UNIVERSITY OF LIÈGE

DOCTORAL THESIS

Computer vision systems for automatic analysis of face and eye images in specific applications of interpretation of facial expressions

---

Author:
Thomas HOYOUX

Supervisor:
Jacques G. VERLY

Examiners:
Olivier BARNICH
Justus H. PIATER
Steven LAUREYS
Marc VAN DROOGENBROECK
Louis WEHENKEL (*President*)

A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy

in the

Laboratory for Signal and Image Exploitation
Department of Electrical Engineering and Computer Science

September 2017

UNIVERSITY OF LIÈGE

DOCTORAL THESIS

Computer vision systems for automatic analysis of face and eye images in specific applications of interpretation of facial expressions

---

Author:
Thomas HOYOUX

Supervisor:
Jacques G. VERLY

Examiners:
Olivier BARNICH
Justus H. PIATER
Steven LAUREYS
Marc VAN DROOGENBROECK
Louis WEHENKEL (*President*)

A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy

in the

Laboratory for Signal and Image Exploitation
Department of Electrical Engineering and Computer Science

September 2017

Abstract

This thesis is about the computer vision-based automation of specific tasks of face perception, for specific applications where they are essential. These tasks, and the applications in which they are automated, deal with the interpretation of facial expressions.

Our first application of interest is the automatic recognition of sign language, as carried out via a chain of automatic systems that extract visual communication cues from the image of a signer, transcribe these visual cues to an intermediary semantic notation, and translate this semantic notation to a comprehensible text in a spoken language. For use within the visual cue extraction part of such a system chain, we propose a computer vision system that automatically extracts facial communication cues from the image of a signer, based on a pre-existing facial landmark point tracking method and its various robust refinements. With this system, our contribution notably lies in the fruitful use of this tracking method and its refinements within a sign language recognition system chain. We consider the facial communication cues extracted by our system as facial expressions with a specific interpretation useful to this application.

Our second application of interest is the objective assessment of visual pursuit in patients with a disorder of consciousness. In the clinical practice, this delicate assessment is done by a clinician who manually moves a handheld mirror in front of the patient's face while simultaneously estimating the patient's ability to track this visual stimulus. This clinical setup is appropriate, but the assessment outcome was shown to be sensitive to the clinician's subjectivity. For use with a head-mounted device, we propose a computer vision system that attaches itself to the clinical procedure without disrupting it, and automatically estimates, in an objective way, the patient's ability to perform visual pursuit. Our system, combined with the use of a head-mounted device, therefore takes the form of an assisting technology for the clinician. It is based on the tracking of the patient's pupil and the mirror moved by the clinician, and the comparison of the obtained trajectories. All methods used within our system are simple yet specific instantiations of general methods, for the objective assessment of visual pursuit. We consider the visual pursuit ability extracted by our system as a facial expression with a specific interpretation useful to this application.

To some extent, our third application of interest is the general-purpose automatic recognition of facial expression codes in a muscle-based taxonomic coding system. We do not actually provide any new computer vision system for this application. Instead, we consider a supervised classification problem relevant to this application, and we empirically compare the performance of two general classification approaches for solving this problem, namely hierarchical classification and standard classification ("flat" classification, in this comparative context). We also compare these approaches for solving a classification problem relevant to 3D shape recognition, as well as artificial classification problems we generate in a simulation framework of our design. Our contribution lies in the general theoretical conclusions we reach from our empirical study of hierarchical vs. flat classification, which are of interest for properly using hierarchical classification in vision-based recognition problems, for example for an application of facial expression recognition.

Résumé

Cette thèse traite de l'automatisation par vision par ordinateur de tâches spécifiques de perception du visage, pour des applications spécifiques dans lesquelles elles sont essentielles. Ces tâches, et les applications où elles sont automatisées, portent sur l'interprétation des expressions faciales.

Notre première application d'intérêt est la reconnaissance automatique de la langue des signes, par une chaîne de systèmes qui extraient les indices de communication visuelle de l'image d'un interprète, transcrivent ces indices en une notation sémantique intermédiaire, et traduisent cette notation en un texte compréhensible dans une langue parlée. Pour son utilisation dans une telle chaîne, nous proposons un système de vision par ordinateur qui extrait automatiquement les indices de communication faciale de l'image d'un interprète, sur base d'une méthode pré-existante pour le suivi de points de repère du visage. Notre contribution réside surtout dans l'utilisation fructueuse de cette méthode de suivi et de ses raffinements, au sein d'une chaîne de systèmes pour la reconnaissance de la langue des signes. Nous considérons les indices de communication faciale extraits par notre système comme des expressions faciales avec une interprétation spécifique, qui est utile à la reconnaissance automatique de la langue des signes.

Notre deuxième application d'intérêt est l'évaluation objective de la poursuite visuelle chez les patients atteints d'un trouble de la conscience. Dans la pratique clinique, cette évaluation délicate est effectuée par un clinicien qui déplace manuellement un miroir de poche devant le visage du patient tout en estimant sa capacité à suivre ce stimulus. Cette procédure est appropriée, mais cependant sensible à la subjectivité du clinicien. Pour une utilisation jointe avec un appareil monté sur tête, nous proposons un système de vision par ordinateur qui s'attache à cette procédure sans la perturber et estime automatiquement, de manière objective, la capacité du patient à effectuer une poursuite visuelle. Joint à l'appareil monté sur tête, notre système prend la forme d'une technologie d'assistance pour le clinicien, basée sur le suivi de la pupille du patient, le suivi du miroir déplacé par le clinicien, et la comparaison des trajectoires obtenues. Nous considérons la capacité de poursuite visuelle extraite par notre système comme une expression faciale avec une interprétation spécifique, utile à l'évaluation objective de la poursuite visuelle.

Dans une certaine mesure, notre troisième application d'intérêt est la reconnaissance automatique des expressions faciales, telles que décrites par un système de codage basé sur les muscles du visage. Nous considérons un problème de classification lié à cette application, et nous comparons empiriquement deux approches générales pour le résoudre: la classification hiérarchique et la classification standard (dite "plate" dans ce contexte de comparaison). Nous comparons aussi ces approches pour résoudre un problème de classification lié à la reconnaissance de formes 3D, ainsi que des problèmes générés dans un environnement de simulation de notre conception. Notre contribution ne consiste pas en un nouveau système de vision par ordinateur, mais réside dans les conclusions théoriques que nous tirons de notre étude empirique. Celles-ci sont d'intérêt pour l'utilisation avantageuse de la classification hiérarchique dans des problèmes de reconnaissance basée sur la vision, par exemple pour une application de reconnaissance des expressions faciales.

Acknowledgements

I would like to express my gratitude to Jacques Verly and Justus Piater, for the opportunity they gave me to do research in computer vision. I thank them both for their support and trust in my work, and their ever so valuable insight on how to properly explain and promote my findings. I thank Jacques Verly in particular for his thorough review of the present manuscript.

I am grateful to Steven Laureys, with whom every interaction was not only pleasant but profoundly useful to help me highlight the pertinence of the part of my work related to patients with a disorder of consciousness.

I would like to thank all members of the jury, including Louis Wehenkel, Marc Van Droogenbroeck, and Olivier Barnich, for having agreed to read and evaluate the present manuscript. I repeat my thanks to Olivier in particular, who became a close friend and one of my life-coaches, especially when I was writing this dissertation.

I thank the Walloon Region, the European Community, and the universities of Liège and Innsbruck, for funding the projects and grants in which I participated. On a related note, I thank Jacques Verly, Justus Piater, and Steven Laureys for the financial support they gave me through these projects and grants.

Many thanks to my former colleagues and fellow doctoral students for their relevant feedback, their invaluable moral support, and the nice time we spent together in and out of the workplace. These include but are not limited to Philippe R., Quentin, Mike, Wei, Nicolas, Jérôme, Clémentine, Thomas L., Sarah, Renaud, Minh, David, Philippe L., and Angel at the university of Liège. Rodrigo, Simon, Boris, Heiko, Sandor, Emre, Hanchen, and Damien at the university of Innsbruck. Oscar, Jens, Christoph, Philippe D., and Yannick at the university of Aachen. Jaume and Gregorio at the research and innovation center of Catalonia. Ellen and Onno at the university of Nijmegen. My special thanks go to Rodrigo and Sarah, for their very significant help in the realization of my doctoral work, and also to Philippe R., Quentin, Mike, Wei, and Nicolas, for their friendship.

Thanks to my other friends, Salion, Arnaud, Stéphane, Erdal, Pierre, Anne-Lise, and Thomas J., who mostly had no clue what I was doing with my life during the doctoral work but were ever supportive. Likewise, I dearly thank my family, including my parents Dany and Pierre, my grandmother Josette, my grandfather René H., and my sister Caroline, who has been herself through the dire challenge of trying to bring forth a contribution in a scientific research field.

Finally and foremost, I thank Livia, for so many reasons.

Contents

List of Figures

List of Tables

To my late grandparents Gigi and René R.

Chapter 1

Introduction

The face is a central sense organ complex that gathers most of the sensory inputs from the environment, to transmit them to the brain where they are processed [1]. In addition to its major role in sensing for its owner, the face is a powerful visual source of information for an external observer. On the one hand, each person's face is unique, notably through the anatomical structure of the skull. On the other hand, the contractions of muscles beneath the face skin and other neighboring muscles allow each person's face to move and deform in a variety of ways (Fig. 1.1). These static and dynamic variations of the face can be perceived through mere macroscopic visual observation, and further be given an interpretation, a cognitive process known as face perception [2]. People use face perception thoroughly and effectively in their everyday social interactions. Recognizing a person's identity, age, or gender, determining the presence of an emotional state or a physiological activity, detecting communication attempts and impending intents, all of those and more are face perception tasks, and somewhat basic ones from a human perspective. Indeed, face perception is typically carried out by humans without much effort [3], for the purpose of laying some ground knowledge toward inferring more complex semantics, such as understanding the course of a conversation, or assessing specific skills or elaborated personality traits in a person. However simple and casual it feels to humans, extensive and various areas of their brain are involved in face perception [4]. A part of the temporal lobe called the fusiform gyrus is even hypothesized to be specialized in face recognition, and is sometimes referred to as the fusiform face area for that reason [5]. The expertise of the human brain in face perception also develops at a very young age. For instance, provided that a normal social and perceptual experience is given in their early developmental stages, infants are able to recognize familiar people by their face before one year old [6]. At about the same age, they are also able to recognize basic emotions and infer simple intents in people by watching their face, e.g., they can recognize anger and further interpret it as a potential threat [7]. Because of the intuition and scientific evidence that face perception plays such an important role in human society, developing automatic systems able to perform face perception tasks as well as humans do became a major prospect in artificial intelligence. Since face perception is by definition performed through visual observation, the job naturally went to the field of artificial intelligence that is given "artificial eyes", namely computer vision.

The ambition in computer vision is of course greater than automating face perception only. Broadly speaking, the working hypothesis in computer vision is that any and all cognitive processes that humans carry out through their visual system can be emulated by computerized algorithms, designed to aptly analyze and give an interpretation to raw visual data coming from imaging technologies. Most interesting computer vision problems are hard and remain challenging as

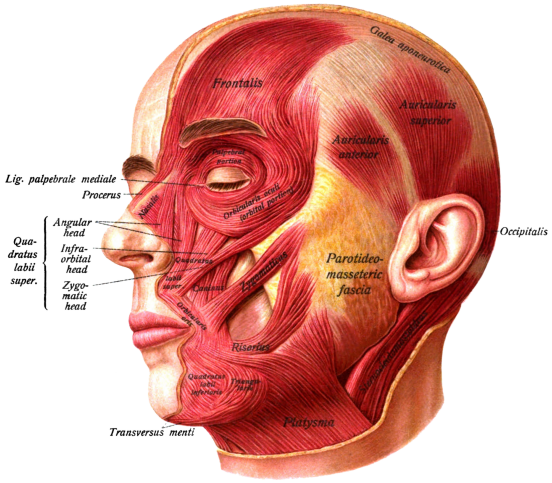


FIGURE 1.1: The superficial layer of the facial muscles and the neighboring muscles of the neck. Illustration from the *Atlas and text-book of human anatomy* [8].

of today, such as object recognition with thousands of categories [9], and scene understanding, which consists in giving a description of what is happening in a random scene of the world using the relevant elements composing the scene and their relationships [10]. Many such problems require advanced, sophisticated solutions, sometimes for only marginally approaching the performance of humans in the same tasks. As computer vision is essentially a field of applied sciences and technologies, additional challenge comes from the fact that computer vision solutions should meet the needs of industry and business in order to be truly useful solutions for real-world applications, and hence become the engines of effective vision-based systems. As a result, anticipating limited resources is an early and legitimate concern when designing computer vision solutions, and researchers often have to compromise on the quality and suitability of the available visual data, as well as the accuracy, robustness, and time complexity of their algorithms. Limited resources is especially a concern in application domains where the degree of personal and social intrusiveness of the proposed solutions is critical, i.e., in applications designed to interact with humans, or analyze human behaviors. Indeed, vision and interpretation are non-interfering, discreet processes by nature, and intrusive vision-based systems may quickly bring discomfort to their users and/or bias social cues, therefore losing the greater part of their real-world applicability. In particular, as a result of the major role of face perception in social-cognitive processes, computer vision solutions to face perception automation problems are notably subjected to the systematic, inherently imposed constraint to be not or only mildly intrusive in their setup, their functioning, and their production of an interpretation.

Nowadays, effective vision-based face perception systems are used in a variety of application domains, e.g., in biometrics [11], video surveillance [12], image database management [13], human computer interaction [14], marketing studies [15], and behavioral studies [16], to name a few. At the core of such systems are computer vision solutions developed and perfected throughout many years of research, toward emulating face perception tasks that may seem easy and basic from a human perspective. For instance, face detection, i.e., the problem of finding the locations of human faces in a random image, was truly effectively tackled for the first time in [17], where

the machine learning approach known as boosting was used with simple image features rapidly computed from the summed area table of the image. This breakthrough seminal work was later improved in various ways [18], but it demonstrated by itself that vision-based face detection could be used in real-world applications, with human-like robustness and processing speed. Face alignment, i.e., the problem of finding the locations of fiducial landmark points in a face image, such as the eye corners, the nose tip, the mouth corners, etc., is another problem that has been widely studied in the computer vision community with great success. The first practical, fast and accurate solutions to face alignment were based on the iterative refinement of the locations of facial landmark points from a coarse initial estimate, using a deformable shape model and an image cost function [19, 20]. Building on this optimization approach, robust face alignment solutions were later proposed to cope with random changes in identity and illumination conditions [21, 22, 23, 24]. Another popular computer vision problem is emotion recognition, which is normally defined as recognizing in face images the so-called prototypical emotions, i.e., happiness, sadness, anger, disgust, surprise, and fear. Very effective solutions have been proposed for this problem, which have human-like performance on data produced in a laboratory environment, i.e., in an environment where the illumination conditions are controlled and the subjects are collaborative. For instance, the authors of [25] proposed to use image features that encode the local directional patterns of the face texture, and perform emotion recognition from these features with the machine learning method known as support vector machine. Face recognition, i.e., the problem of identity retrieval from a face image, has also received a lot of attention in the computer vision community [26]. Recently, Facebook proposed its face recognition system called DeepFace, which uses 3D face modeling and the powerful machine learning approach known as deep learning with artificial neural networks. DeepFace has a performance nearly equivalent to that of humans in the task of distinguishing thousand of faces from each other [27], which is particularly impressive because DeepFace was benchmarked on the "Labeled Faces in the Wild" dataset [28].

