapter 6, we used linguistically motivated features (hand shapes, movements and locations) that are extracted over video lengths of at least 10 seconds. The current system uses learned features that are extracted over much smaller video lengths (about half a second). Note that the disadvantage of the current system is its high computational complexity; it took us days to extract features.

Tables 7.1 and 7.2 indicate that K-means performs better with 2D filters and that sparse autoencoder performs better with 3D filters. With smaller filter sizes,

1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10

(a) K-means features

(b) Sparse autoencoder features

Figure 7.3: 100 features (filters or feature maps) learned from 100,000 patches of size 15 ∗ 15. K-means learned relatively more curving edges than the sparse auto encoder.

Table 7.1: 2D filters (15 ∗ 15): Leave-one-signer-out cross-validation average accuracies.

	K-means			**Sparse Autoencoder**		
K	**LR-L1**	**LR-L2**	**SVM**	**LR-L1**	**LR-L2**	**SVM**
# of frames = 1						
100	69.23	70.60	67.42	73.85	**74.53**	71.8
300	76.08	77.37	74.80	72.27	70.67	68.90
500	**83.03**	79.88	77.92	67.50	69.38	66.20
# of frames = 2						
100	71.15	72.07	67.42	72.78	**74.62**	72.08
300	77.33	78.27	76.60	71.85	71.07	68.27
500	**83.58**	79.50	79.90	67.73	70.15	66.45
# of frames = 3						
100	71.42	73.10	67.82	65.70	67.52	63.68
300	78.40	78.57	76.50	**72.53**	71.68	68.18
500	**83.48**	80.05	80.57	67.85	70.85	66.77
# of frames = 4						
100	71.88	73.05	68.70	64.93	67.48	63.80
300	79.32	78.65	76.42	**72.27**	72.18	68.35
500	**84.03**	80.38	80.50	68.25	71.57	67.27

K = Number of features (# of centroids or hidden nodes)
LR-L? = Logistic Regression with L1 or L2 penalty
SVM = SVM with linear kernel

sparse autoencoder performs better than K-means. Note that features from 2D filters are pooled over each frame and concatenated, whereas features from 3D

Table 7.2: 3D filters ($15 * 15 * 2$): Leave-one-signer-out cross-validation average accuracies.

	K-means			**Sparse Autoencoder**		
K	**LR-L1**	**LR-L2**	**SVM**	**LR-L1**	**LR-L2**	**SVM**
# of frames = 2						
100	70.63	69.62	68.87	67.40	66.53	65.73
300	73.73	74.05	73.03	72.83	73.48	70.52
500	75.30	**76.53**	75.40	72.28	**74.65**	68.72
# of frames = 3						
100	72.48	73.30	70.33	68.68	67.40	68.33
300	74.78	74.95	74.77	74.20	74.72	70.85
500	77.27	**77.50**	76.17	72.40	**75.45**	69.42
# of frames = 4						
100	74.85	73.97	69.23	68.68	67.80	68.80
300	76.23	76.58	74.08	74.43	75.20	70.65
500	**79.08**	78.63	76.63	73.50	**76.23**	70.53

Table 7.3: Confusion matrix – confusions averaged over all settings for K-means and sparse autoencoder with 2D and 3D filters (for all # of frames, all filter sizes and all classifiers).

	BSL	**DSL**	**FBSL**	**FSL**	**GSL**	**NGT**
BSL	**56.11**	2.98	1.79	3.38	*24.11*	11.63
DSL	2.87	**92.37**	0.95	0.46	3.16	0.18
FBSL	1.48	1.96	**79.04**	4.69	6.62	6.21
FSL	6.96	2.96	2.06	**60.81**	*18.15*	9.07
GSL	5.50	2.55	1.67	2.57	**86.05**	1.65
NGT	9.08	1.33	3.98	*18.76*	4.41	**62.44**

filters are pooled over all frames. For K-means, max pooling is performed. For sparse autoencoder, mean pooling is performed, as it performed poorly with max pooling.

Which filters are active for which sign language? We illustrate this with the smallest number of filters that we have (i.e. 100). Figure 7.3 shows the 100 features learned by K-means and sparse autoencoder. How are these filters activated for each sign language? Figure 7.4 shows a visualization of the strength of filter activation for each sign language. It shows the weight of the coefficients of each filter in the four non-overlapping pooled regions of the video frame for the six languages.

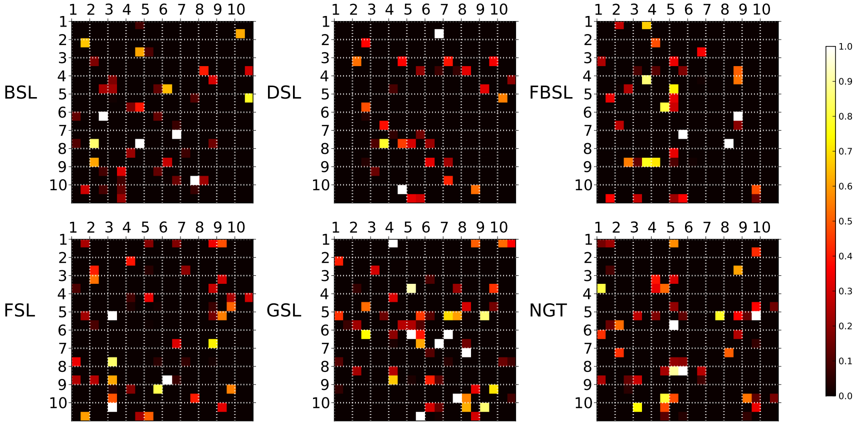
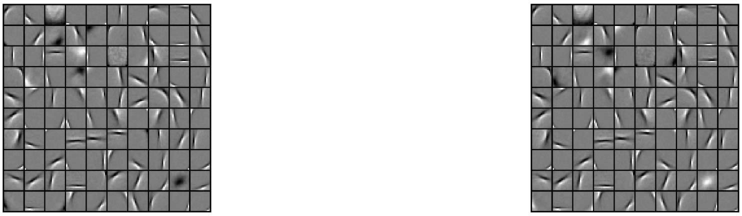


Figure 7.4: Visualization of coefficients of Lasso (logistic regression with L1 penalty) for each sign language with respect to each of the 100 filters of the sparse autoencoder. The 100 filters are shown in figure 7.3 (b). Each grid cell represents a frame and each filter is activated in 4 non-overlapping pooling regions.



(a) K-means features (at time t)

(b) K-means features (at time $t-1$)

Figure 7.5: K-means 3D features

Classification confusions are shown in table 7.3. We can see that the best average accuracy is obtained for Danish sign language (92.37%) and the worst for British sign language (56.11%). Most sign languages are confused with Greek sign language.

What do the learned features represent? This is hard to answer without knowledge of the sign languages. There is, however, one feature type that we can easily see from 3D filters and this is movement. The change in shape of a filter from one form to another and the appearance or disappearance of a filter tells us that a change or movement has taken place. In figure 7.5, we can see that while most corresponding cells from figures 7.5 (a) and 7.5 (b) are nearly the same, others are different. For example, the filter at the 9th row and 9th column is a filter for motion (the filter turns from black to white).

7.6 Conclusions and future work

This chapter presented a system for determining the identity of sign languages from raw videos. The system uses unsupervised feature learning techniques to capture features which are then used to learn a classifier. In a leave-one-signer-out cross-validation involving 30 signers and 6 sign languages, the method achieves about 84% average accuracy. This score is better than the 78% accuracy presented in the previous chapter (chapter 6), which used handcrafted features. Given that sign languages are under-resourced, unsupervised feature learning techniques are useful tools for sign language identification.

Future work can extend this work by: *a*) increasing the number of sign languages and signers to check the stability of the learned feature activations and to relate these to iconicity and signer differences, and *b*) comparing our shallow method with deep learning techniques. In our experiments, we used a single hidden layer of features, but it is worth looking into deeper layers to gain more insight into the hierarchical composition of features in sign languages.

Other questions for future work are: how good are human beings at identifying sign languages? How much of the problem in sign language identification is related to issues arising from computer vision? How accurate is sign language identification based on glosses (transcription)? This will tell us how much of the challenge is related to the computer vision and how much of it is linguistic. Can a machine be used to evaluate the quality of sign language interpreters by comparing them to a native language model? The latter question is particularly important given what happened at Nelson Mandela's memorial service[2]. In this memorial, the sign language interpreter seemed to be using correct signs but the signs together did not make sense. This raises the question: how do we verify whether a given sign language utterance is meaningful even when it is composed of meaningful signs arranged in a non-meaningful way?

[2] `http://www.youtube.com/watch?v=X-DxGoIVUWo`

Chapter 8

Gesture stroke detection

Content

This chapter presents a method for automatic gesture stroke detection, the problem of segmenting and identifying meaningful gesture units. The method uses classifiers trained on visual features extracted from videos based on feedback and interaction with the user. The chapter also studies the role of speech features as extra features in gesture stroke detection. Our results show that *a*) the best scores are achieved using visual cues, *b*) acoustic cues do not contribute to performance more than visual cues alone, and *c*) acoustic cues alone can, to some degree, predict where strokes occur.

Based on

B. G. Gebre, P. Wittenburg and P. Lenkiewicz (2012). “Towards automatic gesture stroke detection”. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC12)*, pages 231-235, European Language Resources Association (ELRA).

Keywords

Gesture stroke, videos, speech, preparation, hold, retraction, gesture phases

8.1 Introduction

The task of segmenting and annotating an observation sequence arises in many disciplines including gesture studies. One main preprocessing task in gesture studies is the annotation of gesture strokes. This task involves identifying and marking out the meaningful parts of body movements from video recordings. It can be likened to text tokenization, which is the process of breaking a stream of text into characters, words, phrases, or other meaningful elements called tokens [Fagan *et al.*, 1991; Carrier *et al.*, 2011]. It can also be likened to speech segmentation, which is the process of identifying the boundaries between words or phonemes in spoken languages [Waibel *et al.*, 1989; Graves *et al.*, 2013].

Currently, gesture stroke detection is carried out by manually going through video frames and marking out the start and end times of each stroke. This manual process is labor-intensive, time-consuming and non-scalable. Therefore, there is a growing need to solve the problem using more automatic approaches.

From a machine-learning point of view, gesture stroke detection is a classification or sequence labeling problem. Each frame from the video stream (or a vector of visual features extracted from it) is an observation and the whole video stream or a section of it is an observation sequence. The task is then to label each frame as 1 or 0, indicating whether it is a part of a stroke or not.

This study is different from other gesture recognition studies. Many other gesture recognition studies focus on classifying a set of a priori known gestures [Wu and Huang, 1999; Mitra and Acharya, 2007; Bevilacqua *et al.*, 2010]. In our study, we focus on the high level task of classifying gesture phases (distinguishing the relevant from the non-relevant movements) without attempting to identify the meaning of the gestures. Other approaches do not make such an explicit distinction (i.e. a distinction between the meaning of gestures and whether the gestures are meaningful to begin with).

This study is also different from other gesture recognition studies because we consider the role of speech in gesture stroke detection. Considering speech in gesture stroke detection is very important given that in natural settings, gestures rarely occur in isolation (i.e. when people speak, they usually gesture [Kendon, 1980; Kita, 2014]). In this spirit, we raise two questions: *a*) does including acoustic cues to visual cues significantly improve gesture stroke detection, *b*) can acoustic cues alone be used to detect where strokes occur? To answer these questions, we run experiments using manually annotated data and different supervised machine learning algorithms. Our results show that a) acoustic cues do not contribute to performance more than visual cues alone, and b) acoustic cues alone can, to some degree, predict where strokes occur. The rest of the chapter gives more details.

8.2 Gesture stroke

The gesture stroke is the most important message-carrying phase of the series of body movements that make people while speaking. The body movements usually include hand and face movements. The relevant questions for automatic gesture stroke detection are: *a*) what is a gesture? *b*) where does a gesture start and end? *c*) what are the phases in a gesture? *d*) which one is the stroke?

The literature of gesture studies does not give completely consistent answers to the above questions [Kendon, 1980, 1972; Kita *et al.*, 1998; Bressem and Ladewig, 2011]. However, the most prominent view is that a gesture unit consists of one or more gesture phrases and each gesture phrase consists of different phases [Kendon, 1980]. The gesture unit is defined as the period of time between successive rests of the hands; it begins the moment the hands begin to move from rest position and ends when they have reached a rest position again.