The term "in the wild" is used to distinguish data produced in a laboratory environment from data acquired from one or several more natural sources, i.e., data acquired in random environments where no control is applied *a priori*. Achieving human-like performance in the wild is actually an attractive challenge common to most interesting computer vision problems. For some years now, recurrent events are organized where computer vision researchers are invited to test their methods and present their results on in the wild benchmark datasets for various visual tasks. To name a few which pertain to the automation of face perception tasks: the "300 Faces In-the-Wild Challenge" for face alignment [29], the "IARPA Janus Benchmark A Challenge" for face recognition [30], and the "Emotion Recognition in the Wild Challenge" for emotion recognition [31]. Good performance in the wild is however not necessarily a must-have in every real-world application. If for some application the visual conditions can somehow be controlled without foreseeable issue, then designing a solution on the basis of these controlled conditions will most likely require less effort to eventually offer a vision-based system that is still effective for this application. Additionally, for most visual tasks performed in the wild, there remains a significant gap in performance between computer vision solutions and humans, often restricting such solutions to applications where severe inaccuracies and serious errors can be tolerated. This performance gap also impacts the real-world applicability of computer vision solutions proposed for face perception tasks in

the wild, e.g., emotion recognition in the wild performs decently, but is still far to match human performance [32, 33].

Face perception tasks are of course more numerous than the few examples we have mentioned so far. The computer vision problems of face detection, face alignment, emotion recognition, and face recognition are especially salient and popular, because they are intrinsically very general-purpose. High-performance standalone solutions to such problems typically spark ideas for improving various already existing applications, and sometimes inventing brand new ones. As a matter of fact, the research effort in the development of novel computer vision solutions to face perception automation is not as often driven by the requirements of specific applications than by the overall goal of matching human performance in general-purpose problems. In this thesis, however, we are mostly interested in satisfying the requirements of specific applications about the automation of specific face perception tasks. Before we go into detail about these tasks and applications of interest to us, we believe it is useful to present and discuss what we consider as three possible main axes of distinction for categorizing a face perception task.

The first main axis of distinction is whether a face perception task is about characterizing a property of the face seen as a mere object, or about recognizing the state of some hidden process that shows through the face. For instance, face detection and face alignment are computer vision problems associated with object-focused face perception tasks. They are exclusively about the localization of the face object, and the deformation patterns of the face object shape, respectively. Face recognition and emotion recognition are in contrast associated with hidden process-related face perception tasks: a person's identity shows on the face, as well as a person's emotions, but the face is in these cases a means to access information about the hidden processes of individual names and emotional states, respectively. Likewise, face-based age estimation and gender recognition are face perception tasks that aim at revealing semantics beyond the face object, about the hidden processes that are the current effect of time on a person, and the biological sex, respectively. Solving a problem of object-focused face perception task is often considered an upstream stage toward solving what is seen as a more advanced problem of hidden process-related face perception task. This problem decomposition strategy is intuitive and often successful, but it has its drawback: it may wrongly presuppose what the true relationship between the face object and the hidden process is, and therefore lead to suboptimal performance in the automation of the end task despite intense research efforts dedicated to automate the intermediary task presumed to be necessary.

The second main axis of distinction is about the amount of complementary context information that is needed to infer a valid interpretation in a face perception task. In other words, this distinction is about the degree of dependence of a face perception task on information that cannot be gathered by visual observation of the face. Object-focused face perception tasks are typically context-free, as they can be carried out solely by visual observation of the face and provide a unique, valid interpretation on this basis, e.g., "the face is there" in face detection, or "the nose tip and eye corners are there" in face alignment. Some hidden process-related face perception tasks require little to no context information to be carried out, e.g., in face recognition, the uniqueness of a person's identity shows on the face without much visual ambiguity (except for a few exceptions, e.g., identical twins or doppelgangers). Likewise, in emotion recognition with sane, collaborative subjects, the patterns of facial muscle contractions used to display a prototypical emotion are

mostly without visual ambiguity (even though some subjects may display prototypical emotions via their face in an unexpected manner). Many other hidden process-related face perception tasks are however strongly influenced by context. For instance, the recognition of more complex, subtle emotions in the face, such as anguish, boredom, embarrassment, envy, frustration, guilt, humiliation, lust, pity, remorse, shyness, and worry, to name a few, represents a much harder problem that is seldom addressed, as it requires to integrate cultural and personal factors, and probably other pieces of context information, in order to provide valid interpretations. Besides, we believe that the recognition of the prototypical emotions in the wild, i.e., with non-collaborative subjects, is also plagued by ambiguities at the visual level in the face, which could explain the performance gap between humans and computer vision solutions in this case. Another example is the task of age estimation by face observation. Humans actually use more information than what shows on the face to narrow down a plausible age range for a person, such as the estimated physical condition of the subject, and various social cues [34]. One last, maybe more obvious example is the face perception task of physical attractiveness estimation, which requires to take into account not only various cultural factors, but also observer-related factors, like the observer's idea of his/her own attractiveness [35]. A common strategy used in the automation of context-dependent face perception tasks is to assume the context to be fixed *a priori* once and for all, in order to match a unique interpretation to any visual appearance of the face. This strategy may however recast the complex face perception task one originally tries to automate into another, simpler one of lesser interest from a practical standpoint. Such semantic loss is illustrated by the solutions commonly proposed for the problem of emotion recognition, which are limited to the recognition of six prototypical emotions, deemed to be mostly independent of any context and fully characterized by the visual information found in the face.

The third main axis of distinction, which is of particular interest to us in this thesis, is whether a face perception task deals with a facial expression or not. Precisely defining what facial expressions are is not a trivial exercise. In the usual language, facial expressions are hardly dissociable of complex semantics beyond the visual level, and are thus naturally associated to hidden process-related face perception tasks: "an expression of anger" conveys an emotional state, "an expression of approval" gives a communication cue, "an expression of weariness" indicates a physiological activity, etc. In the field of behavioral psychology, it is proposed to consider facial expressions as patterns of facial muscle contractions, stripped of any interpretation that is beyond the face object. Therefore, in behavioral psychology, distinguising and recognizing facial expressions are object-focused face perception tasks, e.g., simultaneously raising the inner eyebrows and the upper eyelids is a facial expression with the code `[1+5]` in the taxonomic coding system of facial expressions proposed in [36]. It is then proposed as a second step to study how the facial expression codes may be further interpreted to reveal the state of a hidden process, e.g., an emotion, a feeling of pain, a depression cue, and so forth. This psychology-based definition of facial expressions is appealing to computer vision researchers, mostly because it is almost purely visual and requires very little context. It is however quite bare and postpones any effort of interpretation that is beyond the face object, whereas examples in the usual language clearly indicate that very often facial expressions carry complex meaning. Furthermore, focusing on the facial muscles only is insufficient in our opinion, because we think that head and eye movements largely contribute to

facial expressions in many cases. We therefore propose to use the following definition for a facial expression and its possible interpretation, and we construct the third main axis of distinction of face perception tasks on the basis of this definition.

> A *facial expression* is a shared and temporary human behavior, purposeful or not, conscious or not, which primarily involves the whole face or parts of the face.
>
> An *interpretation* for a facial expression is a semantic concept derived by the joint analysis of the visual information obtained about the face during the display of the facial expression, and the context information relevant to the particular interpretation.

Our definition of facial expressions encompasses and extends the psychology-based definition. Facial muscle contractions may be involved in our definition, but not only. Also, the recognition of a facial expression is a face perception task that is not limited to provide an object-focused interpretation in our definition. To clarify, we give below some examples of face perception tasks that do and do not belong to the interpretation of facial expressions according to our definition.

- Recognizing facial expression codes in a descriptive taxonomic coding system.

  Yes. In this case, facial expressions are interpreted as patterns of facial muscle movements in a coding system. This face perception task is object-focused, but may require a some context, e.g., to know to which extent a particular person can contract some facial muscle.

- Recognizing emotions, like happiness, sadness, fear, etc.

  Yes. This face perception task interprets facial expressions as emotional states, therefore it is hidden process-related. Context information should here include how emotions are facially displayed in the targeted population, for example accounting for cultural conventions.

- Recognizing a physiological activity, like blinking, sneezing, yawning, etc.

  Yes. Such face perception tasks interpret a facial expression as the indicator of a physiological activity. They are therefore hidden process-related. They are not context-free, since reflex or semi-autonomous facial expressions can be visually mistaken with purposeful facial expressions, e.g. blinking with winking.

- Recognizing a head movement, like a head shake, nodding, etc.

  Yes. Even though it is the neck muscles that are involved here and not the facial muscles, the observer performs a face perception task in the category of facial expression interpretation according to our definition. Such tasks are most often hidden process-related and context-dependent as, for instance, the recognition of nonverbal approval.

- Recognizing an eye movement, like eye-rolling, target fixation, etc.

  Yes. Even though it is the eye muscles that are involved here, the eyes are part of the face and their temporary movements are facial expressions according to our definition. Such tasks are typically hidden process-related, e.g., determining whether a person is aware of his/her surroundings or not. They are also most often context-dependent, e.g., it may be required to know the position of the target fixed by the person to give a valid interpretation.

- Recognizing a personal characteristic, like identity, gender, age, etc.

  No. Personal characteristics are not shared by people, by definition. Also, personal characteristics are typically persistent, and hardly definable as temporary. Such face perception tasks do therefore not belong to the category of facial expression interpretation, according to our definition. They are however hidden process-related, and often context-dependent.

- Recognizing the presence or location of the whole face, the nose tip, the eye corners, etc.

  No. These object-focused, context-free face perception tasks are not about human behaviors *a priori*, and therefore do not fit in our definition of a facial expression. They may however consist of an upstream stage toward recognizing a facial expression.

We now introduce the practical and theoretical contributions of this thesis, which is about the computer vision-based automation of specific face perception tasks in specific applications of facial expression interpretation. Our practical contributions consist of the development of effective vision-based systems within two specific application domains: sign language recognition, for which we propose our facial cue extraction system based on the analysis of face images (Chap. 2), and visual pursuit assessment in patients with a disorder of consciousness, for which we propose our objective score calculation system based on the analysis of eye images (Chap. 3). In both of our systems, we make a fruitful use of well-studied, mature computer vision methods to solve the problems related to our application domains of interest. Our theoretical contributions come from an empirical study where we compared the performance of hierarchical and standard, "flat" classification in specific vision-based recognition problems (Chap. 4). The first problem considered in this study is the recognition of facial expression codes in a descriptive taxonomic coding system. We believe that the general conclusions we reached in this study are of interest for properly using hierarchical classification in vision-based recognition problems, in particular in muscle-based facial expression recognition, toward a performance gain over flat classification.

Our contributions do not reside in the development of general-purpose computer vision methods for the automation of tasks of facial expression interpretation. In other words, this thesis is not about inherently novel algorithmic techniques that would be universally usable for solving any and all problems related to the interpretation of facial expressions. Instead, the focus in this thesis is given to (1) our practical use of state-of-the-art computer vision methods for developing two effective vision-based systems in two application domains about facial expression interpretation, and (2) our empirical study and theoretical conclusions about hierarchical vs. flat classification in vision-based recognition problems such as facial expression code recognition. As an aside, we try to use a comprehensive terminology in this thesis for structuring the concept of facial expression interpretation, the main elements of which terminology have been presented in this introductory chapter. Finally, in the conclusion of this thesis, we give further comments about our overall work, and consider possible paths for improvement (Chap. 5). We also briefly mention in the conclusion the contribution we made to the development of another vision-based system within the application domain of drowsiness monitoring. This system was proven effective enough to allow the creation of a commercial activity via a spin-off company from the university of Liège.

Chapter 2

Facial cue extraction system for automatic sign language recognition

In this chapter, we present our computer vision system that extracts specific facial expressions used in the recognition of sign language. Specifically, our system robustly tracks landmark points on a signer's face in a video, and derives sign language facial communication cues from the tracked landmark points. We originally developed this system as a contribution to the SignSpeak project, funded by the European Community's 7th Framework Programme, where the goal was to create a new vision-based technology for translating sign language to text and improve the communication between deaf and hearing people [37]. To showcase the usefulness of our sign language facial cues, we report some results obtained with the sign language recognition system chain originally proposed in SignSpeak, which incorporates our system. We also give an evaluation of the facial landmark point tracking performance of our system. Besides in SignSpeak activity reports, part of our work was published in an paper gathering early SignSpeak results on video-based sign language facial and hand cue extraction, in the proceedings of the 4th Workshop on the Representation and Processing of Sign Languages [38]. Another publication related to our work in SignSpeak describes a face shape-annotated sign language recognition dataset, in the proceedings of the 8th International Conference on Language Resources and Evaluation [39]. After SignSpeak, we participated to follow-up published work where our facial cue extraction system was used, including (1) a system that recognizes the visible phonemes, which was shown to improve sign language translation, in the proceedings of the 10th International Workshop on Spoken Language Translation [40], and (2) a technique to enhance sign language avatar animation with facial expressions and mouthing, in the 3rd International Symposium on Sign Language Translation and Avatar Technology [41]. The content of this chapter is an original compilation of the unpublished work on our facial cue extraction system, and the jointly published work where our system was used or partly described. In any case and unless explicitly stated otherwise, all work presented in this chapter is our own.

2.1 Introduction

Sign languages are used by about 70 million deaf people around the world as their mother language, and several millions of hearing people as their secondary language [42]. People able to sign, i.e., to perform some form of sign language, are commonly referred to as signers. A generic and simple "contact" sign language, called International Sign, helps signers to quickly understand

each other in congresses, sports events, and when traveling and socializing. Yet the richer local sign languages are preferred by signers in their everyday interactions [43]. Local sign languages are not simple gestural codes nor pantomimes, but complex ways of communication which naturally develop and diversify through time and location. Each country typically has one or more local sign languages, and even some dialect forms do exist. Interestingly, local sign languages share linguistic roots in the same way as spoken languages do [44], but are otherwise largely independent of the surrounding spoken languages used in the same regions. This is because the natural local sign languages typically develop within the deaf communities, independently of the spoken languages. Actually, ways of communication that are primarily based on exploiting the visual-gestural medium are thought to have existed and developed in parallel for as long as the articulated speech in human history [45]. As a consequence of the strong differences between spoken languages and local sign languages, signers which depend the most on the visual-gestural medium, i.e., the deaf and hard of hearing communities, encounter huge difficulties to communicate with the hearing people that do not use sign language (and the other way around). Therefore, deaf people cannot easily integrate into the educational, social, and work environments typically designed for the hearing people [46]. Since the vast majority of hearing people cannot be compelled to learn functioning elements of sign language, and since the deaf people obviously cannot adjust any further to environments designed for hearing people, the social gap between the deaf and hearing communities is strongly present, even in the countries plainly aware of, and willing to tackle this issue [47]. In our modern times, in which the research in artificial intelligence and the industrialization of information technologies are booming, it is expected that this gap between signers and speakers be somehow bridged by the development and spread of automatic systems specialized to recognize sign language [48].

From a perceptual point of view, the difference between sign and spoken languages lies in the type and number of modalities that are at play during the communication process [49]. Indeed, spoken languages primarily use the auditory-vocal medium and therefore focus on a single modality that is the vocal tract. Sign languages use the visual-gestural medium, where the hand gestures, facial expressions, and body posture correspond to multiple parallel modalities, which altogether convey communication cues from the signer to the viewer. Hand gestures are of course essential to sign languages, but the other modalities play an important role as well, especially facial expressions. In fact, during a communication in sign language, the viewer looks more at the face than at the hands of the signer, for seizing clarity and sensitivity. In frequent occurrences, hand gestures are even ambiguous in isolation, and facial expressions then behave as nonmanual grammatical and semantic markers that are crucial to convey the specific message, e.g., to indicate the negation, or the interrogative, as shown in Fig. 2.1. In many sign languages, the most important, disambiguating facial expressions include raising or lowering the eyebrows, opening or closing the eyes or the mouth, and nodding or shaking the head.

Perceptually, sign languages are multimodal by nature. From a linguistic point of view, however, a communication in sign language has a sequential structure composed of small grammatical units, i.e. morphemes, just like words compose a communication in a spoken language. Fluent signers can point out which sequence of morphemes corresponds to a sentence in sign language,

FIGURE 2.1: Left: the "Yes/No" interrogative facial expression used in the American sign language (ASL). Right: the hand gesture "You" in ASL with the "Yes/No" facial expression in ASL, meaning either "Is it you?", or "Are you...?", or "Did you?". Pictures from the ASL University website (http://www.lifeprint.com/).

by watching and combining information from the hand gestures, facial expressions, and body posture [50, 44]. This sequential linguistic view of sign language helps designing an automatic sign language recognition system. Indeed, the problem of sign language recognition can therefore be divided into two parts: (1) converting the sign language multimodal communication signal into a sequence of morphemes, and (2) converting the sequence of morphemes into a well-constructed text in a target spoken language. Assuming that sign language morphemes can be expressed in a text form, the first part corresponds to a transcription process, and the second part corresponds to a translation process.

Transcribing a sign language is not as natural a process as transcribing a spoken language. When transcribing a spoken language, the target text form is most of the time given, rich, and intuitive, because writing is actually a natural complementary technology to speech [51]. Sign languages, however, do not have natural text forms associated to them [52], and writing down a communication in sign language first requires to define its target text form. A sign language can receive a somewhat expressive text form, via a morpheme-by-morpheme annotation procedure known as glossing, a glossed sentence being the resulting sequence of morpheme labels, or glosses [50]. Glossing is actually more general than its application to the transcription of sign languages. This annotation procedure was originally invented as an accessory tool to help a reader understand, via labels in a familiar language, the lexical and morphological patterns of a sentence in a foreign language. For example, German "Es geht mir gut" may be glossed using English as "It goes to-me good", which is not a correct translation, but a sequence of labels that retain the necessary information to reconstruct the smooth, grammatically correct and meaningful corresponding sentence in English, i.e., "I'm doing fine". In sign language glossing, the glosses are typically composed of words in the target spoken language, as well as of various other exotic notations to account for what cannot be expressed with words. For instance, in American sign language (ASL) glossing, the contraction between two words is denoted by "__^__", which gives the ASL gloss "DO__^__NOT KNOW I" for the sentence "I don't know". Using an effective glossing procedure, one can therefore accurately transcribe sign language, and further produce so-called parallel corpora that include glossed sentences in a source sign language, and written sentences in a target spoken language. In all generality, parallel corpora are datasets of aligned transcribed

sentences from a pair of source and target languages. Such corpora are typically used by machine translation methods to learn a translation model between a source and a target language of any nature [53]. This means that the translation process part of a sign language recognition system can be pretty much a boilerplate implementation of a standard machine translation method based on parallel corpora. If the source language is a glossed sign language, the quality of the translation will mostly rest upon the quality and expressiveness of the glossing procedure.

An end-to-end, fully automatic sign language recognition system should include a part that performs automatic glossing, in the same way as a voice-based translation system between two spoken languages includes a part that performs automatic speech transcription. Speech to text technologies attempt to segment the vocal tract signal and assign their associated text elements to the voice segments [54]. In sign language, the visual signal to transcribe, i.e., to gloss, is composed of multiple modalities that are all important for the communication process [49]. Automatically glossing sentences in sign language thus requires a system able to divide the multimodal visual signal with gloss-level time boundaries, and assign their associated glosses to the segments defined by these time boundaries. Note that this view corresponds to the automatic glossing of continuous sign language [37], where full sentences are to be analyzed, with the signer performing one sign after the other naturally, without pausing. However, the visual signal segmention part is sometimes left aside in the problem of automatic glossing, therefore focusing on the simpler isolated sign recognition problem [55, 56, 57]. In any case, effectively tackling single-gloss recognition from the multimodal visual signal is an essential part of any automatic glossing system.

In practice, a cheap yet effective way to capture the multimodal visual signal from sign language is to acquire videos where the signer appears whole from the waistline up. Using such video data, one can take a component-based view and independently extract information about the hands, face, and body posture, using computer vision methods. The information extracted from all modalities can then be combined into a full high-level representation of the multimodal visual signal [38], and, from this point onward, automatic glossing can be treated as a supervised machine learning problem, where the machine has to learn how to infer gloss labels from the combined representation of visual features extracted from videos of the signers. The recognized glosses can then be fed to a standard machine translation system, trained with parallel corpora of glossed and spoken language-transcribed sentences, to produce the final spoken language translation.

The work presented in this chapter is about the computer vision system we developed for the automatic extraction of sign language-specific facial expressions from a communication in sign language recorded on video. Our system, an early version of which was partly introduced in [38], is based on the tracking of a set of facial landmark points in the signer's face image. For tracking these landmark points, we use a face alignment method called active appearance models, in particular the formulation proposed by [21], with several refinements. On the basis of the frame-by-frame configurations of the tracked facial landmark points, our system continuously extracts specific facial expressions useful for sign language recognition, e.g., measurements of the mouth and eye opening degrees. It is noteworthy that we did not specify these facial expressions according to codes in a general-purpose facial expression coding system, such as the one in [36]. Instead, the facial expressions we extract with our system were specified in a way that is directly useful to sign language recognition, notably with the help of linguistics and machine translation

experts specialized in sign language. Because this specification comes from the contextual need to effectively identify facial communication cues in sign language, we consider that our system automates a face perception task of facial expression interpretation (cf. our discussion in Chap. 1), and we call our system a sign language facial [communication] cue extraction system, to emphasize that, ultimately, its role is to automatically provide facial cues within a sign language recognition system chain.

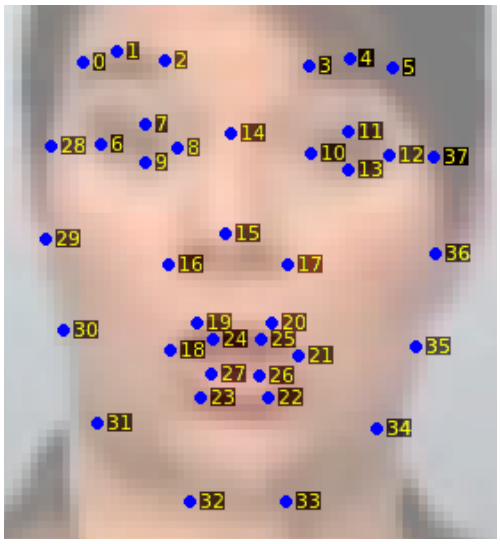
In the results section of this chapter, we first present some facial landmark point tracking results, obtained with the face alignment method on which our sign language facial cue extraction system is based. As a byproduct of the design of our system, we created facial landmark point annotations, which were incorporated into a sign language recognition dataset (described in length in [39]). We also briefly present this dataset in the results section of this chapter. Then, to showcase the usefulness of our system in the automatic recognition of sign language, we present some results obtained with a gloss recognition system, which uses our facial cues in combination with hand cues [58, 37]. Finally, we present some typical machine translation results from glosses to spoken language, and we show how these results can be improved by closely integrating the gloss recognition and machine translation frameworks with a viseme, i.e., visible phoneme recognition system, as opposed to the standard sequential approach of gloss recognition followed by machine translation. This viseme recognition system is based on our system to extract sign language facial cues, and seemingly performs the automatic recognition of mouthing, i.e., silently pronouncing words of a spoken language while signing [40].

2.2 Material and methods

2.2.1 Face alignment

The problem of face alignment is often treated as a particular instance of the general problem of deformable object alignment, which consists in finding the best image locations of a set of landmark points *a priori* defined for some object of interest that has a deformable shape, e.g., the face, an internal organ, etc. Therefore, even though in the following our phrasing may seem limited to face alignment, the reader should keep in mind that most of the technicalities presented in this section can be used as is for solving other deformable object alignment problems. Also note that all material and methods presented in this section about face alignment can be found in the literature, in one form or another. Yet all formulation of the concepts and all implementation choices are our own, and we hope that the reader will find our presentation useful to grasp what face alignment is all about, and how we implemented face alignment in our application.

Formally, the problem of face alignment in a 2D image is as follows. We define a facial landmark point as a couple $(\mathbf{x}_i, c_i)$, where $\mathbf{x}_i$ is a variable image location, i.e., a 2D point in some image coordinate system, and c_i is a fixed, predefined semantic concept about the face, e.g., "the inner corner of the left eye". Let $\mathcal{L} = \{(\mathbf{x}_i, c_i)\}, i \in \{1, \ldots, L\}$, be a set of facial landmark points. The size L of this set is arbitrary, but finite. Let $\mathcal{I}$ be a (possibly infinite) set of images containing a face. The problem of face alignment consists in finding an algorithmic method which, for any image $I \in \mathcal{I}$, moves the L facial landmark points of $\mathcal{L}$, so that their image location $\mathbf{x}_i$ in I best corresponds to their semantic concept c_i.



Point indices	Face parts
0–2	right eyebrow upper part
3–5	left eyebrow upper part
6–9	right eye eyelids
10–13	left eye eyelids
14–15	nasal ridge
16–17	nasal base
18–21, 24–25	upper lip
18, 21–23, 26–27	lower lip
28–37	cheeks and chin

FIGURE 2.2: An example of a face shape annotation with 38 landmark points. We annotated 369 such images with these landmark points for the seven signers of the RWTH-PHOENIX-Weather sign language recognition dataset (introduced in [39]). These images and shape annotations are available for download at https://iis.uibk.ac.at/datasets/phoenix-annotations.

The number L of facial landmark points one can define is arbitrary, as well as the anatomical face regions where to anchor them, embodied by the semantic concepts $\{c_i\}$. However, as face alignment methods usually involve building statistical models from shape-annotated face images provided by human annotators, e.g., the shape-annotated image in Fig. 2.2, there are two rules of thumb for defining the facial landmark points: (1) to place them in regions where a human annotator can find them consistently for various people's faces with various facial expressions, and (2) to increase their density in regions that deform the most within the face, e.g., the mouth region. The expected resolution of the face image may be an important aspect as well, because if the face image resolution is poor, a human annotator will find it difficult to accurately adjust a fine-grained face shape composed of many landmark points.

Of the many methods proposed for face alignment, we use active appearance models in this work (AAMs, first proposed by [20]), as a part of our facial cue extraction system, which is to be used eventually as a part of a sign language recognition system chain. Our choice to use AAMs is arbitrary, and another face alignment method could be used instead, for the same purpose and with similar benefits. Indeed, there exist other methods than AAMs which have been proven to be better suited for performing face alignment in the wild, e.g., constrained local models [59], cascaded shape regression models [23, 60], and deep convolutional network models [24]. AAMs however have multiple interesting qualities, for which they deserve consideration:

1. They have a principled and appealing formulation: their machinery is easy to grasp, based on what is called the generative approach to modeling.

2. They are widely studied: many refinements have been proposed to improve their robustness to occlusions, their time efficiency, and their generalization capabilities.

3. They are flexible: the gradient descent-based formulation of their alignment procedure allows to incorporate various custom terms in the form of smooth constraints.

4. They are effective in practice: they integrate smoothly as a functional part of real-world systems for various face-centered applications.

In light of these qualities, and despite their suboptimal performance in the wild [61], we believe that AAMs constitute a method of choice in our application. Indeed, our application context does not require a face alignment method with outstanding performances in the wild. We aim at extracting facial communication cues from videos of signers who are aware of being filmed by a camera. Moreover, signers naturally try to make themselves be well-understood via the visual-gestural medium, notably meaning here to be "easy-to-see", for the purpose of obtaining a high-quality communication in sign language. Yet it is natural for signers to perform large head movements, extreme facial expressions, and to wave their hands in front of their face. AAMs allow us to tackle these difficulties elegantly by the addition of refinements designed to be robust against such cases.

It must however be pointed out that AAMs have difficulties to effectively generalize with respect to identity, i.e., to be fast, robust, *and* generic to all people, even in controlled, not in the wild conditions [21]. To tackle this limitation, we use the strategy proposed in [62], for creating on-the-fly a fast and robust AAM specific to a new signer, by adapting a slower and less robust generic AAM built from many different people's faces. This strategy helps us achieve a generalization performance of a quality that allows our system to work with most people's faces.

Modeling the deformable face shape

One could consider it is a good strategy to break up the problem of face alignment into multiple independent problems of single landmark point alignment. However, ignoring the geometric relationships between the landmark points of the face shape is actually prone to failure in most practical cases, because the relationship between a single landmark point and the image is typically ambiguous at the local level. AAMs and many other face alignment methods therefore take a holistic approach for modeling the face shape, i.e., they consider the landmark points as a whole, by representing them and their geometric relationships with a single shape model. A holistic shape model can indeed be used to regularize the face alignment procedure, i.e., to resolve the local shape-to-image ambiguities and thus prevent absurd, non-valid face shapes to be even considered as possible solutions by the procedure.

AAMs build and use a holistic shape model in the form of a parametric point distribution model (PDM). A parametric PDM is a mapping between a set Θ of shape parameters and the set of image locations $\{\mathbf{x}_i\}$ of the L facial landmark points in $\mathcal{L}$ (see previous section). We denote a parametric PDM, from this point forward simply called PDM, as a vector mapping

$$\mathbf{s}: \quad \mathbb{R}^{|\Theta|} \rightarrow \mathbb{R}^{2L}, \quad \theta \mapsto [\mathbf{x}_1; \ldots; \mathbf{x}_L], \tag{2.1}$$

where θ is the shape parameter vector that contains the shape parameters in the (ordered) set Θ, and where $[\mathbf{x}_1; \ldots; \mathbf{x}_L]$ is the $2L$ long shape vector that contains the stacked 2D coordinates of

image locations $\{\mathbf{x}_i\}$ of the L facial landmark points. Without loss of generality, the parameterized shape vector $\mathbf{s}(\boldsymbol{\theta})$ can be denoted as $[\mathbf{x}_1(\boldsymbol{\theta}); \ldots; \mathbf{x}_L(\boldsymbol{\theta})]$. Also, in the following, when the shape parameters have fixed values, we often conveniently abuse the notation in Eq. 2.1 by omitting the explicit mention of the fixed shape parameter vector $\hat{\boldsymbol{\theta}}$, and denote the mapped fixed shape vector $\mathbf{s}(\hat{\boldsymbol{\theta}})$ simply as $\mathbf{s}$, and a mapped fixed landmark point location $\mathbf{x}_i(\hat{\boldsymbol{\theta}})$ simply as $\mathbf{x}_i$.

A PDM can produce a complete face shape for any values of the shape parameters, which is why this type of modeling is called generative, as opposed to discriminative modeling. A discriminative model cannot be used to actually produce face shapes, but retains only what is needed to apply shape regularization within the face alignment procedure, e.g., in [60], where shape regularization is learned as a series of regression models, one for each stage of an iterative alignment procedure, each model being used to apply a holistic correction to independently estimated facial landmark point locations at this stage.