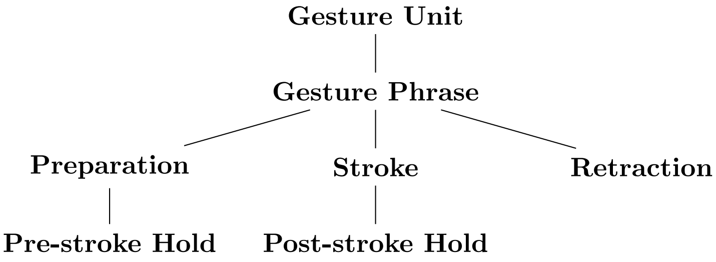


Figure 8.1: Gesture Phases
[Kendon, 1980, 1972]

Figure 8.1 shows the different phases in a gesture unit. A gesture unit consists of one or more gesture phrases and each gesture phrase consists of phases that are called preparation, pre-stroke hold, stroke, post-stroke hold and retraction. Except for strokes, which are obligatory, the rest of the phases in a gesture phrase are optional. McNeill [1992b] defines the five gesture phases as follows:

Preparation

The preparation is the movement of the hands away from their rest position to a position in gesture space where the stroke begins. Gesture space is the space in front of the speaker (see figure 8.2).

Pre-stroke hold

The pre-stroke hold is the position and hand posture reached at the end of the preparation, usually held briefly until the stroke begins. This phase is more likely to co-occur with discourse connectors; it is a period in which the gesture waits for speech to establish cohesion so that the stroke co-occurs with the co-expressive portion of speech [Kita, 1990].

Stroke

The stroke is the peak of effort in the gesture. It is in this phase that the meaning of the gesture is co-expressed with speech. It is typically performed in the central gesture space bounded roughly by the waist, shoulders, and arms (see figure 8.2).

Post-stroke hold

The post-stroke hold is the final position and posture of the hand reached at the end of the stroke, usually held briefly until the retraction begins. Its function is to temporally extend a single movement stroke so that the stroke and the post stroke hold together will synchronize with the co-expressive portion of speech [Kita, 1990].

Retraction

The retraction is the return movement of the hands to a rest position at the end of post-stroke hold or stroke phase.

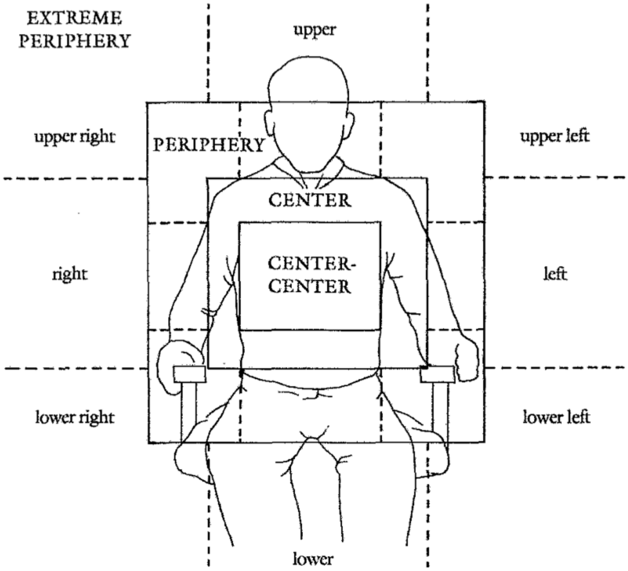


Figure 8.2: Typical gesture space of an adult speaker.
[McNeill, 1992b]

For the purpose of this study, any hand/face movement is classified into two classes: strokes and non-strokes. The non-stroke gesture phases include the preparation, hold, retraction and any other body movements excluding the strokes.

8.3 Our stroke detection method

Our approach to detecting gesture strokes involves three steps: *a*) detect the face and hands of the individual in the video *b*) extract visual features (shapes, movements, locations of hands/face) and audio features (MFCC, LPC, energy) *c*) learn a binary classifier to distinguish between strokes and non-strokes.

8.3.1 Face and hand detection

We use skin color to detect the hands and face [Vezhnevets *et al.*, 2003; Phung *et al.*, 2005]. Using skin color to detect hands/face has advantages and challenges. The advantages are that it is invariant to scale and orientation and it is easy to compute. The challenges are that *a*) perfect skin color ranges for one individual do not necessarily apply to another (diversity of skin colors) and *b*) distracting objects in the video may have the same color as the hands/face (ambiguity).

To overcome the first challenge, we did explicit manual selection of skin color HSV ranges for each individual video. This is done by selecting a representative skin color region from the first frame of the video and selecting the HSV ranges between which the skin color lies. To support the process of finding the right skin color ranges, visual feedback and sliders are provided that can be adjusted until skin color regions are clearly separated from background.

The alternative to manual skin color range selection is developing parametric or non-parametric distributions of skin color and non-skin color using training data. But this turned out to be less effective. Building a skin color model offline for all human skin colors is not only more complex (e.g. hard to find representative data) but also less accurate when applied on any particular individual video. However, models built online for a given video initialized by input from user achieve qualitatively higher performance at no more cost than the initialization and adjustment of skin color ranges.

To overcome the ambiguity problem of skin color ranges between skin color and other distracting objects, we applied dilation/erosion operations and constraint rules to remove objects that have skin color but have unexpected sizes. This approach does not solve all ambiguity problems. For example, as can be seen from figure 8.3, the chair that the person is sitting on has virtually the same color as the hands and face of the person.

8.3.2 Feature extraction

We extract features from both video and audio. The visual features encode posture of the upper body, locations of hands and face and movements. The audio features include MFCCs, energy and LPC.

Visual features

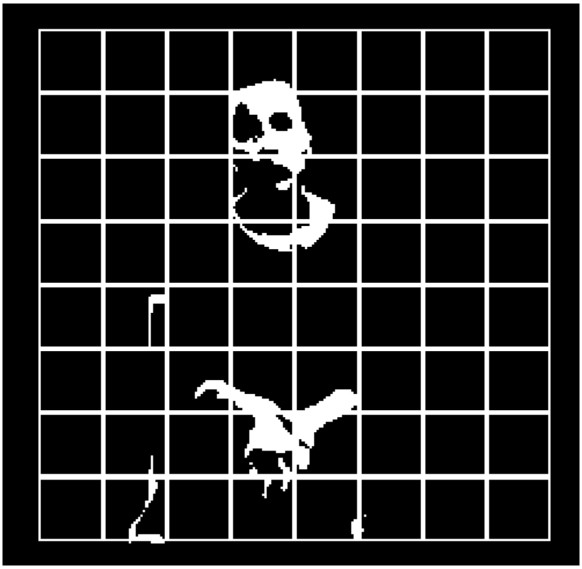


Figure 8.3: Location grid and skin color: each grid cell in the grid is a square whose side is half of the height of the face. The white regions of the picture show skin color and are obtained using HSV color ranges. Both the size of the grid and HSV skin color ranges are interactively selected by the user.

We encode and extract the shapes, locations and movements of skin-colored regions. To encode the shapes of skin-colored regions in the video, we use the Hu set of seven invariant moments ($H_1 - H_7$) [Hu, 1962], calculated from the gesture space of the speaker - the region bounded by the external lines of the grid shown in figure 8.3. The values of the seven Hu moments capture shapes and arrangements of the foreground objects (in our case, skin color regions) and are among the most widely used features in human activity recognition [Davis and Bobick, 1997; Bradski and Davis, 2002]. They offer invariance to scale, translation, rotation and skew [Hu, 1962].

To encode body locations of the speaker, we use grids of $8 * 8$ with the face used as a reference. The location and size of the face is determined by the user and is used to calculate the position and scale of the grid as shown in figure 8.3. Each side of every cell in the grid is half of the height of the face. A cell is assigned 1 if more than 20 percent of the area is covered by skin, otherwise, it will be assigned 0. The values in the cells are changed into a single row vector of size 64 by concatenating one row after another, forming a location vector.

To encode body movements, the location vector in the current frame is compared with respect to that in the previous frame. By subtracting the previous location vector from current location vector (pairwise element subtraction), we get a motion vector. Note that the location vectors are obtained from the grid cells as described in the previous paragraph.

Velocity and acceleration of the hands are not directly represented in the features. But we can assume that movement vectors indirectly encode velocity. Kita [1990] notes that acceleration (and deceleration) of the hands are good indicators of strokes, although a downward retraction may have bigger acceleration.

Audio features

We extract different audio features using a toolkit called yaafe [Mathieu *et al.*, 2010]. These features are MFCCs and their derivates, LPCs, energy, loudness and zero crossing rates.

Mel-frequency cepstral coefficients (MFCCs) are commonly used in various speech-related tasks (speech recognition, speaker recognition, speaker diarization, etc.) [Davis and Mermelstein, 1980]. We used the 13 Mel-frequency cepstral coefficients (MFCCs) along with their first order and second order derivates for a total of 39 features.

Linear Predictive Coding (LPC) coefficients of a speech signal represent each speech sample as a linear combination of previous samples. These prediction coefficients characterize the formants of the speech signal [Makhoul, 1975]. In our experiments, we used three coefficients.

Energy is the root mean square of the sum of the squares of the samples in a given frame. Loudness [Moore *et al.*, 1997] and zero crossing rates are also used as features. The loudness of a sound is a perceptual measure of the effect of the energy content of sound on the ear. The 24 loudness coefficients are the energy in each Bark band [Zwicker, 1961], normalized by the overall sum [Moore *et al.*, 1997].

The zero-crossing rate is the rate of sign-changes along a signal, i.e., the rate at which the signal changes from positive to negative or back.

8.3.3 Classification

We use three different supervised machine learning algorithms: random forest, logistic regression and support vector machines [Pedregosa *et al.*, 2011].

8.4 Experiments

8.4.1 Datasets

We conducted our experiments on three videos taken from The Language Archive[1] at the Max Planck Institute for Psycholinguistics. Each video has a single person

[1] https://corpus1.mpi.nl/

speaking and gesturing and has been annotated for gesture strokes. Table 8.1 shows the details of each video. The details are extracted from manually annotated data.

Table 8.1: Details of experiment dataset

Name	Length	Strokes	Fraction	Mean ± STD
ITCS	3.63	61	0.18	0.63 ± 0.24
sub49	5.25	129	0.14	0.34 ± 0.24
sub50	8.08	278	0.25	0.43 ± 0.24

Length = Video length in minutes
Strokes = Total number of gesture strokes
Fraction = Fraction of stroke time over video length
Mean ± STD = Mean and STD of stroke durations (in seconds)

8.4.2 Evaluation

We evaluate the performance of our system in terms of Area Under the Curve (AUC) of Receiver Operating Characteristic (ROC) graphs [Fawcett, 2006]. Evaluating classifiers using AUC scores have advantages over other methods like F1-scores. First, AUC of ROC curves is insensitive to changes in class distribution (if the proportion of positive to negative instances changes in a test set, the ROC curves will not change). Second, AUC does not depend on a single cut-off point above which the target variable is part of the positive class; instead, AUC evaluates at all cut-off points, giving better insight into how well the classifier is able to separate two classes. Because the reliability of AUC is brought into question [Lobo *et al.*, 2008], we also evaluate our system using precision, recall and F1-scores.

The three videos are evaluated separately using video features, audio features and both audio and video features. Because, the setting is supervised machine learning and the class label distribution is unbalanced, we perform 10-fold stratified cross validation (stratified means the folds produced preserve the percentage of samples for each class). No separate development set was used for parameter tuning. We use default values of the learning algorithms from the scikit–learn library [Pedregosa *et al.*, 2011]. The basic unit of evaluation is the video frame and the features are extracted from the current frame and neighboring frames (four preceding and four following frames).

8.5 Results and discussion

On 10-fold stratified cross-validation, random forest achieves the best mean AUC score of 0.96 for ITCS data using video features alone (see figure 8.4); when audio features are included, the AUC score drops to 0.95. We can observe that random forest does not benefit from including audio features (see table 8.2), whereas logistic regression benefits from having audio features (see table 8.3). For example, for the ICTS data, we can observe from figure 8.5 that the mean AUC score increases from 0.84 to 0.87 when audio features are added to video features. However, the best score of logistic regression (0.87) is much less than that of random forest (0.96).

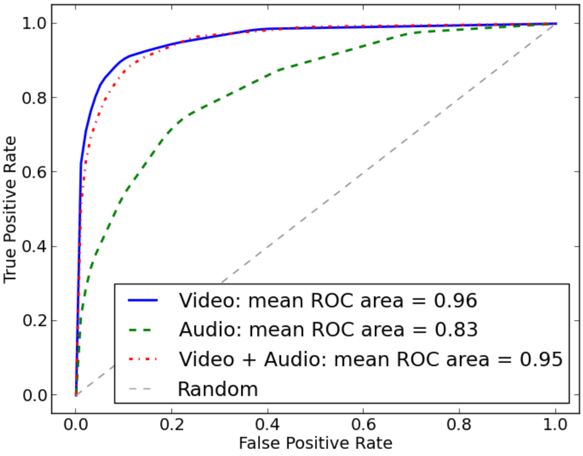


Figure 8.4: Random forest classifier: mean AUC scores on stratified 10-fold cross-validation for ICTS data using video, audio and both video and audio features.

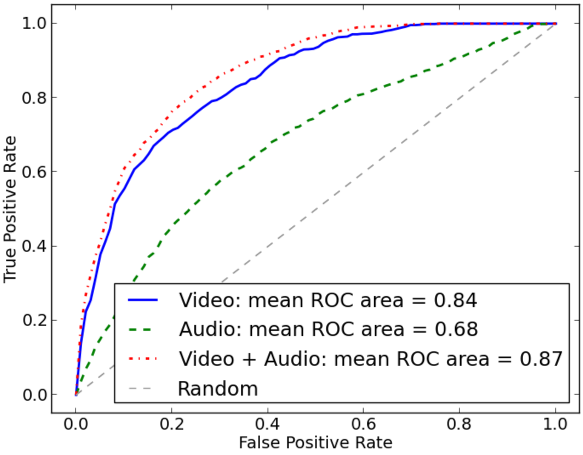


Figure 8.5: Logistic regression classifier: mean AUC scores on stratified 10-fold cross-validation for ICTS data using video, audio and both video and audio features.

Table 8.2: Scores for a random forest classifier (10 trees): precision, recall, F1 and AUC scores

Data	**Features**	**P**	**R**	**F1**	**AUC**
ITCS	Video	0.75	**0.61**	**0.67**	**0.96**
	Audio	0.65	0.43	0.52	0.83
	Video + Audio	**0.79**	0.51	0.62	0.95
sub49	Video	0.71	**0.51**	**0.60**	**0.95**
	Audio	0.60	0.39	0.47	0.81
	Video + Audio	0.71	0.44	0.54	0.93
sub50	Video	**0.67**	0.45	0.54	0.84
	Audio	0.60	0.47	0.53	0.81
	Video + Audio	0.64	**0.52**	**0.57**	**0.85**
Baseline F1 (random classifier) = 0.28, 0.21, 0.33					

Table 8.3: Scores for a logistic regression classifier (L1 penalty): precision, recall, F1 and AUC scores

Data	**Features**	**P**	**R**	**F1**	**AUC**
ITCS	Video	0.41	0.69	0.52	0.84
	Audio	0.26	0.63	0.37	0.68
	Video + Audio	**0.42**	**0.73**	**0.54**	**0.87**
sub49	Video	0.33	0.72	0.45	0.82
	Audio	0.23	0.66	0.35	0.72
	Video + Audio	0.33	**0.76**	**0.46**	**0.85**
sub50	Video	0.41	0.61	0.49	0.71
	Audio	0.41	0.69	0.52	0.77
	Video + Audio	**0.44**	**0.73**	**0.55**	**0.82**
Baseline F1 (random classifier) = 0.28, 0.21, 0.33					

From the scores in tables 8.2 and 8.3, we can also observe that stroke detection can be performed using acoustic cues alone (much better than chance) but the resulting scores are much less than scores resulting from using visual cues.

8.6 Conclusions and future work

The study in this chapter proposed an adaptive gesture stroke detection algorithm that takes user involvement into consideration. The user is involved in developing a skin color model applicable to a particular video. The skin color model is used to detect the face and hands of a person in a video. Based on skin color detection, three feature types (location, movement and shape features of face/hands) are extracted. These visual features are then augmented with standard audio features.

Our experimental results show that *a*) stroke detection using visual cues performs the best (up to 0.67 F1), *b*) stroke detection using both visual and acoustic cues does no better than stroke detection using visual cues alone, and *c*) stroke detection using acoustic cues alone performs much better than chance. The second result puts doubt as to whether speech carries more information about where strokes occur than is available in the visual cues. The third result suggests that speech carries information about where strokes occur but not as much as gesture.

Future work should examine the extent to which human subjects can predict where strokes occur based only on speech, based only on video frames and based on both speech and frames. This will shed new light on the redundancy and complementarity of speech and video frames in the task of stroke annotation. Future work should also experiment with applying features learned through unsupervised learning techniques. This will perhaps increase accuracy of automatic gesture stroke detection. Note that, in chapter 7, we have shown that unsupervised feature learning techniques give excellent performance for sign language identification.

Chapter 9

Conclusions

Content

This chapter concludes the thesis with a summary of the contributions made in the previous chapters and suggestions for future work. It also answers the two research questions raised in the introduction chapter.

Keywords

Speaker diarization, signer diarization, sign language identification, gesture stroke detection, primitive recognizers, adaptive recognizers

9.1 Introduction

We started this thesis with three high-level observations: *a*) our capacity to record, collect and store video data is growing much faster than our capacity to make use of it, *b*) our machines cannot fully understand content in videos, and *c*) our current process of making videos machine readable (i.e. manual annotation) is non-scalable, unlikely to ever match the scale of big data. After giving this general context to the thesis, we focused on four gesture-related problems: *speaker diarization*, *signer diarization*, *sign language identification* and *gesture stroke detection*. All of the four problems are types of gesture recognition, where given a video, we wish to detect and classify gestures: *a*) according to who produced them (speaker and signer diarization), *b*) according to the sign language of the signer (sign language identification), *c*) according to whether the movement is meaningful (gesture stroke detection). Underlying these problems are two research questions.

Research question 1:

> *How can a machine recognize gestures in diverse environments?*

Answer

> We addressed this research question by designing and developing "primitive" recognizers, leaving out the identification of the environment to human beings or to another primitive recognizer. Primitive recognizers are those that do one thing and do it well, but which when combined become more complex pattern recognizers. A good analogy for recognizers are unix commands, which are mostly simple but when combined, become general and useful tools. For example, to perform gesture recognition the "unix" way, we need a recognizer for human detection, a recognizer to segment out individuals and a gesture recognizer for individuals. In this spirit, we developed four recognizers: an active speaker recognizer, an active signer recognizer, a sign language recognizer, and a gesture stroke recognizer. Note that the philosophy for these recognizers came from the AVATeCH[1] project, which aims at developing many such audio and video recognizers. Our recognizers, though considered primitive, solve difficult pattern recognition problems. For example, all gestures involve movements but not all movements are gestures. So, how do we know which movements are gestures? Because it is impossible to fully qualitatively describe the patterns to recognize (in this example, gesture vs. other movements), our solution approach depends on statistical learning from well-annotated data, which by itself leads to the following research question.

Research question 2:

> *How can a machine effectively use data to learn to recognize gestures?*

[1] `https://tla.mpi.nl/projects_info/avatech/`

Answer

> We addressed this research question by designing adaptive recognizers. Adaptive recognizers are those that are trained off-line but that can also be adapted to a given set of conditions. The philosophy for the adaptation is that a recognizer designed to give the best average performance in a variety of scenarios is usually less accurate for a particular scene than a recognizer tailored to the characteristics of that scene. In the adaptive spirit, we developed the active speaker recognizer (it adapts a UBM, trained initially on all audio, to each speaker based on speech samples co-occurring with their gestures), the sign language recognizer (it learns features in an unsupervised way but the importance of the discriminative features is adapted to the given task of sign language identification using a small set of training data), and the gesture stroke recognizer (skin-color is selected by the user during system initialization and features are extracted to apply to the given video only).

The two research questions we raised above are related to tasks that are effortless for humans but difficult for machines. How do humans recognize patterns (e.g. gestures) in diverse environments? How do they learn from experience? Do they have many recognizers that are primitive and adaptive? Can we build intelligent machines by emulating humans? The latter question has been raised and treated in a book by Jeff Hawkins [Hawkins and Blakeslee, 2007]. The book tries to answer the following two questions. What are the operating principles of the neocortex? How can we build intelligent machines based on these principles?

Simply speaking, the first question is answered by saying that “the neocortex is a memory system, not a computer system”. The neocortex has six working principles: *i*) it learns online from streaming data *ii*) it has a hierarchy of memory regions (self-similar memory regions) *iii*) it stores a sequence memory (for inference and for motor behavior) *iv*) it has sparse distributed representations (few neurons are active and most are inactive) *v*) all regions are sensory and motor (learns a sensory-motor model of the world) *vi*) attention (has an ability to attend to various parts of information in time and space). Jeff Hawkins claims that these six principles are both necessary and sufficient for biological and machine intelligence. Based on these principles, he proposed an online machine learning system called Hierarchical Temporal Memory (HTM)[2].

How do our primitive and adaptive recognizers relate to the six working principles of the neocortex? The concepts of primitive and adaptive recognizers are related to working principles *i* and *ii*. More specifically, our primitive recognizers are related to a hierarchy of self-similar memory regions. Depending on their input, these memory regions are assigned different levels (otherwise, the memory regions are very similar). If the input to the memory region is raw information from sensors (through receptive fields), then that memory region is a primitive recognizer. If the input to the memory region is output from another memory region, then it is a

[2] `http://numenta.org/resources/HTM_CorticalLearningAlgorithms.pdf`

higher level memory region (a more complex recognizer). A complex recognizer in our case is a combination (cascade) of primitive recognizers.

Our adaptive recognizers are related to the first working principle, which states that the neocortex is an online learning system that continuously learns from streaming data. Our adaptive recognizers take in latest information (in chunks) to improve performance and this adaptation is related to the concept of online learning (online learning is a form of adaptation).

In summary, the primitive and adaptive design approach is a powerful strategy to deal with complexity. As presented above, it is also grounded in the working principles of the neocortex, the best example of an intelligent system. This primitive and adaptive design approach has been the basis of the following contributions.

9.2 Summary: speaker diarization

Contribution highlights

We presented a novel hypothesis that claims that *the gesturer is the speaker* and showed the well-foundedness of the hypothesis by presenting evidence from the literature of speech-gesture synchrony studies. The evidence includes the observations that gestures occur mainly during speaking, fluency affects gesturing (more fluency, more gestures), the congenitally blind also gesture and delayed auditory feedback does not interrupt speech-gesture synchrony.

Capitalizing on the above hypothesis, we designed and developed two speaker diarization algorithms based on: *a*) detection and tracking of corner features (optical flow), and *b*) motion history images. The latter algorithm, which we designed to be probabilistic, is more efficient and we showed it to be suitable for online speaker diarization.

The two diarization algorithms have two assumptions: *a*) any motion that is not brief and not isolated is a gesture, and *b*) speech is always accompanied by gesture. These assumptions do not always hold (i.e. brief motions can be gestures, long motions can be non-gestures and people can speak without gesturing). Despite this, our speaker diarization using only gesture performs much better than random (as a speaker is more likely to produce gestures while speaking than while listening). To take into account the cases in which the assumptions do not hold, we use speech in conjunction with gesture and solve speaker diarization in a novel way.

We treat speaker diarization as a continuous speaker identification problem after developing speaker models from speech samples co-occurring with gestures (the presence of gesture indicates the presence of speech and the location of gesture indicates the identity of the speaker). Accordingly, we proposed a novel speaker diarization system that works as follows: a UBM is first trained on all speech samples and the UBM is adapted to each speaker using speech samples co-occurring with their gestures. This adaptation gives us speaker models, which we then use to perform continuous speaker identification.

The continuous speaker identification (i.e. diarization) gives us speech samples for each speaker, which we then use to create better speaker models by adapting the UBM once again. This process of adaptation and diarization is then repeated until we are satisfied with the results or until we see no improvements. With 3 iterations, our tests on 4.24 hours of the AMI meeting data show that our approach makes DER score improvements of 19% on speech-only segments and 4% on all segments including silence (the comparison is with the AMI system, which is a diarization system based on agglomerative clustering).

In summary, compared to previous multimodal diarization systems, our diarization system has better accuracy, is faster (avoids agglomerative clustering) and is more flexible (controllable trade-off between computation and accuracy, can easily incorporate prior knowledge of the number of speakers, speaker models, etc).

Future work

Our gesture detection model can be enriched to model and fuse various types of information, such as visual focus of attention of speakers (listeners tend to look at the active speaker) and lip movements (this information is not always available but can be used whenever available). While enriching the model can be useful, it is also important to note that it comes at the cost of computation. Note that our gesture detection model, which is based on Motion History Images, has the advantage of being computationally minimal.

In our conversation dynamics model, individual speaking patterns are modeled the same way, but we may gain benefit from modeling each speaker's speaking patterns separately. We may also gain benefit from modeling the relationships and interactions between participants involved in a conversation. In the latter case, research in turn-taking may prove useful [Sacks *et al.*, 1974].