The PDM shape parameters in Θ are partitioned in two parameter subsets, which account for two different types of shape deformation called global and local, i.e., in vectorial form, $\boldsymbol{\theta} = [\boldsymbol{\theta}^{(g)}; \boldsymbol{\theta}^{(l)}]$, where $\boldsymbol{\theta}^{(g)}$ accounts for global shape deformation, and $\boldsymbol{\theta}^{(l)}$ accounts for local shape deformation. The notion of global and local shape deformation has to do with the geometric invariants defined for a type of shape deformation. For instance, two face shapes can be considered locally equivalent even though one is twice as large as the other, because the local shape deformation is considered invariant to scaling. Conversely, two face shapes can be considered globally equivalent even though they depict different people, because the global shape deformation is considered invariant to changes in identity. In total, four global shape parameters are chosen for the PDM to account for global shape deformation, i.e., $\boldsymbol{\theta}^{(g)} = [s; t_x; t_y; \alpha]$, where s controls isotropic scaling, t_x and t_y control 2D horizontal and vertical translation, respectively, and α controls 2D rotation. These parameters correspond to the four degrees of freedom (DoF) of the 2D similarity transform, and can also be used to transform any fixed shape $\mathbf{s} = [\mathbf{x}_1; \ldots; \mathbf{x}_L]$ to another shape $\mathbf{s}' = [\mathbf{x}'_1; \ldots; \mathbf{x}'_L]$ with a similarity mapping $\mathbf{N}$, as

$$\mathbf{N}: \quad \boldsymbol{\theta}^{(g)}, \mathbf{s} \mapsto \mathbf{s}', \quad \text{where} \quad \mathbf{x}'_i = s \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \mathbf{x}_i + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad \forall i \in \{1, \ldots, L\}. \tag{2.2}$$

The PDM local shape parameters $\boldsymbol{\theta}^{(l)}$ are meant to control any type of shape deformation that cannot be described by global shape deformation based on the similarity transform. In AAMs, the quantities used to define and manipulate the local shape parameters are obtained by statistical analysis on a set of example face shapes, via a two-step procedure. Let $\{\mathbf{s}_j\}, j \in \{1, \ldots, T\}$, be a set of example face shapes, provided by human annotators. The first step of the procedure is to align those shapes globally using a Procrustes analysis [63]. This Procrustes analysis effectively removes the difference in global deformation between the example shapes $\{\mathbf{s}_j\}$, by transforming every example shape $\mathbf{s}_j$ with a 2D similarity $\mathbf{N}$ (Eq. 2.2), so that their transformed shape $\mathbf{s}'_j$ is globally equivalent to a common reference shape $\mathbf{s}_0$, i.e.,

$$\mathbf{s}'_j = \mathbf{N}(\hat{\boldsymbol{\theta}}^{(g)}_j, \mathbf{s}_j) \overset{g}{=} \mathbf{s}_0, \quad \forall j \in \{1, \ldots, T\}, \tag{2.3}$$

where $\hat{\boldsymbol{\theta}}^{(g)}_j$ denotes the optimal global shape parameter values used to align $\mathbf{s}_j$ to $\mathbf{s}_0$ and give $\mathbf{s}'_j$,

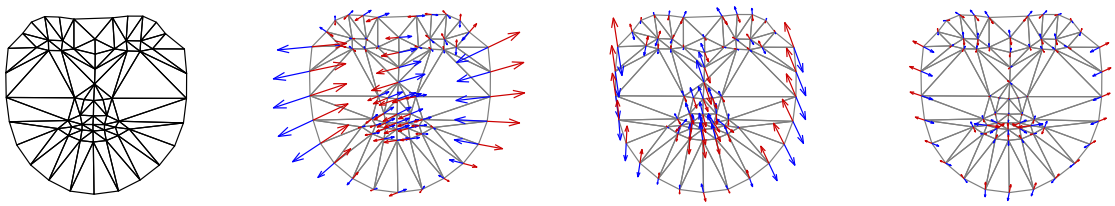


FIGURE 2.3: The local part of a PDM for the face shape, as a subspace model built by PCA. The (triangulated) reference face shape $\mathbf{s}_0$ is shown to the left, and to the right are the actions on $\mathbf{s}_0$ of the first three modes of local deformation $\boldsymbol{\Phi}_1$, $\boldsymbol{\Phi}_2$, and $\boldsymbol{\Phi}_3$. The lengths of the deformation vectors are indicative of the variances observed in the PCA for those modes. This illustration is from [65].

and $\overset{g}{=}$ denotes the global equivalence between two shapes. The second step of the procedure is to apply a principal component analysis (PCA) to the set of aligned face shapes $\{\mathbf{s}'_j\}$ [64], which eventually gives a linear subspace model for the (globally aligned) face shape, as the mapping

$$\begin{aligned} \mathbf{s}' : \quad \boldsymbol{\theta}^{(l)} \;\mapsto\; \mathbf{s}_0 + \boldsymbol{\Phi}\boldsymbol{\theta}^{(l)} = \mathbf{s}_0 + \sum_{i=1}^{N} \theta_i^{(l)} \boldsymbol{\Phi}_i, \\ \boldsymbol{\Phi} = [\boldsymbol{\Phi}_1 \dots \boldsymbol{\Phi}_N] \in \mathbb{R}^{2L \times N}, \quad \boldsymbol{\theta}^{(l)} = [\theta_1^{(l)}; \dots; \theta_N^{(l)}], \end{aligned} \tag{2.4}$$

where $\mathbf{s}_0$ is the reference face shape used in Eq. 2.3, here corresponding to the origin of the subspace, and $\boldsymbol{\Phi}$ is an orthogonal basis containing linear modes of local shape deformation, which consist of the N principal components calculated on the globally aligned example face shapes. Together, the 2D similarity transform defined in Eq. 2.2 and the subspace model defined in Eq. 2.4 give the complete form of the PDM mapping defined in Eq. 2.1, as

$$\mathbf{s} : \quad \boldsymbol{\theta} = [\boldsymbol{\theta}^{(g)}; \boldsymbol{\theta}^{(l)}] \;\mapsto\; \mathbf{N}(\boldsymbol{\theta}^{(g)}, \mathbf{s}'(\boldsymbol{\theta}^{(l)})) = \mathbf{N}_{\boldsymbol{\theta}^{(g)}}(\mathbf{s}_0 + \boldsymbol{\Phi}\boldsymbol{\theta}^{(l)}), \tag{2.5}$$

where we have used $\mathbf{N}_{\boldsymbol{\theta}^{(g)}}(\cdot)$ in place of $\mathbf{N}(\boldsymbol{\theta}^{(g)}, \cdot)$ to make the notation less cluttered. So defined, a PDM has $4 + N$ DoF, which is typically less than the $2L$ DoF corresponding to the image locations $\{\mathbf{x}_i\}$ of the L landmark points, therefore limiting the possible face shapes that the PDM can produce. To further prevent the production of non-valid face shapes, one should also keep trace of the variances observed along the principal components calculated on the aligned face shapes. Indeed, those variances estimated from valid face shapes provided by human annotators can be used to define a validity region within the subspace, e.g., a hyperellipsoid or a hyperbox, within which the values of the local parameters $\boldsymbol{\theta}^{(l)}$ should remain. Figure 2.3 shows a subspace model built by PCA, representing the local part of a PDM for the deformable face shape.

Note that there exist appropriate methods other than PCA for building a PDM of the face. For example, one can apply an independent component analysis to the example face shapes [66], or model the shape as an elastic material using finite element modes of deformation, without using any statistical analysis [67]. Still, PCA is most commonly used and met in the literature on generative models of the face shape, because of its simplicity and adequacy in providing ways to represent valid face shapes.

Modeling the relationship between the face shape and the image

AAMs model the relationship between the face shape and the image as an image similarity function between a synthetic, shape-parameterized face image and a real image. For any given, real target image, this similarity function is defined over the parameters of a fully generative model of the face shape and face texture, i.e., a model that can produce photorealistic synthetic face images. In this view, aligning the face to a target image consists in finding the optimal model shape and texture parameter values, i.e., the parameter values that give the largest similarity value for the target image. The aligned facial landmark points for the target image can then be retrieved from the optimal shape parameter values. In contrast to this generative approach to modeling, discriminative modeling of the relationship between the face shape and the image is more commonly met in current face alignment methods, which focus on achieving high-quality performance in the wild [61]. However, as mentioned above, our application to sign language recognition does not need outstanding performance in the wild, and face alignment based on generative modeling, notably as proposed in AAMs, is suitable in our case.

In order to get a fully generative model of the face shape and texture, AAMs couple their holistic model of the deformable face shape, i.e., the PDM in Eq. 2.5, with a holistic "shape-free" model of the variations of the face pixel intensities. Specifically, AAMs separately model the dense and shape-normalized face texture, using a generative model that we call a parametric texture distribution model (TDM). Let the normal shape be the reference shape $\mathbf{s}_0$, from the PDM local subspace model in Eq. 2.4. Let $\mathcal{S}_0 = \{(\mathbf{u}_i, a_i)\}, i \in \{1, \ldots, K\}$, be a set of normal pixels, composed of the K fixed, normal locations $\{\mathbf{u}_i\}$ and their corresponding variable intensities $\{a_i\}$[1]. To get a dense view of the shape-normalized texture, the normal pixel locations $\{\mathbf{u}_i\}$ are typically organized as a tight grid that spans the convex hull of $\mathbf{s}_0$. A parametric TDM is then defined as a mapping between a set Λ of texture parameters and the set of intensities $\{a_i\}$ of the K normal pixels. We denote a parametric TDM, from this point forward simply called TDM, as a vector mapping

$$\mathbf{a}: \quad \mathbb{R}^{|\Lambda|} \rightarrow \mathbb{R}^K, \quad \boldsymbol{\lambda} \mapsto [a_1; \ldots; a_K], \tag{2.6}$$

where $\boldsymbol{\lambda}$ is the texture parameter vector that contains the texture parameters in the (ordered) set Λ, and where $[a_1; \ldots; a_K]$ is a texture vector that contains the stacked intensities of the K normal pixels in $\mathcal{S}_0$. Without loss of generality, the parameterized texture vector $\mathbf{a}(\boldsymbol{\lambda})$ can be denoted as $[a_1(\boldsymbol{\lambda}); \ldots; a_K(\boldsymbol{\lambda})]$. Also, in the following, when the texture parameters have fixed values, we often conveniently abuse the notation in Eq. 2.6 by omitting the explicit mention of the fixed texture parameter vector $\hat{\boldsymbol{\lambda}}$, and denote the mapped fixed texture vector $\mathbf{a}(\hat{\boldsymbol{\lambda}})$ simply as $\mathbf{a}$, and a mapped fixed pixel intensity $a_i(\hat{\boldsymbol{\lambda}})$ simply as a_i.

For any values of the texture parameters, a TDM can produce a face texture, dense and shape-normalized. A synthetic, photorealistic image of the face can then be obtained by projecting a texture generated by the TDM onto a shape generated by a PDM, using the facial landmark points as the control points of a warping function. Suitable warping functions parameterized by control points include thin-plate splines, and the more commonly used piecewise affine warp,

[1] For simplicity, we only consider the case where a pixel value is a scalar, gray-level intensity. The case where multiple color channels are involved, e.g., red, green, and blue, would be treated in a similar manner. However, since it would make the notation somewhat cumbersome without bringing much more insight, we do not detail it.

which involves the triangulation of the topology defined by the landmark points, as shown in Fig. 2.3. AAMs actually evaluate the similarity function between real and generated images within the normal domain, thus it is the texture of the real image that is in fact projected onto the normal shape $\mathbf{s}_0$, to be compared to a texture generated by a TDM. Given a real image I and a face shape $\mathbf{s}$ defined in the 2D coordinate system of I, a dense shape-normalized face texture vector can be sampled from I as

$$[a_1; \ldots; a_K], \quad \text{with} \quad a_i = I(\mathbf{w}(\mathbf{u}_i, \mathbf{s}_0, \mathbf{s})) \quad \forall i \in \{1, \ldots, K\}, \tag{2.7}$$

where I is seen as a function, $I : \mathbb{R}^2 \rightarrow \mathbb{R}$, which maps a pixel location $\mathbf{w}$ in the image coordinate system (i.e., the image domain) to its corresponding intensity value a_i, and where $\mathbf{w}(\mathbf{u}_i, \mathbf{s}_0, \mathbf{s})$ is a control-point based warping function, defined by the landmark point correspondences between $\mathbf{s}_0$ and $\mathbf{s}$, which maps a pixel location $\mathbf{u}_i$ in the normal domain to its warped location in the image domain. To increase readability, we re-write more compactly Eq. 2.7 for the sampling of a normal texture $\mathbf{a}$ from an image I given a face shape $\mathbf{s}$, as

$$\mathbf{a} = \mathbf{I}(\mathbf{W}_{\mathcal{S}_0}(\mathbf{s})), \tag{2.8}$$

where $\mathbf{W}_{\mathcal{S}_0}(\mathbf{s})$ is a warping function, defined by the landmark point correspondences between $\mathbf{s}_0$ and $\mathbf{s}$, which maps all the locations of the K normal pixels in $\mathcal{S}_0$ to their warped locations in the image domain, and where $\mathbf{I}$ is a vector function, $\mathbf{I} : \mathbb{R}^{2K} \rightarrow \mathbb{R}^K$, which maps K pixel locations in the image domain to their K corresponding intensity values $\mathbf{a} = [a_1; \ldots; a_K]$.

Similarly to a PDM, the TDM texture parameters in Λ are partitioned in two parameter subsets to separately account for global and local texture variation, i.e., in vectorial form, $\boldsymbol{\lambda} = [\boldsymbol{\lambda}^{(g)}; \boldsymbol{\lambda}^{(l)}]$, where $\boldsymbol{\lambda}^{(g)}$ accounts for global texture variation, and $\boldsymbol{\lambda}^{(l)}$ accounts for local texture variation. Global texture variation parameters include illumination gain and bias, denoted γ and δ, respectively, i.e., $\boldsymbol{\lambda}^{(g)} = [\gamma; \delta]$. These parameters represent represent a simple linear transform in the pixel intensities, and can also be used to map any fixed texture vector $\mathbf{a}$ to another one $\mathbf{a}'$, as

$$\boldsymbol{\lambda}^{(g)}, \mathbf{a} \;\mapsto\; \mathbf{a}' = \gamma \mathbf{a} + \delta \tag{2.9}$$

The TDM local texture parameters $\boldsymbol{\lambda}^{(l)}$ are meant to control any type of texture variation that cannot be described by global illumination gain or bias. Again similarly to a PDM, the quantities used to define and manipulate the local texture parameters are obtained by statistical analysis on a set of examples. Let $\{(I_j, \mathbf{s}_j)\}, j \in \{1, \ldots, T\}$, be a set of example face shape-annotated images. By repeatedly applying the sampling function in Eq. 2.8, a set $\{\mathbf{a}_j\}, j \in \{1, \ldots, T\}$, of example face textures are obtained, which are dense and shape-normalized. A Procrustes analysis is used [63], so that the difference in illumination gain and bias between the example textures $\{\mathbf{a}_j\}$ is removed. Specifically, each example texture $\mathbf{a}_j$ is transformed into a corresponding texture $\mathbf{a}'_j$ using the linear transform in Eq. 2.9, so that it is globally equivalent to a common reference face texture $\mathbf{a}_0$, i.e.,

$$\mathbf{a}'_j = \hat{\gamma}_j \mathbf{a}_j + \hat{\delta}_j \overset{g}{=} \mathbf{a}_0, \quad \forall j \in \{1, \ldots, T\}, \tag{2.10}$$

where $\hat{\gamma}_j$ and $\hat{\delta}_j$ denote the optimal global illumination gain and bias, respectively, used to align

FIGURE 2.4: The local part of a TDM for the dense and shape-normalized face texture, as a subspace model built by PCA. The reference face texture $\mathbf{a}_0$ is shown to the left, and to the right are color-based depictions of the first three modes of local texture variation $\boldsymbol{\tau}_1$, $\boldsymbol{\tau}_2$, and $\boldsymbol{\tau}_3$. This illustration is from [65].

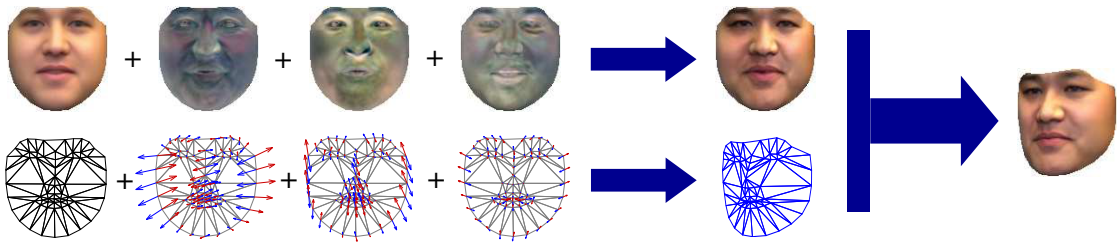


FIGURE 2.5: A photorealistic face image is generated (right) by projecting the texture generated by a TDM (upper line) onto the shape generated by a PDM (lower line), using a warping function, which is roughly the inverse of the function $\mathbf{W}_{\mathcal{S}_0}$ used in Eq. 2.8. This illustration is from [65].