Our idea of using gestures in speaker diarization offers new opportunities to deal with overlapped speech, which still presents problems to traditional diarization approaches. Overlapped speech can be identified based on detection of gestures that are overlapping in time but that are spatially separate. In our current model, overlapped speech cannot be detected, because the most likely speaker approach always forces a choice between speakers. However, this can be changed by making decisions based on a speaking probability threshold, assigning speaker status to a person whenever the probability for speaking exceeds the threshold.

Our research has a direct impact on video conference technologies, where gestures can be used as cues to determine who is speaking and use that information to zoom in on the speaker. Using gesture cues, speaker models can then be developed for each participant. The speaker models can later be used for speaker identification, speaker diarization, speaker adaptation, speech recognition and automatic minute-taking.

9.3 Summary: signer diarization

Contribution highlights

We identified and studied signer diarization as an important problem. Our work motivated signer diarization by drawing similarities with speaker diarization, which is a dedicated discipline of research in spoken language processing.

Our solution approach to signer diarization is also similar to our approach to gesture-based speaker diarization. The difference is that movement is a necessary part of signing, whereas it is optional in speaking. Our previous hypothesis that *the gesturer is the speaker* holds here too by changing it to *the gesturer is the signer*. Accordingly, we proposed two signer diarization algorithms based on: *a*) detection and tracking of corner features (optical flow), and *b*) motion history images. Note that these are the same algorithms we developed for gesture-based speaker diarization.

The challenge in signer diarization is that not all body movements constitute signing activity (even though all signs involve movements). Our first algorithm (the algorithm based on optical flow) tries to overcome the challenge by removing short and isolated movements (at the cost of missed signs). Our second algorithm (the probabilistic algorithm based on motion history images) tries to overcome the challenge by using Gamma distributions to model signing and non-signing activity (the parameter values of the Gamma distributions are trained on manually annotated data). The advantage of the two algorithms is that they are language-independent as they do not look at the meaning of the movements.

Future work

More preprocessing: body motion is an inexpensive source of information and as such can be used as a baseline for signer diarization. But, not all body motions are signing activity. A signing activity detector (in a manner similar to a speech activity detector) may need to be applied as preprocessing to remove non-signing segments. Such a detector can be trained on annotated data using features extracted from body posture and head orientations.

A richer model: in our proposed model, each person has an independent model of signing and only one person is assumed to be signing at a time. But one can enrich the model by adding in extra parameters (e.g. to model the interactions of signers and interlocutors) and extra information (e.g. to model the fact that interlocutors look at the active signer). In signing communication, interlocutors need to look at the signer to be part of the conversation. This is an important cue to use for signer diarization. To make use of this cue, we need to develop a gaze detector for each signer and be able to combine the gaze detections of each signer to determine the signer being gazed at.

9.4 Summary: sign language identification

Contribution highlights

Previous research on language identification focused only on written and spoken languages. In this thesis, we identified and studied sign language identification as an important and challenging pattern recognition problem. We discussed several challenges in sign language identification arising from three sources – differences between signers (individuals have their own unique physical and signing characteristics), iconicity in sign languages (sign languages tend to be more iconic, and hence more similar) and diversity in video recording conditions (computer vision issues).

To overcome these challenges and still identify sign languages, we proposed machine learning solutions using two types of features: *a*) linguistically motivated features, and *b*) features learned through unsupervised techniques. With the first solution, three types of features (hand shapes, movements and locations), each motivated by the phonemes of sign language, are extracted. Using these features, we performed signer-independent classification of two sign languages (British and Greek sign languages) based on video samples of at least 10 seconds. We obtained an accuracy of 78%.

The linguistically motivated solution relies on the detection and localization of the face and hands of the signer and for this purpose, we use skin color. Because skin color is different across individuals and recording conditions, our skin detection depended on manual selection of skin color ranges, which we found to be tedious and non-scalable. We, therefore, opted for sign language identification using unsupervised feature learning techniques.

With the unsupervised solution, we showed how K-means and sparse autoencoder can be used to learn feature maps from videos of sign languages (using many small patches of $15 * 15$ and $15 * 15 * 2$ pixels). Through convolution and pooling, we also showed the use of these feature maps in classifier feature extraction. Finally, we showed the impact on accuracy of varying the number of feature maps (using both 2D and 3D feature maps).

The unsupervised solution, despite being more computationally intensive, is fully automatic (uses raw video pixels alone) and it performs better than the linguistically motivated solution. In a classification task of six sign languages involving 30 different signers, it achieved the best average accuracy of 84% (leave-one-signer-out cross-validation). This score is achieved using 500 K-means features extracted over video lengths of about 0.5 seconds.

Future work

Our sign language identification method should be further studied and evaluated in a context of: *a*) many more sign languages, *b*) many signers, and *c*) diverse environment (e.g. various video backgrounds). It is important to realize that a confounding factor in sign language identification may be that signers of the same sign language may share physical features. Theoretically, this problem can be dealt with by using multilingual signers. However, it will be difficult to find enough signers who are fluent in all the combinations of sign languages.

Our unsupervised feature learning based on sparse autoencoder used a single hidden layer of features (one hidden layer in a neural network), but it is worth looking into deeper layers to study the hierarchical composition of features and to gain insight into differences and similarities between various sign languages. Such a study can help us to develop sign language typology. This will show that fully automatic and unsupervised techniques are useful not only for practical applications but also for scientific study of sign languages (this is very important because sign languages are under-resourced).

A psycholinguistic experiment should be done to discover the extent to which humans (with and without knowledge of sign language) can learn to identify sign languages. In addition to the scientific interest in such an experiment, the outcome can serve as a benchmark for the evaluation of machine identification of sign languages. Note that a similar experiment has been done for spoken language identification [Muthusamy *et al.*, 1994b].

9.5 Summary: gesture stroke detection

Contribution highlights

We proposed an adaptive gesture stroke detection algorithm that takes user involvement into consideration. The user draws a rounded box around the face of the person in the first frame of the video and a skin-color model is developed using the distribution of colors in the box. The skin color model is used to detect the face and hands in subsequent frames of the video. Classification features are then extracted in frame regions where the skin color is detected. These visual features encode location, shape and movement features.

We also examined the role of acoustic cues in gesture stroke detection. We used various types of speech features (MFCCs, LPCs, energy, loudness and zero crossing rates). We showed that *a*) the best scores are achieved using visual cues, *b*) stroke detection using both visual and acoustic cues does no better than stroke detection using visual cues alone, and *c*) stroke detection using acoustic cues alone performs much better than chance. The second result suggests that speech does not carry more information about strokes than is available in the visual cues. The third result suggests that speech carries some information about where strokes occur, but not as much as visual cues.

Future work

Unsupervised feature learning techniques, which we showed to be effective in learning features for sign language identification, can also be used to learn features for gesture stroke detection.

The gesture stroke phase is preceded by optional gesture phases (preparation and pre-stroke hold) and it is followed by optional gesture phases (post-stroke hold and retraction). In our experiments, we modeled these phases as non-stroke gesture phases, but modeling them independently may contribute to the accuracy of stroke detection.

It is important to perform experiments to determine the upper limit of accuracy of gesture stroke detection as performed by humans, i.e. the extent to which human subjects can predict where strokes occur based only on speech, based only on gesture and based on both speech and gesture. The conclusion from such an experiment will shed light on the redundancy and complementarity of speech and gesture in the task of stroke detection.

9.6 Putting it all together

Gesture is an important source of information during communication in spoken and signed languages. Recognizing it helps us solve many human-related video content understanding problems. In this thesis, we demonstrated innovative application of it in the tasks of *speaker diarization*, *signer diarization*, *sign language identification*, and *gesture-stroke detection*. To perform each task, we developed *primitive* and *adaptive* recognizers as part of the AVATecH project[3]. In the design and development of these recognizers, machine learning played a central role. Future work should continue the development and refinement of many such recognizers in order to handle the complexity of video content understanding. We imagine a world, where a toolset of recognizers is easily available for applications requiring video content understanding.

[3] `https://tla.mpi.nl/projects_info/avatech/`

Bibliography

E. Adelson, C. Anderson, J. Bergen, P. Burt and J. Ogden (**1984**). “Pyramid methods in image processing”. *RCA engineer*, 29(6):33–41. 16

M. A. R. Ahad (**2013**). *Motion History Images for Action Recognition and Understanding*. Springer. 39, 48

J. Ajmera, H. Bourlard, I. Lapidot and I. McCowan (**2002**). “Unknown-multiple speaker clustering using HMM”. In “Proceedings of INTERSPEECH”, . 51

J. Ajmera and C. Wooters (**2003**). “A robust speaker clustering algorithm”. In “2003 IEEE Workshop on Automatic Speech Recognition and Understanding”, pages 411–416. IEEE. 23

S. Akram, J. Beskow and H. Kjellstrom (**2012**). “Visual recognition of isolated swedish sign language signs”. *arXiv preprint arXiv:1211.3901*. 29

X. Anguera (**2007**). *Robust speaker diarization for meetings*. Ph.D. thesis. 24, 57

X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland and O. Vinyals (**2012**). “Speaker diarization: A review of recent research”. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2):356–370. 13, 29, 39, 48, 51, 57

X. Anguera, C. Wooters and J. Hernando (**2006**). “Friends and enemies: a novel initialization for speaker diarization”. In “Proceedings of INTERSPEECH”, . 24

T. Baldwin and M. Lui (**2010**). “Language identification: The long and the short of the matter”. In “Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics”, pages 229–237. Association for Computational Linguistics. 75

B. Bauer, H. Hienz and K.-F. Kraiss (**2000**). “Video-based continuous sign language recognition using statistical methods”. In “Proceedings of International Conference on Pattern Recognition”, volume 2, pages 463–466. IEEE. 29

O. Ben-Harush, I. Lapidot and H. Guterman (**2012**). “Initialization of iterative-based speaker diarization systems for telephone conversations”. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2):414–425. 24

F. Bevilacqua, B. Zamborlin, A. Sypniewski, N. Schnell, F. Guédy and N. Rasamimanana (**2010**). “Continuous realtime gesture following and recognition”.

In "Gesture in embodied communication and human-computer interaction", pages 73–84. Springer. 89

J. Bouguet (**1999**). "Pyramidal implementation of the Lucas-Kanade feature tracker: Description of the algorithm, OpenCV documentation". *Santa Clara, CA: Intel Corp., Microprocessor Research Labs.* 16

J. Bouguet (**2001**). "Pyramidal implementation of the affine Lucas-Kanade feature tracker: Description of the algorithm". *Intel Corporation.* 21, 32, 47

G. Bradski (**2000**). "The OpenCV Library". *Dr. Dobb's Journal of Software Tools.* 16, 32

G. Bradski and A. Kaehler (**2008**). *Learning OpenCV: Computer vision with the OpenCV library.* O'Reilly Media, Incorporated. 66

G. R. Bradski and J. W. Davis (**2002**). "Motion segmentation and pose recognition with motion history gradients". *Machine Vision and Applications*, 13(3):174–184. 39, 48, 93

L. Breiman (**2001**). "Random forests". *Machine learning*, 45(1):5–32. 66, 67

J. Bressem and S. H. Ladewig (**2011**). "Rethinking gesture phases: Articulatory features of gestural movement?" *Semiotica*, 2011(184):53–91. 90

B. Butterworth and G. Beattie (**1978**). "Gesture and silence as indicators of planning in speech". *Recent advances in the psychology of language: Formal and experimental approaches*, 4:247–360. 14

B. Butterworth and U. Hadar (**1989**). "Gesture, speech, and computational stages: A reply to McNeill." 14

N. Campbell and N. Suzuki (**2006**). "Working with very sparse data to detect speaker and listener participation in a meetings corpus". In "Workshop Programme", volume 10, page 1. 14

J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal *et al.* (**2006**). "The AMI meeting corpus: A pre-announcement". *Machine Learning for Multimodal Interaction*, pages 28–39. 17, 43, 56

J. Carrier, A. B. Carus, W. F. Cote, J. Dowd, K. Del La Femina, A. Frankel, W. V. Han, L. Lapshina, B. Rechea, A. Santisteban *et al.* (**2011**). "System and method for tokenization of text using classifier models". US Patent 7,937,263. 89

R. Caruana and A. Niculescu-Mizil (**2006**). "An empirical comparison of supervised learning algorithms". In "Proceedings of the 23rd international conference on Machine learning", pages 161–168. ACM. 66

T. Chen, C. Huang, E. Chang and J. Wang (**2001**). "Automatic accent identification using Gaussian mixture models". In "IEEE Workshop on Automatic Speech Recognition and Understanding", pages 343–346. 75

G. Choueiter, G. Zweig and P. Nguyen (**2008**). “An empirical study of automatic accent classification”. In “IEEE International Conference on Acoustics, Speech and Signal Processing”, pages 4265–4268. IEEE. 75