$\mathbf{a}_j$ to $\mathbf{a}_0$ and give $\mathbf{a}'_j$, and $\overset{g}{=}$ denotes the global equivalence between two textures. PCA is then applied to the set of aligned face textures $\{\mathbf{a}'_j\}$ [64], which eventually gives a linear subspace model for the (globally) aligned, dense, and shape-normalized face texture, as the mapping

$$\begin{aligned} \mathbf{a}' : \quad \boldsymbol{\lambda}^{(l)} \;\mapsto\; \mathbf{a}_0 + \mathbf{T}\boldsymbol{\lambda}^{(l)} = \mathbf{a}_0 + \sum_{i=1}^{M} \lambda_i^{(l)} \boldsymbol{\tau}_i, \\ \mathbf{T} = [\boldsymbol{\tau}_1 \dots \boldsymbol{\tau}_M] \in \mathbb{R}^{K \times M}, \quad \boldsymbol{\lambda}^{(l)} = [\lambda_1^{(l)}; \dots; \lambda_M^{(l)}], \end{aligned} \tag{2.11}$$

where $\mathbf{a}_0$ is the reference face texture used in Eq. 2.10, here corresponding to the origin of the subspace, and $\mathbf{T}$ is an orthogonal basis containing linear modes of local texture variation, which consist of the M principal components calculated on the globally aligned face textures. Together, the linear transform defined in Eq. 2.9 and the subspace model defined in Eq. 2.11 give the complete form of the TDM mapping defined in Eq. 2.6, as

$$\mathbf{a} : \quad \boldsymbol{\lambda} = [\boldsymbol{\lambda}^{(g)}; \boldsymbol{\lambda}^{(l)}] \;\mapsto\; \gamma \mathbf{a}'(\boldsymbol{\lambda}^{(l)}) + \delta = \gamma(\mathbf{a}_0 + \mathbf{T}\boldsymbol{\lambda}^{(l)}) + \delta. \tag{2.12}$$

The total number of TDM parameters is $2 + M$, which typically represents much less DoF than the K DoF corresponding to the pixel intensities of a face texture. This lower dimensionality favors the production of valid, realistic face textures. Figure 2.4 shows a subspace model built by PCA, representing the local part of a TDM for the shape-normalized, dense face texture. Figure 2.5 illustrates how a PDM and a TDM can be used to generate a photorealistic image of the face.

Minimizing the image difference function

AAMs consider face alignment as the problem of maximizing the similarity between (1) the target image and (2) a synthetic image of the face generated using a PDM and a TDM. Face alignment with AAMs can therefore be seen as an instance of the problem of deformable template matching, which has been studied in length for decades in computer vision, and for which a comprehensive methodological compendium can notably be found in [68]. In general, deformable template matching is posed as an image difference minimization, which is typically a non-linear least-squares problem solved by numerical optimization. In particular in AAMs, given a target image I, the image difference function E to be minimized with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ is defined as

$$E(\boldsymbol{\theta}, \boldsymbol{\lambda}) = || \, \mathbf{a}(\boldsymbol{\lambda}) \, - \, \mathbf{I}(\mathbf{W}_{\mathcal{S}_0}(\mathbf{s}(\boldsymbol{\theta}))) \, ||^2, \tag{2.13}$$

where $\mathbf{a}(\boldsymbol{\lambda})$ is a TDM (Eq. 2.6 and 2.12), $\mathbf{s}(\boldsymbol{\theta})$ is a PDM (Eq. 2.1 and 2.5), and $\mathbf{I}(\mathbf{W}_{\mathcal{S}_0}(\cdot))$ is an image sampling function for I involving a warping function from the normal, reference domain to the image domain (see previous section and Eq. 2.8). It is proposed in AAMs to use a Gauss-Newton algorithm to solve the optimization problem. Depending on the manner in which Eq. 2.13 is reparameterized to prepare the first order approximation, different Gauss-Newton algorithms can be obtained, which offer different possible ways to trade correctness off for time-efficiency. We are interested in particular in an "inverse compositional" reparameterization of the problem, which leads to a rather efficient Gauss-Newton algorithm with close to no tradeoff on its analytical correctness. The resulting algorithm is called the simultaneous inverse compositional AAM (SICAAM [21]). In SICAAMs, minimizing the function in Eq. 2.13 is reparameterized as

$$\underset{\Delta\boldsymbol{\theta}, \Delta\boldsymbol{\lambda}}{\operatorname{argmin}} \, || \, \mathbf{A}_{\boldsymbol{\lambda}+\Delta\boldsymbol{\lambda}}(\mathbf{W}_{\mathcal{S}_0}(\mathbf{s}(\Delta\boldsymbol{\theta}))) \, - \, \mathbf{I}(\mathbf{W}_{\mathcal{S}_0}(\mathbf{s}(\boldsymbol{\theta}))) \, ||^2, \tag{2.14}$$

where $\boldsymbol{\lambda}$ and $\boldsymbol{\theta}$ are fixed, $\Delta\boldsymbol{\lambda}$ and $\Delta\boldsymbol{\theta}$ are the "increment parameter" variables upon which to optimize, and $\mathbf{A}_{\boldsymbol{\lambda}+\Delta\boldsymbol{\lambda}}$ is a shape-normalized synthetic image formed by setting the K intensity values generated by $\mathbf{a}(\boldsymbol{\lambda} + \Delta\boldsymbol{\lambda})$ to their K corresponding normal locations $\{\mathbf{u}_i\}$ in $\mathcal{S}_0$. The linear approximation of Eq. 2.14 gives closed-form solutions for $\Delta\boldsymbol{\lambda}$ and $\Delta\boldsymbol{\theta}$, which can be efficiently calculated, since the warp Jacobian $\partial\mathbf{W}_{\mathcal{S}_0}/\partial\boldsymbol{\theta}$ is always evaluated at the constant point $\mathbf{0}$ and can therefore be precomputed, as well as the synthetic image gradients $\nabla\mathbf{A}_0, \nabla\mathbf{A}_1, \ldots, \nabla\mathbf{A}_M$ corresponding to the constant texture vectors $\mathbf{a}_0, \boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_M$, respectively. At each iteration, after the efficient calculation of $\Delta\boldsymbol{\lambda}$ and $\Delta\boldsymbol{\theta}$, the SICAAM update equations for $\boldsymbol{\lambda}$ and $\boldsymbol{\theta}$ are

$$\begin{aligned} \boldsymbol{\lambda} &\leftarrow \boldsymbol{\lambda} + \Delta\boldsymbol{\lambda}, \\ \mathbf{W}_{\mathcal{S}_0}(\mathbf{s}(\boldsymbol{\theta})) &\leftarrow \mathbf{W}_{\mathcal{S}_0}(\mathbf{s}(\boldsymbol{\theta})) \circ \mathbf{W}_{\mathcal{S}_0}^{-1}(\mathbf{s}(\Delta\boldsymbol{\theta})). \end{aligned} \tag{2.15}$$

In Eq. 2.15, the texture parameters $\boldsymbol{\lambda}$ are updated in the conventional additive way. However one can see why this algorithm is called "inverse compositional" by looking at the shape parameter update part in Eq. 2.15. Indeed, $\mathbf{W}_{\mathcal{S}_0}^{-1}(\mathbf{s}(\Delta\boldsymbol{\theta}))$ is an incremental inverse warp, that is to be composed with the current forward warp $\mathbf{W}_{\mathcal{S}_0}(\mathbf{s}(\boldsymbol{\theta}))$ for updating the shape parameters $\boldsymbol{\theta}$. To the first order in $\Delta\boldsymbol{\theta}$, we have $\mathbf{W}_{\mathcal{S}_0}^{-1}(\mathbf{s}(\Delta\boldsymbol{\theta})) \approx \mathbf{W}_{\mathcal{S}_0}(\mathbf{s}(-\Delta\boldsymbol{\theta}))$, which is a good approximation for the usually small shape parameter increments calculated within the iterative optimization procedure.

The "simultaneous" part of the SICAAM appellation comes from the fact that both the shape and texture parameters are optimized by the Gauss-Newton algorithm. In contrast, an inverse compositional AAM formulation that avoids including the texture parameters in the optimization problem has been proposed, which "projects out" the texture part from the problem (project-out inverse compositional AAM, or POICAAM, [65]). A POICAAM is typically very time-efficient, but its "projecting out" of the texture part is a somewhat abusive approximation which impacts the robustness of the alignment procedure in terms of generalization capabilities to new faces [21]. We therefore decided to use a SICAAM for face alignment in our application.

Algorithmically, face alignment is considered to be achieved in AAMs when the iterative optimization procedure implemented by the Gauss-Newton algorithm meets a convergence criterion. For example, convergence can be declared when the residual error coming from the minimization of the objective function in Eq. 2.13 goes below some threshold, or when, for some iterations, the difference becomes little enough between, e.g., consecutive residual errors, shape parameters, or landmark point locations. The exact form and threshold(s) set for the convergence criterion are typically chosen empirically.

Adding robustness to face alignment

In videos made to be used in sign language recognition, signers are compliant to clearly display their hands and face to the camera, and a good control of the illumination conditions can be expected. However, several difficulties remain, which are inherently bound to the production of a natural and good-quality communication in sign language. Indeed, a signer's face is oftentimes (1) very expressive, (2) non-frontal, and (3) occluded by the hands, as some signs are required to be performed by touching the face or wave the hands in front of it. In order to add robustness against these cases in our application, we equip the AAM alignment procedure with some refinements proposed in the literature.

Controlling the alignment procedure with shape regularization is particularly helpful in the case of extreme facial expressions, because strong local shape deformations emphasize warping inaccuracies, which deteriorate the quality of the shape-normalized face texture used in AAMs. Shape regularization can be performed by adding a shape deformation penalty to the objective function in Eq. 2.13, on the basis of the variances $\{\sigma_i^2\}, i \in \{1, \ldots, N\}$, observed for the local shape parameters $\mathbf{\theta}^{(l)}$ when building the PDM from example face shapes by PCA. Indeed, with a zero-mean Gaussian distribution assumed for $\mathbf{\theta}^{(l)}$, it is customary to minimize $E(\mathbf{\theta}, \mathbf{\lambda}) + C \,||\, \mathbf{\theta}^{(l)} \,||^2_{\mathbf{\Sigma}^{-1}}$, where the square of the Mahalanobis distance from $\mathbf{0}$ to $\mathbf{\theta}^{(l)}$ is used with the covariance matrix $\mathbf{\Sigma} = \mathrm{diag}\{[\sigma_1^2; \ldots; \sigma_N^2]\}$, and C is a factor balancing the effect of the soft constraint. This regularization however has the disadvantage that it encourages the local shape parameters $\mathbf{\theta}^{(l)}$ to tend to $\mathbf{0}$, and thus the face shape to be close to the reference shape $\mathbf{s}_0$, up to a similarity transformation. In our application, it is better to allow more flexibility in local shape deformation within the alignment procedure. We therefore assume that the distribution of $\mathbf{\theta}^{(l)}$ is uniform within a hyperellipsoidal validity region, with zero probability elsewhere, and we express this assumption as a hard constraint in the minimization of the objective function in Eq. 2.13, i.e.,

we search

$$\underset{\boldsymbol{\theta},\boldsymbol{\lambda}}{\operatorname{argmin}} E(\boldsymbol{\theta},\boldsymbol{\lambda}) \quad \text{subject to (s.t.)} \quad \sum_{i=1}^{N} \frac{\theta_i^{(l)2}}{\sigma_i^2} \leq V, \tag{2.16}$$

where the validity region boundary threshold V is chosen empirically.

To robustly deal with non-frontal faces, i.e., large off-image plane head rotations, we use the 2D+3D AAM method proposed in [69]. This method adds a soft constraint term to Eq. 2.13 that encourages the 2D face shape to be a valid projection of a 3D face shape generated by a 3D PDM. We build a 3D PDM for the face by applying a nonrigid structure from motion technique (NRSfM, [70]) to the set of 2D example face shapes $\{\mathbf{s}_j\}, j \in \{1,\ldots,T\}$. The model obtained by NRSfM is composed of a 3D reference face shape, denoted $\bar{\mathbf{s}}_0$, and an orthogonal basis of linear modes of 3D local shape deformation, denoted $\bar{\boldsymbol{\Phi}}$, similar to the linear subspace model in Eq. 2.4. This 3D model can be used to produce a 3D face shape, as $\bar{\mathbf{s}}' = \bar{\mathbf{s}}_0 + \bar{\boldsymbol{\Phi}}\bar{\boldsymbol{\theta}}^{(l)}$, for any values of the 3D local shape parameters $\bar{\boldsymbol{\theta}}^{(l)}$ (see Fig. 2.6). Such 3D shape can also be moved and rotated globally in 3D and projected to 2D with a transform $\mathbf{P}_{\bar{\boldsymbol{\theta}}^{(g)}}$, where the parameters $\bar{\boldsymbol{\theta}}^{(g)}$ account for the 3D rotation and 3D translation of the whole shape, and the projection parameters are fixed *a priori* and therefore not mentioned. With the addition of the 2D+3D soft constraint term, the face alignment procedure also involves minimizing over the global and local 3D face shape parameters $\bar{\boldsymbol{\theta}} = [\bar{\boldsymbol{\theta}}^{(g)}; \bar{\boldsymbol{\theta}}^{(l)}]$, and Eq. 2.16 thus becomes

$$\underset{\boldsymbol{\theta},\bar{\boldsymbol{\theta}},\boldsymbol{\lambda}}{\operatorname{argmin}} E(\boldsymbol{\theta},\boldsymbol{\lambda}) + C \,||\, \mathbf{s}(\boldsymbol{\theta}) - \mathbf{P}_{\bar{\boldsymbol{\theta}}^{(g)}}(\bar{\mathbf{s}}_0 + \bar{\boldsymbol{\Phi}}\bar{\boldsymbol{\theta}}^{(l)}) \,||^2 \quad \text{s.t.} \quad \sum_{i=1}^{N} \frac{\theta_i^{(l)2}}{\sigma_i^2} \leq V, \tag{2.17}$$

where C is an empirical factor balancing the effect of the 2D+3D soft constraint.

Finally, to be robust against the occlusions of the face by the hands (or by any other occluding object), we wrap with a robust M-estimator the residual intensity differences between the target image and the synthetic image. Specifically, we replace the nonrobust L_2 loss function used in Eq. 2.13, with the robust Huber loss function [68], which acts as the L_2 loss function for residuals below a threshold (the inliers), but downweights the residuals above that threshold (the outliers). In our case, the outliers correspond to the target image pixels that are likely to come from an occlusion. Replacing the L_2 loss function with the Huber loss function leads to an iteratively reweighted least squares problem. The complete form of our minimization problem is

$$\underset{\boldsymbol{\theta},\bar{\boldsymbol{\theta}},\boldsymbol{\lambda}}{\operatorname{argmin}} E_\rho(\boldsymbol{\theta},\boldsymbol{\lambda}) + C \,||\, \mathbf{s}(\boldsymbol{\theta}) - \mathbf{P}_{\bar{\boldsymbol{\theta}}^{(g)}}(\bar{\mathbf{s}}_0 + \bar{\boldsymbol{\Phi}}\bar{\boldsymbol{\theta}}^{(l)}) \,||^2 \quad \text{s.t.} \quad \sum_{i=1}^{N} \frac{\theta_i^{(l)2}}{\sigma_i^2} \leq V, \tag{2.18}$$

where $E_\rho(\boldsymbol{\theta},\boldsymbol{\lambda})$ is the robust image difference term which involves the Huber loss function ρ. This term, which is to be compared by the reader to the nonrobust term in Eq. 2.13, is defined as

$$E_\rho(\boldsymbol{\theta},\boldsymbol{\lambda}) = \sum_{i=1}^{K} \rho(r_i(\boldsymbol{\theta},\boldsymbol{\lambda})), \quad \text{with} \quad \mathbf{r}(\boldsymbol{\theta},\boldsymbol{\lambda}) = \mathbf{a}(\boldsymbol{\lambda}) - \mathbf{I}(\mathbf{W}_{\mathcal{S}_0}(\mathbf{s}(\boldsymbol{\theta}))), \tag{2.19}$$

where $r_i(\boldsymbol{\theta},\boldsymbol{\lambda})$ is the i^{th} residual of the difference vector $\mathbf{r}(\boldsymbol{\theta},\boldsymbol{\lambda})$ between the synthetic image and the warped target image, obtained with the TDM and PDM parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\theta}$, respectively.

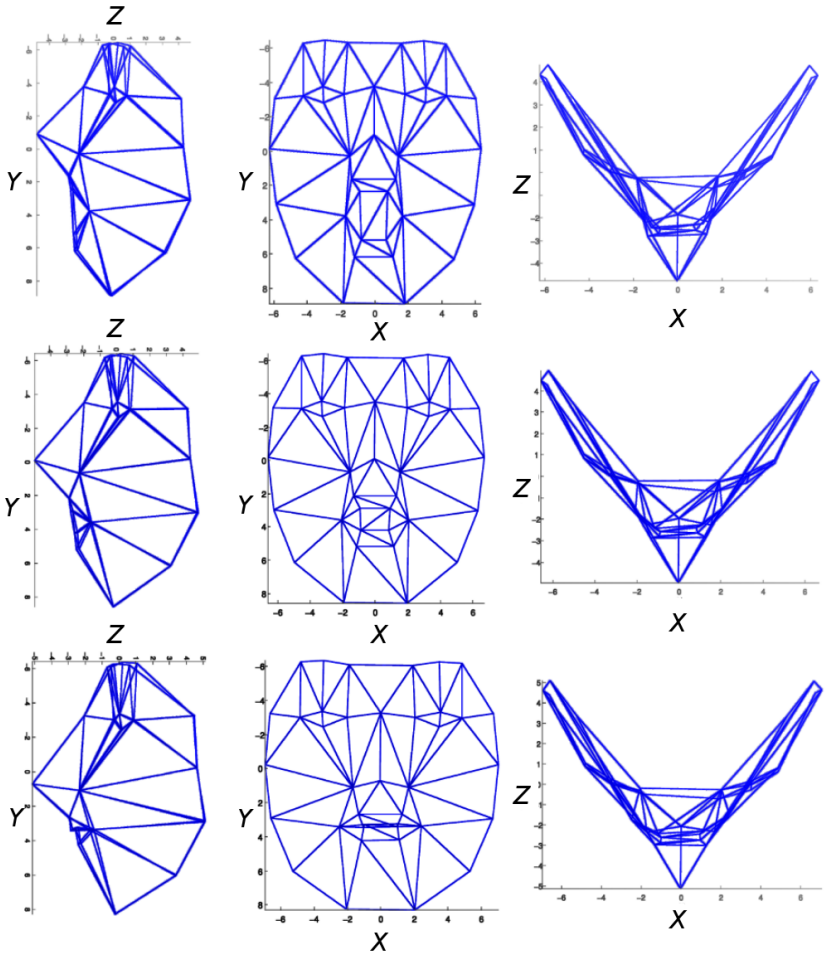


FIGURE 2.6: Illustration of a 3D face PDM obtained by nonrigid structure from motion applied to a set of 2D example face shapes. The left, middle, and right columns show the projections on the YZ, XY, and XZ planes, respectively. The middle row shows the 3D reference shape $\bar{\mathbf{s}}_0$. The top and bottom rows show the action on $\bar{\mathbf{s}}_0$ of adding, resp. subtracting, the first mode of 3D local shape deformation. One can see that this deformation mode is mostly acting on the opening of the mouth.

Continuous face alignment in a video

To our knowledge, all effective face alignment methods, including AAMs, work by iteratively moving the facial landmark points starting from some given, initial, and likely misplaced locations. The overall robustness of iterative face alignment methods very much depends on the condition that such initial locations of the landmark points are in the vicinity of their optimal locations. The maximal initial misplacement of the landmark points depends on the method, and is typically empirically studied. In the scenario of face alignment to a single image, obtaining good initial landmark point locations is generally delegated to a face detection method, which is assumed to have a good robustness itself (e.g., [17]). It is expected that the face detection method accurately delimits the image region where the face is present, so that coarsely inscribing a reference face shape within this region would give good initial landmark point locations. Although it is not always explicitly stated in the literature, most face alignment methods, including AAMs, are designed to work on the hypothesis that a face region is given within the target image, and these methods therefore heavily rely on the existence of robust face detection methods.

When in a tracking scenario, the temporal continuity offered by standard video frame rates provides yet another way to initialize the face alignment procedure within a video frame. Indeed, good initial landmark point locations for the current frame I_t can be obtained as the shape points $\mathbf{s}_{t-1}$ resulting from the alignment to the previous frame I_{t-1}. The face shape $\mathbf{s}_t$ aligned to the current frame I_t can then be passed on to initialize face alignment for the next frame I_{t+1}, and so on. In cases of extreme head poses and/or extreme facial expressions, such initialization strategy based on temporal continuity may be even better suited than the one based on face detection, in particular if the face detection method used is not very robust against non-frontal and/or non-neutral faces. In any case, it is preferable to skip face detection whenever possible for a better overall time-efficiency.

As advantageous as the temporal continuity-based initialization strategy may be, the face detection-based initialization strategy is required for at least the first frame of the video. Additionally, face alignment may fail to fittingly converge for some frames, which situation (1) must be detected, e.g., by using a convergence quality criterion involving I_t and $\mathbf{s}_t$, and (2) invalidates considering $\mathbf{s}_t$ as a good initialization for face alignment in I_{t+1}, since $\mathbf{s}_t$ is considered unfit in I_t. In such case, the face detection-based strategy must be used for as many frames as it takes to find a good fit, before resuming initialization based on temporal continuity.

Adapting a generic face model to a specific face on-the-fly

A holistic PDM such as expressed in Eq. 2.5 has the ability to capture generic face shape deformations quite well. Indeed, a single PDM built from a large and heterogeneous face dataset is capable of producing acceptable face shapes for almost all people, with many different facial expressions, and using a fairly low number of shape parameters. The low-dimensional linear form in Eq. 2.4 has therefore a high representational power in the case of the deformable face shape, and the validity of the generated shapes can furthermore be adequately controlled within the alignment procedure using standard shape regularization techniques. This appealing property is the reason

why PDMs are often met in the literature on face alignment, not only with AAMs, but with other methods as well (e.g., constrained local models, [22]).

However, a holistic TDM (Eq. 2.12) built from a large and heterogeneous dataset for the purpose of being used in generic face alignment typically grows very large in its number of texture parameters, in order to achieve acceptable representations for all example face textures. Additionally, because the face shape to image relationship is nonlinear by nature, the linear form in Eq. 2.11 typically leads to excessive representational power in a generic setup. This notably implies that non-valid face textures cannot be easily prevented by simple assumptions on the texture parameters, such as confining their values to a uniform validity region or considering that they follow a Gaussian distribution [21]. This is overall bad news for generic face alignment with AAMs, since AAMs are based on numerical optimization. Indeed, more texture parameters mean more calculations per iteration, and thus poor time-efficiency. Also, without an appropriate way to apply regularization, a model with too much representational power may settle for local minima when used with first-order optimization techniques like the Gauss-Newton algorithm, and as such may lead to poor convergence properties.

The properties mentioned above, which are unfavorable to AAM-based generic face alignment, lead to the following situation. On the one hand, a person-specific AAM, which uses a TDM built from examples of a single person's face, is very robust and time-efficient to accurately align the face on new images of this same person, but not on images of other people. A person-specific AAM can also handle difficult cases of occlusions, extreme facial expressions, and off-image plane head rotations with the refinements expressed in Eq. 2.18. On the other hand, a generic AAM, i.e., an AAM that uses a TDM built from a heterogeneous dataset, where hundreds of different people and various illumination conditions are present, will fail a significant amount of times to converge to the correct facial landmark point locations in a face image, even when presented with faces of the people from the heterogeneous dataset, and even with the refinements expressed in Eq. 2.18. A generic AAM must also optimize over many texture parameters (in the SICAAM formulation, see Eq. 2.14), and is thus slow to fittingly converge, if it does.

In a scenario where face alignment has to be performed on an isolated image (as opposed to a series of images in a video), AAMs are obviously not the method of choice if the goal is to be generic, i.e., robust against differences in people's faces and varying illumination conditions. However, in a tracking scenario, where multiple images of the same (though potentially unknown) person are presented in a sequence to the face alignment method, a strategy can be devised to overcome the robustness and efficiency limitations of a generic AAM, while still benefiting from the very good performance of a person-specific AAM. Such a strategy was proposed in [62], where the authors focused on solving the template drifting problem within a tracking scenario, and showed how their robust template update strategy could be applied to the case where the template is in the form of an AAM. Basically, this strategy allows to convert a slow and nonrobust generic AAM into a fast and robust person-specific AAM, by re-building its TDM on-the-fly using only face-aligned video frames that passed a strict convergence quality test. Using this strategy quite quickly gives a robust and time-efficient person-specific AAM in our application, where we align the face on consecutive frames of a sign language video. Moreover, if new signers were aware of this strategy and would consider it as a necessary "calibration" stage for using our facial

cue extraction system based on face alignment, we believe that the conversion of a generic AAM to a specific AAM would only take a few seconds, at most.

2.2.2 Extraction of sign language facial cues

We originally developed our facial cue extraction system so that it could be incorporated within a sign language recognition system chain designed during the SignSpeak project [37]. The overall goal of SignSpeak was to create a new, end-to-end, vision-based technology for translating continuous sign language to text in a spoken language, and thus improve the communication between deaf and hearing people. Through discussions with sign language experts who participated in the SignSpeak project, we determined what sort of information about the signer's face is the most useful to consider to perform gloss recognition and, ultimately, sign language translation. Facial communication cues in sign language should ideally convey to which extent

- the mouth is open,
- each of the eyes are open,
- each of the eyebrows are lowered or raised, and
- the head is moved and rotated with respect to the frontal pose.

The elements listed above are essentially related to local and global changes in the signer's face shape. We presented in Sect. 2.2.1 an effective AAM-based face alignment method that is well-suited to robustly track the deformable face shape of a signer within a video. We make the hypothesis that the facial landmark points aligned with this AAM-based tracking method hold most of the necessary shape information we need to extract useful sign language facial cues. The image locations of the aligned facial landmark points are however not trivially equivalent to our facial cues of interest, and some extra modeling and calculations need to be performed. Notably, it is desirable to decouple the landmark point-based face shape information from the signer's identity. Indeed, the shape information pertaining to the personal facial features of the signer is not one we wish to capture, because it is deemed to be irrelevant to the effective glossing of sign language. For instance, the mouth should be recognized as being widely open or totally closed regardless of whether the signer has thin or thick lips, a small or large jaw, and so forth. In other words, we aim at extracting sign language facial cues that are identity-independent, or, equivalently, identity-normalized. Besides identity normalization, we also wish to maximize the exclusivity of the information that each facial cue can provide, i.e., to avoid redundancy. For instance, the mouth opening facial cue should be largely independent of the head rotation facial cues. By avoiding redundancy, we aspire to provide the most useful facial cues to the automatic glossing system that follows within the end-to-end sign language recognition system chain. Indeed, a representation of the sign language facial cues that is not only rich but also concise should favor, in principle, the learning of an effective automatic glossing system.

We emphasize that all material and methods presented in this section are our own. They consist of simple, yet effective ways to extract sign language facial cues, on the basis of the face shape information provided by the AAM-based method presented in the previous section.

Full 3D head pose as facial cues

As a byproduct of the 2D+3D AAM refinement described in Sect. 2.2.1 (originally proposed in [69]), we continuously extract $\bar{\theta}^{(g)}$, i.e., the global shape parameters of the 3D face shape that is used to regularize the 2D face shape being aligned by the AAM, so that the alignment can be robust against large off-image plane head rotations. If the 3D reference face shape $\bar{\mathbf{s}}_0$ of the 3D PDM is designed to be frontal-facing, then the parameters $\bar{\theta}^{(g)}$ directly give the amount of head rotation around each of the X, Y, and Z axes, as well as the head translation along each of these axes, with respect to the frontal pose. We take these six parameters as the head pose facial cues, and provide them continuously throughout the processing of a video of a signer[2]. We assume in the present work that these head pose facial cues are mostly identity-independent, which is true in good approximation.

Normalized apertures as facial cues

In addition to the head pose facial cues, we wish to extract five more facial cues to convey (1) the mouth opening degree, (2) the left eye opening degree, (3) the right eye opening degree, (4) the left eyebrow raising degree, and (5) the right eyebrow raising degree, respectively. We see these facial cues as being closely related in nature by the fact that they represent an aperture, i.e., they share the key property of representing a visual gap, with an amplitude that varies between two extrema. Therefore, in the following, we collectively refer to the mouth/eye opening and eyebrow raising degrees as aperture facial cues, and we describe a general method for extracting an aperture facial cue from the face of a signer in a video, on the basis of the modeling choices given below.

1. An aperture facial cue is adequately represented by a scalar quantity that takes its values in the bounded interval $[0, 1]$, where the value 0 corresponds to "closed", or "maximally lowered", and the value 1 corresponds to "maximally open", or "maximally raised", irrespective of the signer's head pose or identity.

2. Given prior normalization information about the head pose and the identity of a signer, an accurate measurement of an aperture facial cue can be derived in closed form from the instantaneous (i.e., frame-by-frame) configurations of two well-chosen local subsets of facial landmark points, obtained with the AAM-based alignment method described in Sect. 2.2.1.

To normalize the 2D face shape with respect to the head pose, while retaining information about the local shape changes, we again exploit the 2D+3D refinement incorporated in our AAM-based face alignment method (see Sect. 2.2.1). Indeed, for each 2D face shape $\mathbf{s}$ aligned to the signer's face, this refinement gives us the corresponding 3D face shape $\bar{\mathbf{s}}$, which is parameterized by the 3D global and local shape parameters $\bar{\theta}^{(g)}$ and $\bar{\theta}^{(l)}$, respectively. The 3D-normalized 2D face shape $\mathbf{s}^*$ corresponding to $\bar{\mathbf{s}}$ is then obtained by setting to $\mathbf{0}$ the 3D global shape parameters

[2]Note that, without the 2D+3D AAM refinement, the head pose is still recoverable. Indeed, in an early version of our facial cue extraction system, we used the POSIT algorithm [71], which, while being less accurate than the 2D+3D AAM refinement, allows one to recover the full head pose on the basis of the correspondences between the 2D facial landmark points and a fixed 3D structure of the face shape, defined *a priori*.

$\bar{\theta}^{(g)}$, and projecting the resulting 3D shape to 2D, i.e.,

$$\mathbf{s}^* = \mathbf{P_0}(\bar{\mathbf{s}}_0 + \bar{\mathbf{\Phi}}\bar{\theta}^{(l)}), \tag{2.20}$$

where $\bar{\mathbf{s}}_0$ is the frontal-facing 3D reference face shape, $\bar{\mathbf{\Phi}}$ is the orthogonal basis of linear modes of 3D local shape deformation, $\bar{\theta}^{(l)}$ are the 3D local shape parameters, and $\mathbf{P_0}$ simply applies the projective transformation, without any 3D rotation or 3D translation being involved[3].