A. Coates and A. Y. Ng (**2012**). “Learning feature representations with k-means”. In “Neural Networks: Tricks of the Trade”, pages 561–580. Springer. 78

A. Coates, A. Y. Ng and H. Lee (**2011**). “An analysis of single-layer networks in unsupervised feature learning”. In “International Conference on Artificial Intelligence and Statistics”, pages 215–223. 7, 75, 77

J. Coates and R. Sutton-Spence (**2001**). “Turn-taking patterns in deaf conversation”. *Journal of Sociolinguistics*, 5(4):507–529. 30

H. Cooper, E. Ong, N. Pugeault and R. Bowden (**2012**a). “Sign language recognition using sub-units”. *Journal of Machine Learning Research*, 13:2205–2231. 63, 65, 75

H. Cooper, E.-J. Ong, N. Pugeault and R. Bowden (**2012**b). “Sign language recognition using sub-units”. *Journal of Machine Learning Research*, 13:2205–2231. 29, 30

O. Crasborn (**2006**). *Nonmanual structures in sign languages*, volume 8, pages 668–672. Elsevier, Oxford. ISBN 0-08-044299-4. 76

M. Cristani, A. Pesarin, A. Vinciarelli, M. Crocco and V. Murino (**2011**). “Look at who’s talking: Voice activity detection by automated gesture analysis”. In “Workshop on Interactive Human Behavior Analysis in Open or Public Spaces”, . 22

N. Dalal and B. Triggs (**2005**). “Histograms of oriented gradients for human detection”. In “IEEE Conference on Computer Vision and Pattern Recognition”, volume 1, pages 886–893. IEEE. 15

T. Darrell, G. Gordon, M. Harville and J. Woodfill (**2000**). “Integrated person tracking using stereo, color, and pattern detection”. *International Journal of Computer Vision*, 37(2):175–185. 31

J. W. Davis and A. F. Bobick (**1997**). “The representation and recognition of human movement using temporal templates”. In “IEEE Conference on Computer Vision and Pattern Recognition”, pages 928–934. IEEE. 39, 40, 48, 93

S. Davis and P. Mermelstein (**1980**). “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences”. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(4):357–366. 52, 94

A. P. Dempster, N. M. Laird, D. B. Rubin *et al.* (**1977**). “Maximum likelihood from incomplete data via the EM algorithm”. *Journal of the Royal statistical Society*, 39(1):1–38. 55

T. Dunning (**1994**). *Statistical identification of language.* Computing Research Laboratory, New Mexico State University. 63, 75

E. Efthimiou, S. Fotinea, C. Vogler, T. Hanke, J. Glauert, R. Bowden, A. Braffort, C. Collet, P. Maragos and J. Segouat (**2009**). "Sign language recognition, generation, and modelling: a research effort with applications in deaf communication". *Universal Access in Human-Computer Interaction. Addressing Diversity*, pages 21–30. 67

J. L. Fagan, M. D. Gunther, P. D. Over, G. Passon, C. C. Tsao, A. Zamora and E. M. Zamora (**1991**). "Method for language-independent text tokenization using a character categorization". US Patent 4,991,094. 89

T. Fawcett (**2006**). "An introduction to ROC analysis". *Pattern recognition letters*, 27(8):861–874. 95

P. Feyereisen and J. de Lannoy (**1991**). *Gestures and speech: Psychological investigations*. Cambridge University Press. 13, 14

J. Fiscus, J. Ajot and J. Garofolo (**2008**). "The rich transcription 2007 meeting recognition evaluation". *Multimodal Technologies for Perception of Humans*, pages 373–389. 13

G. Friedland, H. Hung and C. Yeo (**2009**). "Multi-modal speaker diarization of real-world meetings using compressed-domain video features". In "IEEE International Conference on Acoustics, Speech and Signal Processing", pages 4069–4072. ISSN 1520-6149. 24, 48, 51

G. Friedland, A. Janin, D. Imseng, X. Anguera, L. Gottlieb, M. Huijbregts, M. Knox and O. Vinyals (**2012**). "The ICSI RT-09 speaker diarization system". *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2):371–381. 20, 21, 45, 51

G. Friedland and O. Vinyals (**2008**). "Live speaker identification in conversations". In "Proceedings of the 16th ACM international conference on Multimedia", pages 1017–1018. ACM. 39

G. Garau and H. Bourlard (**2010**). "Using audio and visual cues for speaker diarisation initialisation". In "ICASSP Proceedings", pages 4942–4945. IEEE. 24

J. Gauvain and C.-H. Lee (**1994**). "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains". *Speech and Audio Processing, IEEE Transactions on*, 2(2):291–298. ISSN 1063-6676. 55

D. M. Gavrila (**1999**). "The visual analysis of human movement: A survey". *Computer vision and image understanding*, 73(1):82–98. 63, 75

B. G. Gebre, O. Crasborn, P. Wittenburg, S. Drude and T. Heskes (**2014**a). "Unsupervised feature learning for visual sign language identification". In "Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics", pages 370–376. Association for Computational Linguistics, Baltimore, Maryland.
URL `http://www.aclweb.org/anthology/P/P14/P14-2061` 9

B. G. Gebre, P. Wittenburg, S. Drude, M. Huijbregts and T. Heskes (**2014**b). “Speaker diarization using gesture and speech”. In “Interspeech 2014: 15th Annual Conference of the International Speech Communication Association”, . 9

B. G. Gebre, P. Wittenburg and T. Heskes (**2013**a). “Automatic signer diarization - the mover is the signer approach”. In “2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)”, pages 283–287. 8

B. G. Gebre, P. Wittenburg and T. Heskes (**2013**b). “The gesturer is the speaker”. In “2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)”, pages 3751–3755. ISSN 1520-6149. 8

B. G. Gebre, P. Wittenburg, T. Heskes and S. Drude (**2014**c). “Motion history images for online speaker/signer diarization”. In “Proceedings of the 2014 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)”, IEEE. 9

B. G. Gebre, P. Wittenburg and P. Lenkiewicz (**2012**). “Towards automatic gesture stroke detection”. In “Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC’12)”, European Language Resources Association (ELRA), Istanbul, Turkey. ISBN 978-2-9517408-7-7. 10

B. G. Gebre, P. W. Wittenburg and T. Heskes (**2013**c). “Automatic sign language identification”. In “2013 IEEE International Conference on Image Processing (ICIP)”, pages 2626–2630. 9

X. Glorot, A. Bordes and Y. Bengio (**2011**). “Deep sparse rectifier networks”. In “Proceedings of the 14th International Conference on Artificial Intelligence and Statistics. JMLR W&CP Volume”, volume 15, pages 315–323. 78

A. Graves, A.-r. Mohamed and G. Hinton (**2013**). “Speech recognition with deep recurrent neural networks”. In “2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)”, pages 6645–6649. IEEE. 89

J. Hawkins and S. Blakeslee (**2007**). *On intelligence*. Macmillan. 102

G. E. Hinton and R. R. Salakhutdinov (**2006**). “Reducing the dimensionality of data with neural networks”. *Science*, 313(5786):504–507. 75, 77

M. Hu (**1962**). “Visual pattern recognition by moment invariants”. *Information Theory, IRE Transactions on*, 8(2):179–187. 65, 93

M. Huijbregts (**2008**). *Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled*. PhD thesis, University of Twente. Publisher: Centre for Telematics and Information Technology University of Twente, publisherlocation: Enschede, ISSN: 1381-3617, ISBN: 978-90-365-2712-5, Numberofpages: 172. 58

M. Huijbregts, D. van Leeuwen and C. Wooters (**2012**). “Speaker diarization error analysis using oracle components”. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2):393–403. 21, 35, 51

H. Hung and S. Ba (2010). “Speech/non-speech detection in meetings from automatically extracted low resolution visual features”. In “ICASSP Proceedings”, pages 830–833. 24

A. Hyvärinen and E. Oja (2000). “Independent component analysis: algorithms and applications”. *Neural networks*, 13(4):411–430. 77

D. Imseng and G. Friedland (2009). “Robust speaker diarization for short speech recordings”. In “IEEE Workshop on Automatic Speech Recognition & Understanding”, pages 432–437. IEEE. 24

D. Imseng and G. Friedland (2010). “An adaptive initialization method for speaker diarization based on prosodic features”. In “2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)”, pages 4946–4949. IEEE. 24

J. Iverson and S. Goldin-Meadow (1997). “What’s communication got to do with it? gesture in children blind from birth.” *Developmental Psychology*, 33(3):453. 15

J. Iverson, H. Tencer, J. Lany and S. Goldin-Meadow (2000). “The relation between gesture and speech in congenitally blind and sighted language-learners”. *Journal of nonverbal behavior*, 24(2):105–130. 15

B. Jiang, Y. Song, S. Wei, J.-H. Liu, I. V. McLoughlin and L.-R. Dai (2014). “Deep bottleneck features for spoken language identification”. *PloS one*, 9(7):e100795. 75

P. KaewTraKulPong and R. Bowden (2002). “An improved adaptive background mixture model for real-time tracking with shadow detection”. In “Video-Based Surveillance Systems”, pages 135–144. Springer. 40

A. Kendon (1972). “Some relationships between body motion and speech”. *Studies in dyadic communication*, 7:177. 90

A. Kendon (1980). “Gesticulation and speech: Two aspects of the process of utterance”. *The relationship of verbal and nonverbal communication*, 25:207–227. 14, 89, 90

S. Kita (1990). “The temporal relationship between gesture and speech: A study of Japanese-English bilinguals”. *MS, Department of Psychology, University of Chicago.* 90, 91, 94

S. Kita (2014). “Production of speech-accompanying”. *The Oxford Handbook of Language Production*, 48:451. 89

S. Kita, I. Van Gijn and H. Van der Hulst (1998). “Movement phases in signs and co-speech gestures, and their transcription by human coders”. *Gesture and sign language in human-computer interaction*, pages 23–35. 90

E. Klima and U. Bellugi (1979). *The signs of language.* Harvard University Press. 68

J. Kovac, P. Peer and F. Solina (2003). *Human skin color clustering for face detection*, volume 2. IEEE. 64

H. Lee, P. Pham, Y. Largman and A. Y. Ng (2009). “Unsupervised feature learning for audio classification using convolutional deep belief networks”. In “Advances in neural information processing systems”, pages 1096–1104. 7

W. J. Levelt, G. Richardson and W. La Heij (1985). “Pointing and voicing in deictic expressions”. *Journal of Memory and Language*, 24(2):133–164. 14

H. Li, B. Ma and C.-H. Lee (2007). “A vector space modeling approach to spoken language identification”. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(1):271–284. ISSN 1558-7916. 75

S. Liddell and R. Johnson (1989). *American sign language: The phonological base.* Gallaudet University Press, Washington. DC. 64

S. K. Liddell (1978). “Nonmanual signals and relative clauses in American Sign Language”. *Understanding language through sign language research*, pages 59–90. 31

J. M. Lobo, A. Jiménez-Valverde and R. Real (2008). “AUC: a misleading measure of the performance of predictive distribution models”. *Global ecology and Biogeography*, 17(2):145–151. 95

D. Lowe (2004). “Distinctive image features from scale-invariant keypoints”. *International journal of computer vision*, 60(2):91–110. 48

A. L. Maas, A. Y. Hannun and A. Y. Ng (2013). “Rectifier nonlinearities improve neural network acoustic models”. In “Proceedings of the ICML”, . 78

J. Makhoul (1975). “Linear prediction: A tutorial review”. *Proceedings of the IEEE*, 63(4):561–580. 94

K. Markov and S. Nakamura (2007). “Never-ending learning system for on-line speaker diarization”. In “IEEE Workshop on Automatic Speech Recognition & Understanding”, pages 699–704. IEEE. 39

B. Mathieu, S. Essid, T. Fillon, J. Prado and G. Richard (2010). “Yaafe, an easy to use and efficient audio feature extraction software”. In “11th ISMIR conference, Utrecht, Netherlands”, . 94

R. Mayherry and J. Jaques (2000). “Gesture production during stuttered speech: insights into the nature of gesture-speech integration”. *Language and gesture*, 2:199. 15

D. McNeill (1985). “So you think gestures are nonverbal?” *Psychological review*, 92(3):350. 14, 51

D. McNeill (1992a). *Hand and mind: What gestures reveal about thought.* University of Chicago Press. 14

D. McNeill (1992b). “Hand and mind: What gestures reveal about thought”. *University Of Chicago Press, IL.* 90, 91

D. McNeill (2005). *Gesture and thought.* University of Chicago Press. 14

S. Meignier and T. Merlin (**2010**). "LIUM spkdiarization: an open source toolkit for diarization". In "CMU SPUD Workshop", volume 2010. 39, 51

N. Mirghafori and C. Wooters (**2006**). "Nuts and flakes: A study of data characteristics in speaker diarization". In "ICASSP Proceedings", volume 1, pages I–I. IEEE. 19, 21, 33, 57

R. Mitkov (**2002**). *Anaphora resolution*, volume 134. Longman London. 39

S. Mitra and T. Acharya (**2007**). "Gesture recognition: A survey". *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(3):311–324. 89

B. Moore, B. Glasberg and T. Baer (**1997**). "A model for the prediction of thresholds, loudness, and partial loudness". *J. Audio Eng.* 94

P. Morrel-Samuels and R. M. Krauss (**1992**). "Word familiarity predicts temporal asynchrony of hand gestures and speech." *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 18(3):615. 14

Y. Muthusamy, E. Barnard and R. Cole (**1994**a). "Reviewing automatic language identification". *Signal Processing Magazine, IEEE*, 11(4):33–41. 63

Y. Muthusamy, N. Jain and R. Cole (**1994**b). "Perceptual benchmarks for automatic language identification". In "IEEE International Conference on Acoustics, Speech, and Signal Processing", volume i, pages I/333–I/336 vol.1. ISSN 1520-6149. 107

A. Noulas, G. Englebienne and B. Krose (**2012**). "Multimodal speaker diarization". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1):79–93. 24

A. Noulas and B. J. Krose (**2007**). "On-line multi-modal speaker diarization". In "Proceedings of the 9th international conference on Multimodal interfaces", pages 350–357. ACM. 39, 48

A. Özyürek, R. M. Willems, S. Kita and P. Hagoort (**2007**). "On-line integration of semantic information from speech and gesture: Insights from event-related brain potentials". *Journal of Cognitive Neuroscience*, 19(4):605–616. 14

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay (**2011**). "Scikit-learn: Machine learning in Python". *Journal of Machine Learning Research*, 12:2825–2830. 55, 66, 67, 80, 94, 95

P. Perniss, R. L. Thompson and G. Vigliocco (**2010**). "Iconicity as a general property of language: evidence from spoken and signed languages". *Frontiers in psychology*, 1. 76

S. Phung, A. Bouzerdoum Sr and D. Chai Sr (**2005**). "Skin segmentation using color pixel classification: analysis and comparison". *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(1):148–154. 64, 92

D. Reynolds and R. Rose (**1995**). "Robust text-independent speaker identification using Gaussian mixture speaker models". *IEEE Transactions on Speech and Audio Processing*, 3(1):72–83. 55, 59

D. A. Reynolds, T. F. Quatieri and R. B. Dunn (**2000**). "Speaker verification using adapted Gaussian mixture models". *Digital signal processing*, 10(1):19–41. 55, 56

A. E. Rosenberg, A. L. Gorin, Z. Liu and S. Parthasarathy (**2002**). "Unsupervised speaker segmentation of telephone conversations". In "Proceedings of INTERSPEECH", . 13

M. Rouvier, G. Dupuy, P. Gay, E. Khoury, T. Merlin and S. Meignier (**2013**). "An open-source state-of-the-art toolbox for broadcast news diarization". In "Proceedings of INTERSPEECH", . 39, 51

H. Sacks, E. A. Schegloff and G. Jefferson (**1974**). "A simplest systematics for the organization of turn-taking for conversation". *Language*, pages 696–735. 104

N. Seichepine, S. Essid, C. Févotte and O. Cappé (**2013**). "Soft nonnegative matrix co-factorizationwith application to multimodal speaker diarization". In "IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)", pages 3537–3541. IEEE. 48

E. Singer, P. Torres-Carrasquillo, T. Gleason, W. Campbell and D. Reynolds (**2003**). "Acoustic, phonetic, and discriminative approaches to automatic language identification". In "Proc. Eurospeech", volume 9. 63, 75

E. Singer, P. Torres-Carrasquillo, D. Reynolds, A. McCree, F. Richardson, N. Dehak and D. Sturim (**2012**). "The MITLL NIST LRE 2011 Language Recognition System". In "Odyssey 2012-The Speaker and Language Recognition Workshop", . 63, 75

H. Sloetjes and P. Wittenburg (**2008**). "Annotation by category: ELAN and ISO DCR". In "Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)", . 31

T. Starner and A. Pentland (**1997**). "Real-time american sign language recognition from video using hidden markov models". In "Motion-Based Recognition", pages 227–243. Springer. 63, 75

T. Starner, J. Weaver and A. Pentland (**1998**). "Real-time american sign language recognition using desk and wearable computer based video". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1371–1375. 63, 75

T. Stivers, N. J. Enfield, P. Brown, C. Englert, M. Hayashi, T. Heinemann, G. Hoymann, F. Rossano, J. P. De Ruiter, K.-E. Yoon *et al.* (**2009**). "Universals and cultural variation in turn-taking in conversation". In "Proceedings of the National Academy of Sciences", volume 106, pages 10587–10592. National Acad Sciences. 30

W. Stokoe (**2005**). "Sign language structure: An outline of the visual communication systems of the american deaf". *Journal of deaf studies and deaf education*, 10(1):3–37. 31, 63

S. Taub (**2001**). *Language from the body: iconicity and metaphor in American Sign Language*. Cambridge University Press, Cambridge. 76

C. Teixeira, I. Trancoso and A. Serralheiro (**1996**). "Accent identification". In "Proceedings of ICSLP", volume 3, pages 1784–1787. 75

J. Tetreault, D. Blanchard and A. Cahill (**2013**). "A report on the first native language identification shared task". *NAACL/HLT 2013*, page 48. 75

C. Tomasi and J. Shi (**1994**). "Good features to track". In "IEEE Conference on Computer Vision and Pattern Recognition", pages 593–600. 16, 21, 32, 47

P. Torres-Carrasquillo, E. Singer, M. Kohler, R. Greene, D. Reynolds and J. Deller Jr (**2002**). "Approaches to language identification using Gaussian mixture models and shifted delta cepstral features". In "Proceedings of ICSLP", volume 2, pages 33–36. 63

S. Tranter and D. A. Reynolds (**2004**). "Speaker diarisation for broadcast news". In "Odyssey04-The Speaker and Language Recognition Workshop", . 13

S. Tranter and D. A. Reynolds (**2006**). "An overview of automatic speaker diarization systems". *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1557–1565. 13, 29, 39, 51

H. Vajaria, S. Sarkar and R. Kasturi (**2008**). "Exploring co-occurence between speech and body movement for audio-guided video localization". *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1608–1617. 24

F. Vallet, S. Essid and J. Carrive (**2013**). "A multimodal approach to speaker diarization on TV talk-shows". 48

C. Valli and C. Lucas (**2001**). *Linguistics of American Sign Language Text: An Introduction*. Gallaudet University Press. 30

D. A. Van Leeuwen and M. Huijbregts (**2006**). "The AMI speaker diarization system for NIST RT06s meeting data". In "Machine Learning for Multimodal Interaction", pages 371–384. Springer. 58

C. Vaquero, O. Vinyals and G. Friedland (**2010**). "A hybrid approach to online speaker diarization". In "Proceedings of INTERSPEECH", pages 2638–2641. 39

V. Vezhnevets, V. Sazonov and A. Andreeva (**2003**). "A survey on pixel-based skin color detection techniques". In "Proc. Graphicon", volume 3, pages 85–92. Moscow, Russia. 64, 92

D. Vijayasenan and F. Valente (**2012**). "DiarTk: An open source toolkit for research in multistream speaker diarization and its application to meetings recordings". In "Proceedings of INTERSPEECH", . 39, 51

P. Viola and M. Jones (**2001**). "Rapid object detection using a boosted cascade of simple features". In "Proceedings of CVPR", volume 1, pages I–511. IEEE. 65, 66

P. Viola and M. J. Jones (**2004**). "Robust real-time face detection". *International journal of computer vision*, 57(2):137–154. 51

L. Von Ahn (**2009**). "Human computation". In "Proceedings of Design Automation", pages 418–419. IEEE. 5

C. de Vos (**2012**). *Sign-Spatiality in Kata Kolok: how a village sign language of Bali inscribes its signing space.* Ph.D. thesis, Max Planck Institute for Psycholinguistics. 32, 33, 44

A. Waibel, T. Hanazawa, G. Hinton, K. Shikano and K. J. Lang (**1989**). "Phoneme recognition using time-delay neural networks". *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(3):328–339. 89

C. Wooters, J. Fung, B. Peskin and X. Anguera (**2004**). "Towards robust speaker segmentation: The ICSI-SRI fall 2004 diarization system". In "RT-04F Workshop", volume 23. 23

C. Wooters and M. Huijbregts (**2008**). "The ICSI RT07s speaker diarization system". *Multimodal Technologies for Perception of Humans*, pages 509–519. 21, 23, 35, 51, 57

T. Wu, J. Duchateau, J.-P. Martens and D. Van Compernolle (**2010**). "Feature subset selection for improved native accent identification". *Speech Communication*, 52(2):83–98. 75

Y. Wu and T. Huang (**1999**). "Vision-based gesture recognition: A review". *Gesture-Based Communication in Human-Computer Interaction*, pages 103–115. 89

S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey *et al.* (**2006**). "The HTK book (for HTK version 3.4)". *Cambridge university engineering department*, 2(2):2–3. 52

S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev and P. Woodland (**1997**). *The HTK book*, volume 2. Entropic Cambridge Research Laboratory Cambridge. 52

M. Zampieri and B. G. Gebre (**2012**). "Automatic identification of language varieties: The case of Portuguese". In "Proceedings of KONVENS", pages 233–237. 75

L.-G. Zhang, Y. Chen, G. Fang, X. Chen and W. Gao (**2004**). "A vision-based sign language recognition system using tied-mixture density HMM". In "Proceedings of the 6th international conference on Multimodal interfaces", pages 198–204. 29

J. Zieren and K.-F. Kraiss (**2005**). "Robust person-independent visual sign language recognition". *Pattern Recognition and Image Analysis*, pages 333–355. 29

M. Zissman (**1996**). "Comparison of four approaches to automatic language identification of telephone speech". *IEEE Transactions on Speech and Audio Processing*, 4(1):31–44. 63, 75

E. Zwicker (**1961**). "Subdivision of the audible frequency range into critical bands (frequenzgruppen)". *The Journal of the Acoustical Society of America*, 33:248. 94

Summary

Data collection and analysis are important tasks in many areas of our life. Our capacity to collect data is growing much faster than our capacity to make sense of it. This is certainly the case with video data. With advances in device technology, it has become much easier for virtually anyone to record, collect and store video data. This ease has resulted in data volumes of a scale too big for any human to analyze manually. Can machines watch videos for us and tell us what is interesting? The goal of this thesis is to provide an answer to that question by advancing technologies applied in enriching certain types of video recordings - videos of people engaged in language use.

More specifically, the thesis focuses on solving four related problems: speaker diarization, signer diarization, sign language identification and gesture stroke detection. These problems are types of gesture recognition, where given a video, the goal is to detect and classify gestures: *a*) according to who produced them (*speaker and signer diarization*), *b*) according to the sign language of the signer (*sign language identification*), and *c*) according to whether the movement is meaningful (*gesture stroke detection*). Solving these problems has a wide range of applications such as document and information retrieval, machine translation and automatic minute taking systems. Given that machines don't have human-like eyes and brains, how do we solve these problems?

The thesis solves these problems using machine learning. Machine learning is the art and science of writing programs that learn to perform tasks based on examples. For example, how do we model the voice characteristics of speakers? With machine learning, we collect many speech samples for each speaker and develop mathematical models of the data, which we then use to make predictions on new data. The choice of mathematical models to use and the aspects of data (also called features) to consider are critical in making machine learning work in applications. Also critical is to determine which aspects of the problem are solvable by machine learning and whether there is enough training data for learning to take place.

The remainder of this summary describes the four problems studied in this thesis.

Speaker diarization

Extensive literature exists on speaker diarization, the task of determining *who spoke when* in an audio or video recording. Our contribution is that we proposed the use of gestures in speaker diarization and developed algorithms to exploit them. We hypothesized that *the gesturer is the speaker* and showed the well-foundedness of the hypothesis by presenting evidence from studies on the synchronization of gesture and speech (*see chapter 2*). We then proposed two speaker diarization algorithms based on: *a*) detection and tracking of corner features (*see chapter 2*), and *b*) motion history images (*see section 4.2*). The latter algorithm is more efficient and we showed it to be suitable for online settings (*see chapter 4*).