Temporarily letting aside the matter of normalization with respect to the signer's identity, we now introduce a 3D-normalized distance measure between two subsets of facial landmark points, on the basis of the 3D-normalized 2D face shape given in Eq. 2.20. Let $\mathcal{L} = \{(\mathbf{x}_i, c_i) : i \in \mathcal{I}\}$ be a set of facial landmark points indexed by the set $\mathcal{I} = \{1, \ldots, L\}$. As a reminder, c_i represents an arbitrary semantic concept associated to the 2D point $\mathbf{x}_i$, e.g., "the left eye upper eyelid mid-point". Let $\mathcal{I}_A \subseteq \mathcal{I}$ and $\mathcal{I}_B \subseteq \mathcal{I}$ be two subsets of indices. The 3D-normalized distance d^* between the landmark point subsets $\mathcal{L}_A = \{(\mathbf{x}_i, c_i) : i \in \mathcal{I}_A\}$ and $\mathcal{L}_B = \{(\mathbf{x}_i, c_i) : i \in \mathcal{I}_B\}$ is calculated as

$$d^*(\mathcal{L}_A, \mathcal{L}_B) = \left\| \frac{1}{|\mathcal{L}_A|} \sum_{i \in \mathcal{I}_A} \mathbf{x}_i^* - \frac{1}{|\mathcal{L}_B|} \sum_{i \in \mathcal{I}_B} \mathbf{x}_i^* \right\|, \tag{2.21}$$

i.e., it is the L_2 norm of the difference between the centroids of the 3D-normalized 2D point subsets corresponding to $\mathcal{L}_A$ and $\mathcal{L}_B$. Indeed, the points $\{\mathbf{x}_i^*\}$ are in correspondence with the points $\{\mathbf{x}_i\}$ used in the set $\mathcal{L}$, through the 3D-normalized 2D face shape $\mathbf{s}^* = [\mathbf{x}_1^*; \ldots; \mathbf{x}_L^*]$ calculated using Eq. 2.20.

In our aperture facial cue extraction method, we use the 3D-normalized distance measure defined in Eq. 2.21 to represent the amplitude of the gap (i.e., the aperture) between two face regions (embodied by two facial landmark point subsets). One could therefore qualify our method of point set distance-based. Specifically, extracting our aperture facial cues of interest with this method first requires the definition, for each aperture facial cue, of a pair of index subsets, $\mathcal{I}_A$ and $\mathcal{I}_B$, from the set $\mathcal{I}$ indexing the facial landmark points used within the face alignment method (e.g., the AAM-based method in Sect. 2.2.1). For instance, the first (resp. second) index subset associated to the mouth opening degree may appropriately include the upper (resp. lower) lip point indices[4]. As a complete illustration, Table 2.1 gives five pairs of index subsets that designate workable point subsets for the effective extraction of our five aperture facial cues of interest from the facial landmark points presented in Fig. 2.2.

The quantities obtained with our 3D-normalized distance measure are identity-dependent, i.e.,

[3] Without the 2D+3D AAM refinement, the 3D-normalized 2D face shape can still be estimated. The steps of this alternative method, which we used in an early version of our facial cue extraction system, are as follows. (1) Form a 3D version $\bar{\mathbf{s}}_0$ of the 2D reference face shape $\mathbf{s}_0$, extruding it in the Z-axis according to a fixed hand-crafted 3D face structure. (2) Rotate and translate $\bar{\mathbf{s}}_0$ using the 3D rotation and 3D translation parameters obtained by POSIT. (3) Augment $\mathbf{s}$ with the Z-coordinates of the rotated and translated $\bar{\mathbf{s}}_0$, so as to obtain an approximate 3D face shape $\bar{\mathbf{s}}$. Finally, (4) apply to $\bar{\mathbf{s}}$ the inverse 3D rotation, the opposite 3D translation, and the projective transform, to get the 3D-normalized shape $\mathbf{s}^*$.

[4] It is useful to recall that the set of facial landmark points is designed arbitrarily: it is composed of an arbitrary number of 2D points, associated with equally arbitrary semantic concepts, meant to represent the locations deemed to be of interest within the signer's face. For extracting our aperture facial cues of interest with our point set distance-based method, the design of the facial landmark point set should have been made with focus on the lips, eyelids, and eyebrows, which is actually often the case in practice.

TABLE 2.1: This table gives, for each of our five aperture facial cues of interest, the pairs of index subsets $\mathcal{I}_A$ and $\mathcal{I}_B$ used to extract these facial cues with our point set distance-based method, for signers in the RWTH-PHOENIX-Weather sign language recognition dataset [39]. The set of facial landmark points proposed for this dataset counts 38 points, here indexed by the set $\mathcal{I} = \{0, \ldots, 37\}$. We recommend that the reader consult Fig. 2.2 while examining this table.

Aperture facial cue	First index subset $\mathcal{I}_A \subseteq \mathcal{I}$	Second index subset $\mathcal{I}_B \subseteq \mathcal{I}$
mouth opening degree	$\{19, 20, 24, 25\}$ = upper lip	$\{22, 23, 26, 27\}$ = lower lip
left eye opening degree	$\{7\}$ = left upper eyelid	$\{9\}$ = left lower eyelid
right eye opening degree	$\{11\}$ = right upper eyelid	$\{13\}$ = right lower eyelid
left eyebrow raising degree	$\{0, 1, 2\}$ = left eyebrow	$\{6, 8\}$ = left eye corners
right eyebrow raising degree	$\{3, 4, 5\}$ = right eyebrow	$\{10, 12\}$ = right eye corners

they may vary significantly when different signers are presented with our facial cue extraction system. In order to normalize these quantities with respect to identity, shape-based information pertaining to the personal facial features has to be either given *a priori*, or automatically discovered by the system. There are three cases to consider, which we list below in the order of increasing incertitude about the signer's identity.

1. The signer's identity is given to the system before the extraction procedure, and the system possesses useful normalization information about this specific signer.

2. The signer's identity is not given to the system before the extraction procedure, yet the system possesses useful normalization information about this specific signer.

3. The signer's identity is totally unknown by the system. He/she is a new signer about whom no identity normalization information is available.

The three cases listed above imply fundamentally different sub-problems. In the first case, it suffices to equip the facial cue extraction system with a lookup mechanism in order to retrieve the stored normalization information specific to the signer. The second case requires to solve the problem of face recognition to retrieve the appropriate signer-specific normalization information. The third, and most difficult case corresponds to a generic identity normalization problem, where no signer-specific normalization information is available. In the present work, we only consider the first and third cases. In other words, we propose solutions for the first and third cases, and treat instances of the second case as instances of the generic identity normalization problem. Note, however, that nothing prevents one to plug a face recognition system to our facial cue extraction system, in order to recognize known signers in anonymous videos, and thus treat instances of the second case with more accuracy.

Let us start with the design of a solution for the first and simplest case, i.e., the case where the signer's identity is a given during the extraction procedure, and specific identity normalization information about this signer is available within the system. In this situation, we only need to have modeled the specific identity normalization information that is to be retrieved, and to use it appropriately toward removing the specific identity component from the aperture facial cues. At

the time when the PDM used in our AAM-based face alignment method is built (see Sect. 2.2.1), we have access to a number of example face shapes for each specific signer, the identity of whom may later be communicated to the extraction system in order to retrieve normalization information. Let $\{\mathbf{s}_j\}$ be a set of example face shapes specific to a signer, annotated according to a landmark point set $\mathcal{L}$ indexed by a set $\mathcal{I}$. For each aperture facial cue of interest, we calculate for this signer the minimal and maximal aperture values over the set $\{\mathbf{s}_j\}$, as

$$d_m^* = \min\{d_j^*(\mathcal{L}_A, \mathcal{L}_B)\} \ , \quad d_M^* = \max\{d_j^*(\mathcal{L}_A, \mathcal{L}_B)\}, \tag{2.22}$$

respectively, where the aperture value for an example face shape $\mathbf{s}_j$ is given by the 3D-normalized distance[5] $d_j^*(\mathcal{L}_A, \mathcal{L}_B)$, according to Eq. 2.21. It is assumed that, for each aperture facial cue of interest, a pair of index subsets $\mathcal{I}_A \subseteq \mathcal{I}$ and $\mathcal{I}_B \subseteq \mathcal{I}$ have been defined to designate the face regions involved in its calculation (which regions are embodied by the subsets $\mathcal{L}_A$ and $\mathcal{L}_B$). Then, the minimal and maximal values for each aperture facial cue are stored for the specific signer and used at the time of extraction, so as to obtain an identity-normalized aperture facial cue, as

$$\frac{d^*(\mathcal{L}_A, \mathcal{L}_B) - d_m^*}{d_M^* - d_m^*}. \tag{2.23}$$

For a specific signer and a particular aperture facial cue, the minimal and maximal aperture values (d_m^* and d_M^*, respectively) represent all of the information needed to normalize the aperture facial cue of interest with respect to the signer's identity. These extremum values are constants that need to be estimated only once, at the same time as the AAM construction procedure, thus prior to the facial cue extraction procedure. Due to computational inaccuracies, and to the free tuning of the shape regularization constraints applied within the AAM-based face alignment method (see Sect. 2.2.1), it may be that the identity-normalized aperture facial cues obtained using Eq. 2.22 and Eq. 2.23 do not always exactly lie within the interval $[0, 1]$. We found that this minor "overflowing" inconvenience had no impact on the overall accuracy of the gloss recognition system based on such aperture facial cues, and does therefore not need to be addressed.

We now consider the third, most difficult case, where the signer is totally unknown by the facial cue extraction system, i.e., where no reliable signer-specific identity normalization information is available. Note that in this very case where the signer is unknown, we also need to use the adaptive AAM-based face alignment strategy introduced in Sect. 2.2.1 (see also the template update method proposed in [62]). By design of this adaptive face alignment strategy, (1) we are using a generic AAM, built from many different people's faces, and (2) we adapt this generic AAM on-the-fly to a specific AAM, by gathering the aligned faces that passed a strict convergence test within the video being processed. To solve the generic identity normalization problem posed by the third case, we exploit the adaptive AAM-based face alignment framework. For each of our aperture facial cues of interest, provisional generic values for the minimal and maximal apertures are calculated using Eq. 2.22, over the set of all of example face shapes given for the various signers at the time of the

[5] Since in this case we have access to the 2D shape annotation ground truth $\mathbf{s}_j$, a purely shape-based 3D to 2D alignment procedure is applied to obtain the 3D-normalized face shape $\mathbf{s}_j^*$ corresponding to $\mathbf{s}_j$. This 3D to 2D shape optimization procedure does not involve the robust image difference term E_ρ (Eq. 2.19), but only the 2D+3D term. Also, in this optimization, the 2D face shape $\mathbf{s}_j$ is considered a constant.

generic PDM construction. Then, during the facial cue extraction procedure, we conservatively adapt the minimal and maximal values of each aperture facial cue, using a moving average strategy to incorporate the information from the specific face shapes that pass the strict convergence test of the adaptive AAM-based face alignment method. After some time, the minimal and maximal values for each aperture facial cue converge to fit the specific signer identity, and Eq. 2.23 can be used to reliably extract the identity-normalized aperture facial cues. Once again, we believe that this adjustement from the generic to the specific identity normalization information would be a matter of a few seconds in a setup where new signers would be aware of, and participating to the creation of their specific face model (considering this stage as a necessary system calibration).

2.3 Results

We begin this results section with the presentation of typical illustrations of our system output, which should give the reader an idea of the quality of the sign language facial cues extracted with the method proposed in Sect. 2.2.2.

Figure 2.7, which we produced for the last activity report of SignSpeak, shows the results we obtained for a video of a signer from the RWTH-PHOENIX-Weather corpus [39], where we used the facial landmark point set we proposed in Fig. 2.2, and the index subset pairs we proposed in Tab. 2.1 for the aperture facial cues. Figure 2.8, which we produced with an early version of our facial cue extraction system [38], shows the results we obtained for a video of a signer from the *Nederlandse Gebarentaal* corpus (NGT corpus [72]), where we used another facial landmark point set counting 68 points, and other index subset pairs for the aperture facial cues (not detailed in this thesis). Both Fig. 2.7 and Fig. 2.8 depict the head pose facial cues and the aperture facial cues, minus the eyebrow raising degrees for the NGT signer in Fig. 2.8. They also show the facial landmark points used to calculate the facial cues, which landmark points were obtained using the AAM-based face alignment[6] method presented in Sect. 2.2.1.

Next, we present several quantitative evaluations that are indicative of the performance of our facial cue extraction system. Except for the quantitative evaluation of the facial landmark point tracking performance (for the face alignment part of our system), all of these evaluations (1) were conducted by other researchers from the SignSpeak consortium, for SignSpeak activity reports or joint publications during and after the SignSpeak project, and (2) only indirectly showcase the usefulness of our extracted facial cues within a sign language recognition system chain proposed during, or inspired by the SignSpeak project. In any case, all evaluations involved our facial cue extraction system. Also, all evaluations were conducted on the RWTH-PHOENIX-Weather corpus, which we already mentioned several times in this chapter, and to which we contributed 369 face shape-annotated images. We give below a brief description of this richly annotated sign language recognition dataset, before giving the detail of the evaluations.

[6]There are actually a few minor differences in the face alignment method used for processing the videos of the RWTH-PHOENIX-Weather and NGT signers. For the RWTH-PHOENIX-Weather signer in Fig. 2.7, the head pose facial cues and 3D head pose normalization necessary to calculate the aperture facial cues were obtained using the 2D+3D refinement presented in Sect. 2.2.1. However, for the NGT signer in Fig. 2.8, the face alignment results are those of an early version of our system, which did not incorporate the 2D+3D refinement (see [38]). Therefore, the head pose facial cues and 3D head pose normalization for the NGT signer were obtained with another 3D head pose estimation method, namely, the POSIT algorithm [71].

FIGURE 2.7: Extraction of sign language facial cues from a video of a signer from the RWTH-PHOENIX-Weather corpus [39]. The axis system attached to the signer's face gives a visual impression of the rotation around, and translation along, the X-axis (red bar), Y-axis (white bar), and Z-axis (blue bar). The five vertical bars in the bottom part of the images represent, from left to right, the left eyebrow raising degree (in yellow), the left eye opening degree (in blue), the mouth opening degree (in pink), the right eye opening degree (in blue, again), and the right eyebrow raising degree (in yellow, again). Additionally, we show the face shape obtained with our AAM-based face alignment method, triangulated and superimposed on the signer's face (in green).

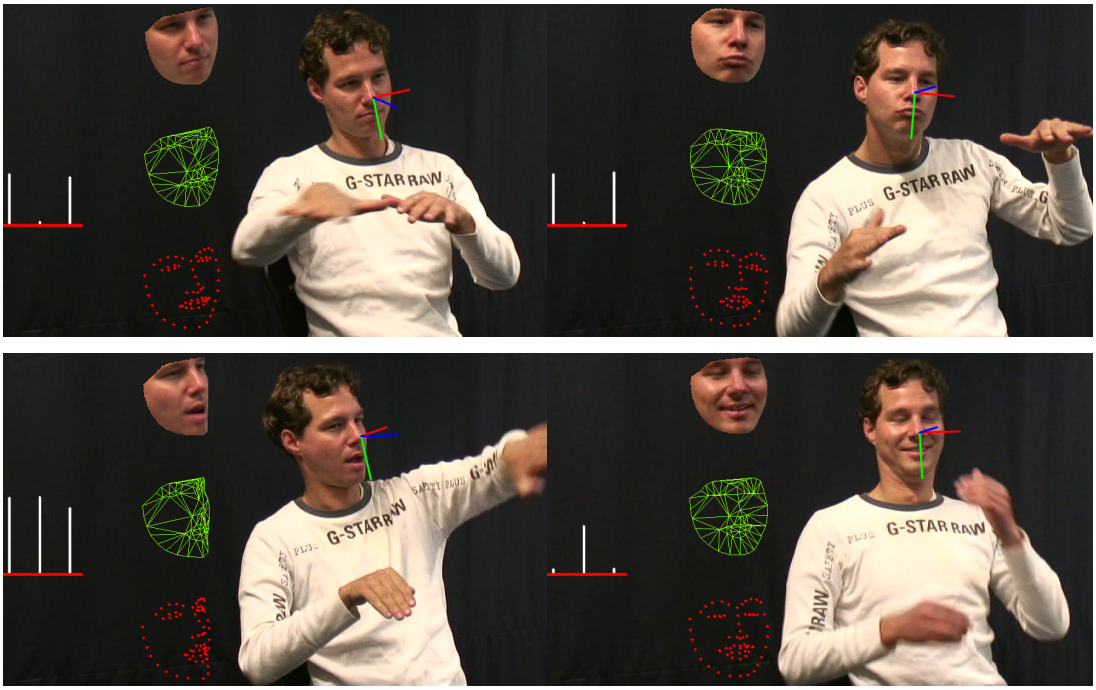


FIGURE 2.8: Extraction of sign language facial cues from a video of a signer from the NGT corpus [72]. The axis system attached to the signer's face gives a visual impression of the rotation around, and translation along, the X-axis (red bar), Y-axis (green bar), and Z-axis (blue bar). The three white vertical bars in the left part of the images represent, from left to right, the left eye opening degree, the mouth opening degree, and the right eye opening degree. In the central region of each image are shown, from top to bottom, the face texture generated with our AAM-based face alignment method, and the face shape obtained with this same method, triangulated (in green), and non-triangulated (red points).