We also proposed another speaker diarization algorithm based on the exploitation of gesture and speech. The use of gesture enables the formulation of the diarization problem in a novel way. We treat speaker diarization as a speaker recognition problem after learning speaker models from speech samples co-occurring with gestures. We train a Gaussian Mixture model on all speech samples and create new models by adapting the model to each speaker using speech samples co-occurring with their gestures. For better performance, we then repeat speaker adaptation and diarization. This new approach has better accuracy, is faster (avoids agglomerative clustering) and is more flexible (better trade-off between computation and accuracy) than previous systems (*see chapter 5*).

Signer diarization

Signer diarization, the task of determining *who signed when*, has similar motivations and applications as speaker diarization except for the difference in modality. While there is significant literature on speaker diarization, very little exists on signer diarization. This thesis identifies signer diarization as an important problem and proposes a solution to it. Given the similarities between sign language and gesturing, our proposed solutions are similar to those presented for speaker diarization, i.e. based on: *a*) detection and tracking of corner features, and *b*) motion history images (*see chapters 3 and 4*).

Sign language identification

Language identification is the task of determining the identity of a language given utterances in the language. It is a basic preprocessing stage in document retrieval and machine translation systems. While previous work on language identification is only for written and spoken languages, this thesis proposes language identification solutions for signed languages. We proposed solutions based on *a*) linguistically motivated features (hand shapes, movements, locations), and *b*) features learned through unsupervised techniques (K-means and sparse autoencoder).

The first solution is based on the hypothesis that sign languages have varying distributions of phonemes (hand shapes, locations and movements) and that these differences in distribution can be used to identify sign languages. The challenge in this first solution is that it is non-trivial to detect and extract the phonemes from videos. Because of this, the second solution is proposed and here, the features are learned from raw video pixels. The place and degree of these feature activations are extracted through convolution and are then used to discriminate between sign languages. The first solution achieved an accuracy of 78% in a classification task involving two sign languages, whereas the second solution achieved 84% for six sign languages (*see chapters 6 and 7*).

Gesture stroke detection

Gesture stroke detection is one of the main preprocessing tasks in gesture studies. The task can be likened to speech segmentation or word tokenization. Our contribution is that we proposed an adaptive gesture stroke detection algorithm that takes user involvement into consideration. The user draws a box around the face of the person in the first frame of the video and a skin color model is developed using the distribution of colors in the box. The skin color model is used to detect the face and hands in subsequent frames of the video. Visual features are then extracted. These features encode hand shapes, movements and locations.

We also examined the role of acoustic cues in gesture stroke detection. We found that *a*) stroke detection using both visual and acoustic cues does no better than stroke detection using visual cues alone, and *b*) stroke detection using acoustic cues alone performs much better than chance. The first result suggests that speech does not carry more information about strokes than is available in the visual cues. The second result suggests that speech carries information about where strokes occur, but not as much as visual cues (*see chapter 8*).

Putting it all together

Gesture is an important source of information during communication in spoken and signed languages. This thesis demonstrated its application in solving several human-related video content understanding problems. We developed *primitive* and *adaptive* recognizers as part of the AVATecH project[4] (*see chapters 1 and 9*). In the design and development of these recognizers, machine learning played a central role. We recommend that many such recognizers be developed in order to manage the complexity of video content understanding. We imagine a world where a toolset of recognizers is easily available for applications requiring video content understanding.

[4] `https://tla.mpi.nl/projects_info/avatech/`

Samenvatting

Het verzamelen en analyseren van data is belangrijk in veel aspecten van ons leven. Onze capaciteit voor het verzamelen van data groeit veel sneller dan onze capaciteit voor het begrijpen van deze data. Dit is zeker het geval met video data. Dankzij technologische vooruitgang is het nu voor iedereen eenvoudig om video-opnames te maken, verzamelen en bewaren. Hierdoor ontstaan er data-verzamelingen die te groot zijn om nog door een mens geanalyseerd te kunnen worden. Kunnen computers video's voor ons bekijken en ons vertellen wat interessant is? Het doel van dit proefschrift is om deze vraag gedeeltelijk te beantwoorden door technologieën te ontwikkelen die toegepast kunnen worden op bepaalde soorten video-opnames: video's van mensen die taal gebruiken.

Specifiek bespreekt dit proefschrift vier gerelateerde problemen: *speaker diarization* (herkennen wie wanneer spreekt), *signer diarization* (herkennen wie wanneer gebaart), identificatie van gebarentaal en *gesture stroke detection* (het detecteren van het meest betekenisvolle gedeelte van een gebaar). Bij al deze problemen is het doel is om in een video gebaren te detecteren en classificeren *a*) aan de hand van wie ze heeft geproduceerd (*speaker diarization en signer diarization*) *b*) aan de hand van de gebarentaal die wordt gebruikt (*het identificeren van gebarentaal*) *c*) aan de hand van de mate waarin een beweging betekenis heeft (*gesture stroke detection*). Oplossingen voor deze problemen hebben verschillende applicaties, zoals *document retrieval*, *information retrieval*, automatische vertaling en automatisch notuleren. Aangezien computers geen menselijke ogen en hersenen hebben, is de vraag: hoe lossen we deze problemen op?

Dit proefschrift lost deze problemen op met gebruik van *machine learning* (automatisch leren). *Machine learning* is de kunst en wetenschap van het schrijven van programma's die zelf taken leren uitvoeren aan de hand van voorbeelden. Bijvoorbeeld, hoe kunnen we de eigenschappen van verschillende stemmen modelleren? Met *machine learning* verzamelen we een grote hoeveelheid segmenten van de spraak van elke spreker en ontwikkelen we op basis daarvan wiskundige modellen, die we vervolgens gebruiken om voorspellingen te maken aan de hand van nieuwe data. De keuze van het wiskundige model en de beslissing welke eigenschappen van de data (*features*) in het model worden meegenomen zijn cruciaal voor het functioneren van *machine learning.* Het is ook van groot belang om te bepalen weke onderdelen van het probleem met behulp van *machine learning* kunnen worden opgelost en of er genoeg data is om van te kunnen leren.

Het vervolg van deze samenvatting beschrijft de vier problemen die in dit proefschrift onderzocht worden.

Speaker diarization

Er bestaat een uitgebreide literatuur over *speaker diarization*, het bepalen *wie wanneer spreekt* in een geluids- of video-opname. Onze bijdrage hieraan is het idee om gebaren te gebruiken voor *speaker diarization*. We begonnen met de hypothese *de gebaarder is de spreker* en presenteerden bewijs voor deze hypothese, afkomstig van studies naar de synchronisatie van spraak en gebaren (*zie hoofdstuk 2*). Vervolgens stelden we twee algoritmen voor *speaker diarization* voor, gebaseerd op: *a*) het detecteren en tracken van *corner features* (*zie hoofdstuk 2*), en *b*) *motion history images* (*zie sectie 4.2*). Het laatstgenoemde algoritme is efficienter en we hebben laten zien dat het gebruikt kan worden voor *online* settings (*zie hoofdstuk 4*).

We hebben ook een algoritme voor *speaker diarization* ontwikkeld dat gebruik maakt van zowel gebaren als spraak. Het gebruik van gebaren maakt het mogelijk om het *diarization*-probleem op een nieuwe manier te formuleren. We behandelen *speaker diarization* als sprekerherkenning nadat modellen van de sprekers worden geleerd op basis van spraaksegmenten die samen met gebaren voorkomen. We trainen een Gaussian Mixture model op alle spraaksegmenten en creren nieuwe modellen door het model aan te passen aan elke spreker, gebruik makend van de segmenten die samen met gebaren voorkomen. Voor een beter resultaat herhalen we vervolgens de aanpassing aan de spreker en de *diarization*. Deze nieuwe aanpak resulteert in een betere nauwkeurigheid, snelheid (aangezien *agglomerative clustering* niet nodig is) en flexibiliteit (een betere balans tussen computatie en nauwkeurigheid) dan die van eerdere systemen (*zie hoofdstuk 5*).

Signer diarization

Signer diarization is het bepalen *wie wanneer gebaart* in gesprekken in gebarentaal. Het heeft toepassingen vergelijkbaar met die van *speaker diarization*. Ondanks de uitgebreide literatuur over *speaker diarization* is er nog nauwelijks onderzoek gedaan naar *signer diarization*. Dit proefschrift beschrijft *signer diarization* als een belangrijk probleem en stelt een oplossing voor. Aangezien gebarentaal overeenkomsten vertoont met gebaren tijdens spraak, zijn onze oplossingen vergelijkbaar met die voor *speaker diarization*. De oplossingen zijn gebaseerd op: *a*) het detecteren en tracken van *corner features*, en *b*) *motion history images* (*zie hoofdstuk 3 en 4*).

Identificatie van gebarentaal

Taalidentificatie is het bepalen van de identiteit van een taal aan de hand van uitingen in die taal. Dit is een taak die als eerste stap gebruikt wordt in systemen voor *document retrieval* en automatische vertaling. Eerder onderzoek naar taalidentificatie behandelde alleen geschreven en gesproken talen. In dit proefschrift hebben we taalidentificatie voor gebarentalen besproken. We stelden oplossingen voor gebaseerd op *a*) taalkundig gemotiveerde *features* (handvorm, beweging, locatie), en *b*) *features* die worden geleerd via *unsupervised learning* (*K-means* en *sparse autoencoder*)

De eerste oplossing is gebaseerd op de hypothese dat gebarentalen verschillende distributies hebben van fonemen (handvormen, beweging en locaties) en dat deze verschillen in distributie gebruikt kunnen worden om gebarentalen te identificeren. Het is echter niet eenvoudig om deze fonemen te detecteren in video-opnames. Daarom stelden we de tweede oplossing voor, waarbij de *features* direct geleerd worden op basis van video pixels. De locatie en mate van de activeringen van deze *features* worden geëxtraheerd met gebuik van convolutie en worden vervolgens gebruikt om onderscheid te maken tussen gebarentalen.

De eerste oplossing resulteerde in een nauwkeurigheid van 78% bij het classificeren van twee gebarentalen, terwijl de tweede oplossing een nauwkeurigheid van 84% had voor zes gebarentalen (*zie hoofdstuk 6 en 7*).

Gesture stroke detection

Gesture stroke detection (het detecteren van het betekenisvolle gedeelte van gebaren) is een van de voornaamste stappen in de voorbewerking van data voor onderzoek naar gebaren. De taak is vergelijkbaar met spraaksegmentatie en tokenisatie. Onze bijdrage is een adaptief *gesture stroke detection* algoritme waarbij input van de gebruiker wordt gevraagd. In het eerste frame van de video plaatst de gebruiker een kader rondom het gezicht van de persoon waarvan de gebaren gedetecteerd moeten worden. Gebaseerd op de distributie van kleuren binnen dit kader wordt een huidskleurmodel ontwikkeld. Dit huidskleurmodel wordt gebruikt om het gezicht en de handen te herkennen in de overige frames van de video. Vervolgens worden visuele *features* geëxtraheerd. Deze *features* betreffen handvorm, beweging en locatie.

We hebben ook de rol van akoestische informatie in *gesture stroke detection* onderzocht. We vonden dat *a*) detectie met behulp van visuele en akoestische informatie niet beter functioneert dan met alleen visuele informatie, en *b*) detectie met alleen akoestische informatie beter functioneert dan kans Het eerste resultaat suggereert dat spraak niet meer informatie bevat over gebaren dan beelden. Het tweede resultaat suggereert dat spraak informatie bevat over waar gebaren voorkomen, maar niet zoveel als beelden (*zie hoofdstuk 8*).

Conclusie

Gebaren bevatten belangrijke informatie tijdens communicatie in zowel gesproken taal als gebarentaal. Dit proefschrift heeft laten zien dat het herkennen van gebaren toegepast kan worden bij het oplossen van verschillende problemen die te maken hebben met het automatisch verwerken van videobeelden. We hebben primitieve en adaptieve *recognizers* ontwikkeld als onderdeel van het AVATecH project [5] (*zie hoofdstuk 1 en 9*). Bij het ontwerpen en ontwikkelen van deze *recognizers* heeft *machine learning* een belangrijke rol gespeeld. We bevelen het ontwikkelen van meer van zulke *recognizers* aan, om de complexiteit en hoeveelheid van videodata aan te kunnen. We stellen ons een wereld voor waarin een aanbod van zulke *recognizers* beschikbaar is voor alle applicaties waarbij videobeelden verwerkt worden.

[5] `https://tla.mpi.nl/projects_info/avatech/`

Publications

My Ph.D. research resulted in the following publications.

B. G. Gebre, P. Wittenburg, S. Drude, M. Huijbregts and T. Heskes (2014). “Speaker diarization using gesture and speech”. In *Interspeech 2014: 15th Annual Conference of the International Speech Communication Association.*

B. G. Gebre, O. Crasborn, P. Wittenburg, S. Drude and T. Heskes (2014). “Unsupervised feature learning for visual sign language identification”. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 370-376. Association for Computational Linguistics, Baltimore, Maryland. URL `http://www.aclweb.org/anthology/P/P14/P14-2061`

B. G. Gebre, P. Wittenburg, T. Heskes and S. Drude (2014). “Motion history images for online speaker/signer diarization”. In *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1537-1541, IEEE.

B. G. Gebre, M. Zampieri, P. Wittenburg and T. Heskes (2013). Improving native language identification with TF-IDF weighting”. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 216-223. Association for Computational Linguistics, Atlanta, Georgia.
URL `http://www.aclweb.org/anthology/W13-1728`

B. G. Gebre, P. W. Wittenburg and T. Heskes (2013). “Automatic sign language identification”. In *Proceedings of the 2013 IEEE International Conference on Image Processing (ICIP)*, pages 2626-2630, IEEE.

B. G. Gebre, P. Wittenburg and T. Heskes (2013). “Automatic signer diarization - the mover is the signer approach”. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 283-287.

B. G. Gebre, P. Wittenburg and T. Heskes (2013). “The gesturer is the speaker”. In *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3751–3755.

B. G. Gebre, P. Wittenburg and P. Lenkiewicz (2012). “Towards automatic gesture stroke detection”. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC12)*, pages 231-235, European Language Resources Association (ELRA).

P. Wittenburg, P. Lenkiewicz, E. Auer, A. Lenkiewicz, **B. G. Gebre** and S. Drude (2012). “AV processing in ehumanities – a paradigm shift”. In *2012 Digital Humanities Confer-*

ence, volume 2, pages 538-541.

M. Zampieri and **B. G. Gebre** (2014). “Varclass: An open-source language identification tool for language varieties”. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, European Language Resources Association (ELRA). URL `http://www.lrec-conf.org/proceedings/lrec2014/pdf/996_Paper.pdf`

M. Zampieri, **B. G. Gebre** and S. Diwersy (2013). “N-gram language models and POS distribution for the identification of spanish varieties”. In *Proceedings of TALN2013*, pages 580-587, Sable d’Olonne, France.

M. Zampieri, **B. G. Gebre** and S. Diwersy (2012). “Classifying pluricentric languages: Extending the monolingual model”. In *Proceedings of the Fourth Swedish Language Technology Conference (SLTC2012)*, pages 79-80.

M. Zampieri and **B. G. Gebre** (2012). “Automatic identification of language varieties: The case of portuguese”. In *Proceedings of KONVENS*, pages 233-237.

Lenkiewicz, P., Wittenburg, P., **Gebre, B. G.**, Lenkiewicz, A., Schreer, O., and Masneri, S. (2011). Application of video processing methods for linguistic research”. In *Proceedings of the 5th Language and Technology Conference (LTC 2011)*, pages 561-564.

Curriculum Vitae

Binyam Gebrekidan Gebre was born in Mekelle, Ethiopia on April 1, 1983. He went on a scholarship to Kellamino Special High School, a boarding school for high-achieving students and graduated in 2002 with *very great distinction.* For his undergraduate studies, he went on a scholarship to Mekelle Institute of Technology, where high-achieving students are admitted and in 2007, he earned a B.Sc. degree in Computer Science and Engineering with *very great distinction.* After graduating, he worked as a teacher at the same institute for one year. In 2008, he won an Erasmus Mundus Masters scholarship in Natural Language Processing and Human Language Technology, taught in two countries: France and the United Kingdom. In 2010, he obtained joint degrees, one with *mention très bien* from Université de Franche-Comté, and another with *distinction* from the University of Wolverhampton. His masters thesis is entitled *Part of speech tagging for Amharic.* In 2010, he started his PhD research on *Machine learning for gesture recognition from videos.* This research was funded by CLARA, a Marie Curie ITN. He currently works as a data scientist for the Rechenzentrum Garching (RZG), a computing center for the Max Planck Society (MPS) and the Max Planck Institute for Plasma Physics (IPP).

MPI Series in Psycholinguistics

1. The electrophysiology of speaking: Investigations on the time course of semantic, syntactic, and phonological processing. *Miranda van Turennout*
2. The role of the syllable in speech production: Evidence from lexical statistics, metalinguistics, masked priming, and electromagnetic midsagittal articulography. *Niels O. Schiller*
3. Lexical access in the production of ellipsis and pronouns. *Bernadette M. Schmitt*
4. The open-/closed-class distinction in spoken-word recognition. *Alette Haveman*
5. The acquisition of phonetic categories in young infants: A self-organising artificial neural network approach. *Kay Behnke*
6. Gesture and speech production. *Jan-Peter de Ruiter*
7. Comparative intonational phonology: English and German. *Esther Grabe*
8. Finiteness in adult and child German. *Ingeborg Lasser*
9. Language input for word discovery. *Joost van de Weijer*
10. Inherent complement verbs revisited: Towards an understanding of argument structure in Ewe. *James Essegbey*
11. Producing past and plural inflections. *Dirk Janssen*
12. Valence and transitivity in Saliba: An Oceanic language of Papua New Guinea. *Anna Margetts*
13. From speech to words. *Arie van der Lugt*
14. Simple and complex verbs in Jaminjung: A study of event categorisation in an Australian language. *Eva Schultze-Berndt*
15. Interpreting indefinites: An experimental study of children's language comprehension. *Irene Krämer*
16. Language-specific listening: The case of phonetic sequences. *Andrea Weber*
17. Moving eyes and naming objects. *Femke van der Meulen*
18. Analogy in morphology: The selection of linking elements in Dutch compounds. *Andrea Krott*
19. Morphology in speech comprehension. *Kerstin Mauth*
20. Morphological families in the mental lexicon. *Nivja H. de Jong*
21. Fixed expressions and the production of idioms. *Simone A. Sprenger*

22. The grammatical coding of postural semantics in Goemai (a West Chadic language of Nigeria). *Birgit Hellwig*
23. Paradigmatic structures in morphological processing: Computational and cross-linguistic experimental studies. *Fermín Moscoso del Prado Martín*
24. Contextual influences on spoken-word processing: An electro-physiological approach. *Daniëlle van den Brink*
25. Perceptual relevance of prevoicing in Dutch. *Petra M. van Alphen*
26. Syllables in speech production: Effects of syllable preparation and syllable frequency. *Joana Cholin*
27. Producing complex spoken numerals for time and space. *Marjolein Meeuwissen*
28. Morphology in auditory lexical processing: Sensitivity to fine phonetic detail and insensitivity to suffix reduction. *Rachèl J. J. K. Kemps*
29. At the same time...: The expression of simultaneity in learner varieties. *Barbara Schmiedtová*
30. A grammar of Jalonke argument structure. *Friederike Lüpke*
31. Agrammatic comprehension: An electrophysiological approach. *Marlies Wassenaar*
32. The structure and use of shape-based noun classes in Miraña (North West Amazon). *Frank Seifart*
33. Prosodically-conditioned detail in the recognition of spoken words. *Anne Pier Salverda*
34. Phonetic and lexical processing in a second language. *Mirjam Broersma*
35. Retrieving semantic and syntactic word properties. *Oliver Müller*
36. Lexically-guided perceptual learning in speech processing. *Frank Eisner*
37. Sensitivity to detailed acoustic information in word recognition. *Keren B. Shatzman*
38. The relationship between spoken word production and comprehension. *Rebecca Özdemir*
39. Disfluency: Interrupting speech and gesture. *Mandana Seyfeddinipur*
40. The acquisition of phonological structure: Distinguishing contrastive from non-contrastive variation. *Christiane Dietrich*
41. Cognitive cladistics and the relativity of spatial cognition. *Daniel B.M. Haun*
42. The acquisition of auditory categories. *Martijn Goudbeek*
43. Affix reduction in spoken Dutch. *Mark Pluymaekers*
44. Continuous-speech segmentation at the beginning of language acquisition: Electrophysiological evidence. *Valesca Kooijman*
45. Space and iconicity in German Sign Language (DGS). *Pamela Perniss*
46. On the production of morphologically complex words with special attention to effects of frequency. *Heidrun Bien*
47. Crosslinguistic influence in first and second languages: Convergence in speech and gesture. *Amanda Brown*

48. The acquisition of verb compounding in Mandarin Chinese. *Jidong Chen*
49. Phoneme inventories and patterns of speech sound perception. *Anita Wagner*
50. Lexical processing of morphologically complex words: An information-theoretical perspective. *Victor Kuperman*
51. A grammar of Savosavo, a Papuan language of the Solomon Islands. *Claudia Wegener*
52. Prosodic structure in speech production and perception. *Claudia Kuzla*
53. The acquisition of finiteness by Turkish learners of German and Turkish learners of French: Investigating knowledge of forms and functions in production and comprehension. *Sarah Schimke*
54. Studies on intonation and information structure in child and adult German. *Laura de Ruiter*
55. Processing the fine temporal structure of spoken words. *Eva Reinisch*
56. Semantics and (ir)regular inflection in morphological processing. *Wieke Tabak*
57. Processing strongly reduced forms in casual speech. *Susanne Brouwer*
58. Ambiguous pronoun resolution in L1 and L2 German and Dutch. *Miriam Ellert*
59. Lexical interactions in non-native speech comprehension: Evidence from electro-encephalography, eye-tracking, and functional magnetic resonance imaging. *Ian FitzPatrick*
60. Processing casual speech in native and non-native language. *Annelie Tuinman*
61. Split intransitivity in Rotokas, a Papuan language of Bougainville. *Stuart Robinson*
62. Evidentiality and intersubjectivity in Yurakaré: An interactional account. *Sonja Gipper*
63. The influence of information structure on language comprehension: A neurocognitive perspective. *Lin Wang*
64. The meaning and use of ideophones in Siwu. *Mark Dingemanse*
65. The role of acoustic detail and context in the comprehension of reduced pronunciation variants. *Marco van de Ven*
66. Speech reduction in spontaneous French and Spanish. *Francisco Torreira*
67. The relevance of early word recognition: Insights from the infant brain. *Caroline Junge*
68. Adjusting to different speakers: Extrinsic normalization in vowel perception. *Matthias J. Sjerps*
69. Structuring language : contributions to the neurocognition of syntax. *Katrien R. Segaert*
70. Infants' appreciation of others' mental states in prelinguistic communication : a second person approach to mindreading. *Birgit Knudsen*
71. Gaze behavior in face-to-face interaction. *Federico Rossano*
72. Sign-spatiality in Kata Kolok: how a village sign language of Bali inscribes its signing space. *Connie de Vos*

73. Who is talking? Behavioural and neural evidence for norm-based coding in voice identity learning. *Attila Andics*

74. Lexical processing of foreign-accented speech: Rapid and flexible adaptation. *Marijt Witteman*

75. The use of deictic versus representational gestures in infancy. *Daniel Puccini*

76. Territories of knowledge in Japanese conversation. *Kaoru Hayano*

77. Family and neighbourhood relations in the mental lexicon: A cross-language perspective. *Kimberley Mulder*

78. Contributions of executive control to individual differences in word production. *Zeshu Shao*

79. Hearing speech and seeing speech: Perceptual adjustments in auditory-visual processing. *Patrick van der Zande*

80. High pitches and thick voices: The role of language in space-pitch associations. *Sarah Dolscheid*

81. Seeing what's next: Processing and anticipating language referring to objects. *Joost Rommers*

82. Mental representation and processing of reduced words in casual speech. *Iris Hanique*

83. The many ways listeners adapt to reductions in casual speech. *Katja Poellmann*

84. Contrasting opposite polarity in Germanic and Romance languages: Verum focus and affirmative particles in native speakers and advanced L2 learners. *Giuseppina Turco*

85. Morphological processing in younger and older people: Evidence for flexible dual-route access. *Jana Reifegerste*

86. Semantic and syntactic constraints on the production of subject-verb agreement. *Alma Veenstra*

87. The acquisition of morphophonological alternations across languages. *Helen Buckler*

88. The evolutionary dynamics of motion event encoding. *Annemarie Verkerk*

89. Rediscovering a forgotten language. *Jiyoun Choi*

90. The road to native listening: Language-general perception, language-specific input. *Sho Tsuji*

91. Infants' understanding of communication as participants and observers. *Gudmundur Bjarki Thorgrímsson*

92. Information structure in Avatime. *Saskia van Putten*

93. Switch reference in Whitesands. *Jeremy Hammond*

94. Machine learning for gesture recognition from videos. *Binyam Gebrekidan Gebre*