2.3.1 The RWTH-PHOENIX-Weather corpus

The RWTH-PHOENIX-Weather corpus [39] consists of ~195 minutes of video data (293,077 frames at 25 FPS), at 210×260 pixel resolution, recorded over a period of two years (2009-2010), and depicting seven different signers who signed the daily news broadcast program of the German public TV station *Phoenix*. Over this period of time, the seven signers altogether performed 1980 sentences in German sign language (DGS, for *Deutsche Gebärdensprache*), totaling 22,822 running glosses from a vocabulary set of 911 different glosses related to the topic of weather forecasting. A gloss from the vocabulary set of this corpus is repeated ~25 times on average, but glosses with far fewer repetitions exist within the corpus, including several singletons. The videos of the corpus are exhaustively annotated with

1. The gloss-level and sentence-level time boundaries,
2. The gloss descriptions with their pronunciation variants, and
3. The sentence translations in written German.

Additionally, the approximate central points of the left and right hand palms and the nose tip are hand-annotated in a subset of 39,691 video frames of the corpus, taken from videos of all of the seven signers. Facial landmark points are manually annotated for a total of 369 video frames of the corpus, evenly distributed among the seven signers. These 369 face shape annotations are our contribution to the RWTH-PHOENIX-Weather corpus. Indeed, we annotated these face shapes according to our landmark point set described in Fig. 2.2 (see also `https://iis.uibk.ac.at/datasets/phoenix-annotations` for more details).

The RWTH-PHOENIX-Weather corpus is a challenging video-based sign language recognition dataset for several reasons. First, it features continuous sign language, performed by hearing signers under real-time constraints. Because the signers had to translate spoken German announcements to signs in real time, there are numerous occurrences of partly interrupted signs. Second, because of the signing speed due to the live conditions, and because of the low temporal and spatial resolution of the videos, vision-based systems designed to track and extract sign language hand or facial cues have to be especially robust, in particular against motion blur effects due to fast hand movements. Third, some glosses are very scarcely represented (i.e., repeated) within the corpus, down to only one instance (i.e., the gloss singletons), which foretells some difficulties for the learning of a gloss recognition model from this corpus.

2.3.2 Facial landmark point tracking results

Our AAM-based face alignment method described in Sect. 2.2.1 is designed to best perform in a tracking scenario, i.e., with videos. It also incorporates refinements designed to cope with the face alignment difficulties typically encountered in videos of signers, such as extreme facial expressions, large head movements, and occlusions by the hands. To conduct a fair and exhaustive evaluation of this method, we would therefore need ground truth data in the form of sign language videos annotated with tens of facial landmark points in each frame. However, although plenty of face shape-annotated image datasets are available in the public domain, datasets of videos with

such annotations in each frame are much less common, and even less so for videos of signers. We therefore restrict the evaluation of our face alignment method to nose tip landmark point tracking results in videos of signers, for a lack of a more detailed landmark point-based ground truth for sign language video data.

Using the 369 images we annotated for the seven signers of the RWTH-PHOENIX-Weather corpus, we built a face model with the AAM construction method described in Sect. 2.2.1. This face model includes a PDM, i.e., a shape model that controls 38 facial landmark points. According to the zero-based point indices given in Fig. 2.2, the 15^{th} of these landmark points is associated with the nose tip, which also corresponds to the point that has been manually annotated for a subset of 39,691 video frames of the RWTH-PHOENIX-Weather dataset. We call this video subset RWTH-PHOENIX-Nose, for short, and we evaluate our AAM-based face alignment method on videos of RWTH-PHOENIX-Nose only. For each video frame, we picked out the single nose tip landmark point from the 38 automatically aligned landmark points, and compared it to the hand-annotated nose tip ground truth. Formally, we used a tracking error rate (TrackER) measure defined for a landmark point $(\mathbf{x}_i, c_i)$, as

$$\text{TrackER} = \frac{1}{T}\sum_{t=1}^{T}\delta_\tau(\mathbf{x}_i^{(t)}, \hat{\mathbf{x}}_i^{(t)}), \quad \text{with} \quad \delta_\tau(\mathbf{x}_i^{(t)}, \hat{\mathbf{x}}_i^{(t)}) = \begin{cases} 0 & \text{if } ||\mathbf{x}_i^{(t)} - \hat{\mathbf{x}}_i^{(t)}|| < \tau \\ 1 & \text{otherwise,} \end{cases} \tag{2.24}$$

where T is the number of samples (T = 39,691 video frames in our case), i is the index of the considered landmark point with the semantic concept c_i (in our case $i = 15$, and c_{15} represents the nose tip), $\mathbf{x}_i^{(t)}$ is the ground truth landmark point location for the t^{th} sample, $\hat{\mathbf{x}}_i^{(t)}$ is the tracked landmark point location for the t^{th} sample, and τ is an arbitrary tolerance threshold for the Euclidean distance, in pixels, between a ground truth and tracked landmark point locations. Below the tolerance threshold, the tracking for a sample is considered to be a hit ($= 1$), and, above this threshold, to be a miss ($= 0$). Therefore, given some tolerance threshold τ, a TrackER of 0% corresponds to perfect tracking results, and a TrackER of 100% corresponds to the most inaccurate tracking results.

Table 2.2 gives the nose tip tracking results obtained with our AAM-based face alignment method, measured in TrackER, over the ~25 minutes of video in RWTH-PHOENIX-Nose. These results show that our method is quite robust and accurate, since a TrackER of 7.08% for τ = 5 pixels means that the tracking of the nose tip was off by more than 5 pixels for less than two minutes out of ~25 minutes of video. Since the interpupillary distance is of about 30 pixels within RWTH-PHOENIX-Nose, we consider that the tracking succeeded for the vast majority of the video frames. We also observe a TrackER of 0.06% for $\tau = 20$ pixels, which means that the tracking of the nose tip really failed for less than one second out of ~25 minutes of video. To give some comparison, we also include in Tab. 2.2 the TrackER results obtained with a nose tip tracker based on the Viola & Jones method (courtesy of the Human Language Technology and Pattern Recognition group at the university of Aachen), which our AAM-based face alignment method very clearly outperforms.

TABLE 2.2: Nose tip tracking results (in TrackER, see Eq. 2.24) obtained with a nose tip tracker based on the Viola & Jones (V&J) method [17], and with our robust AAM-based face alignment method. Both evaluations were made on the video subset RWTH-PHOENIX-Nose, which counts 39,691 video frames. Our method, which is better suited to robustly track facial landmark points, clearly gives better results than the V&J-based nose tip tracker.

	$\tau = 5$ px	$\tau = 10$ px	$\tau = 15$ px	$\tau = 20$ px
V&J-based nose tip tracker	66.34%	17.21%	7.45%	4.23%
AAM-based face alignment	7.08%	0.43%	0.12%	0.06%

2.3.3 Sign language recognition results

We do not present in this thesis any quantitative evaluation specific to the facial cue extraction method described in Sect. 2.2.2 (qualitative illustrations can however be found in Fig. 2.7, 2.8, and 2.9), and this for two reasons. First, we lack the sort of ground truth data that could be matched appropriately to the sign language facial cues extracted with our system. It is indeed hard and time-consuming to annotate sign language video frames with, e.g., the mouth opening degree normalized with respect to the signer's head pose and identity, and such ground truth data do not exist in the public domain, to the best of our knowledge. Second, the purpose of our facial cue extraction system does not lie in its standalone usage, as if it were a general-purpose facial expression recognition tool, but rather in its integration within a sign language recognition system chain. We believe that a proper insight into the usefulness of our system can be gained through the presentation of the performance achieved with other sign language-related systems, which somehow exploit our extracted facial cues for the purpose of sign language recognition.

Gloss recognition

In this section, we present some experimental results of automatic gloss recognition with (and without, for the sake of comparison) the facial cues extracted with our system. We emphasize that the results reported here were obtained by others [58], who implemented automatic gloss recognition using the RWTH automatic speech recognition system (RASR) introduced in [54]. The RASR system was originally designed to perform automatic speech transcription from vocal tract features, using state-of-the-art statistical language modeling techniques. However, the features fed to the RASR system in [58] consisted of various visual cues pertaining to the communication modalities that are important in sign language. The authors, who participated in the SignSpeak project with us [37], notably included the facial cues extracted with our system in some of their experiments, for the facial expression modality.

All experiments in [58] were conducted on a signer-specific subset of the RWTH-PHOENIX-Weather corpus[7]. This corpus subset includes video data for the anonymous signer #3 only, because this signer alone appears in more than 20% of the full RWTH-PHOENIX-Weather corpus.

[7] As mentioned above, the RWTH-PHOENIX-Weather corpus is a challenging sign language recognition dataset. By the end of the SignSpeak project, it was determined by the machine translation experts of the consortium that insightful gloss recognition results could only be obtained on this corpus with signer-specific gloss recognition models. Based

We refer to this corpus subset, which totals ~36 minutes of video at 25 FPS, as RWTH-PHOENIX-Signer03. RWTH-PHOENIX-Signer03 has a vocabulary size of 266 different glosses, among which 90 occur only once as running glosses (gloss singletons). Its training set is composed of ~31 minutes of video (46,638 frames at 25 FPS), 304 distinct sentences, and 3,309 running glosses (including the gloss singletons). Its test set is composed of ~5 minutes of video (6,751 frames at 25 FPS), 47 distinct sentences, and 487 running glosses. The measure used in [58] to evaluate automatic gloss recognition on RWTH-PHOENIX-Signer03 is the word error rate (WER, also often used in the evaluation of speech transcription systems), which is defined as

$$\text{WER} = \frac{\#\text{deletions} + \#\text{insertions} + \#\text{substitutions}}{\#\text{observations}}. \tag{2.25}$$

The WER is well-suited to evaluate the quality of automatically glossed sentences in continuous sign language, because it takes into account the number of gloss deletions, insertions, and substitutions that are necessary to match the ground truth glossed sentences. In the case of isolated gloss recognition systems, the WER can still be used as is, and it simply reduces to the single-word classification error rate.

Table 2.3 presents WER results from [58] for the automatic glossing of continuous sign language on RWTH-PHOENIX-Signer03. For the facial expression modality, the visual features were the facial cues automatically extracted with our method in Sect. 2.2.2. The hand gesture modality was embodied by hand cues for the right (dominant) and left (non dominant) hands, both separately extracted as 3D gradient-based descriptions of spatio-temporal volumes (HoG3D [73]) composed of $\pm$ 4 hand patch images cropped using the ground truth central points of the right and left hand palms, respectively. The body posture modality was embodied by body cues extracted as PCA-reduced spatio-temporal volumes of $\pm$ 2 video frames. The best gloss recognition result in Tab. 2.3, giving the lowest error rate (41.9% WER), comes from the combination of the dominant hand cues with our facial cues. It is also worth noting that, among the results obtained from a single modality in Tab. 2.3, gloss recognition with our facial cues alone (62.6% WER) is better than gloss recognition with either body cues alone (80.1% WER) or non dominant hand cues alone (63.9% WER). In intself, this result showcases how much valuable information is contained in the facial expressions when signing, and how useful our extracted facial cues are in automatic gloss recognition.

The best (lowest) WER values in Tab. 2.3 may still seem somewhat high to the reader, but they are actually good results considering the challenges posed by the RWTH-PHOENIX-Weather corpus and the relatively small size of the RWTH-PHOENIX-Signer03 corpus subset. For reference, on another sign language recognition dataset called the SIGNUM database [74], the RASR system gives a WER of 11.3% for a combination of similarly extracted dominant hand cues and body cues[8]. Indeed, automatic gloss recognition on SIGNUM is less challenging than on RWTH-PHOENIX-Weather (or RWTH-PHOENIX-Signer03), as SIGNUM is composed of videos recorded

on the good facial landmark point tracking results that we obtained on a multi-signer video subset (RWTH-PHOENIX-Nose), we are confident that evaluating gloss recognition in a signer-specific setup is still indicative of the usefulness of our sign language facial cue extraction system in more general conditions, i.e., in multi-signer gloss recognition setups.

[8] We do not report more results on SIGNUM in this thesis, because we did not extract facial cues for a significant amount of the SIGNUM videos with the method proposed in Sect. 2.2.2.

TABLE 2.3: Excerpt from a table in [58], which shows the gloss recognition results obtained with the RASR system [54] on RWTH-PHOENIX-Signer03, using sign language visual cues as features. The sign language modalities consisting of hand gestures, facial expressions (embodied by our facial cues), and body posture are considered independently, as well as in combination. Our facial cues alone lead to a better gloss recognition performance than the body cues alone, or the left (non dominant) hand cues alone. The best modality combination is the one with the right (dominant) hand cues and our facial cues.

Single modalities	WER
Body cues	80.1%
Left (non dominant) hand cues	63.9%
Facial cues (with facial landmark points)	62.6%
Right (dominant) hand cues	45.2%
Combined modalities	**WER**
Left hand cues + body cues	63.7%
Right hand cues + body cues	45.2%
Right hand cues + left hand cues	42.9%
Right hand cues + facial cues	41.9%

in a controlled laboratory environment, at 780×580 pixel resolution and 30 FPS, and with nearly 14,000 running glosses from a vocabulary set of 450 glosses.

Sign language translation

In the standard approach to automatic sign language recognition, the first step is to perform gloss recognition from visual features, extracted from a video depicting a sentence in the source sign language. Then, the second step is to perform machine translation from the recognized glosses, to produce the corresponding sentence in the target spoken language. Note that the cornerstone of this standard approach is the effective capture of the sentence meaning by a ground truth gloss annotation, used as ideal output for gloss recognition, and as ideal input for machine translation. This standard approach was the one chosen in the SignSpeak project [37], as it allowed to design and assess gloss recognition and machine translation techniques mostly independently from each other. Indeed, the working hypothesis in SignSpeak was that an optimal sign language recognition system chain could be obtained from the parallel and independent progress made on the separate tasks of gloss recognition and machine translation.

The tasks of gloss recognition and machine translation were particularly decoupled in SignSpeak, also because of the focus given to the RWTH-PHOENIX-Weather corpus, which poses specific challenges in both tasks and encourages one to base the evaluation of machine translation techniques on ground truth gloss annotations. Consequently, very little investigation was made in SignSpeak to measure the exact influence of automatic gloss recognition performance on machine translation quality, and even less so to evaluate how our extracted facial cues, which are designed to feed a gloss recognition system, may influence the results of machine translation. However, some machine translation results, which we report in Tab. 2.4, are indicative to some

TABLE 2.4: Machine translation results obtained (by others) in SignSpeak on the RWTH-PHOENIX-Weather corpus, from ground truth gloss annotations, as well as ground truth gloss annotations with different amounts of simulated, random errors. The last row of the table shows the TransER obtained on the basis of actual gloss recognition results with dominant hand cues only. Overall, and as expected, the translation quality diminishes (i.e., the TransER increases) when the gloss recognition quality diminishes (i.e., the WER increases).

WER	TransER
0.0% (ground truth)	53.1%
15.0% (simulation)	68.6%
30.8% (simulation)	71.0%
48.7% (simulation)	75.8%
54.4% (recognition)	75.1%

extent of how gloss recognition performed with our facial cues may influence the performance of machine translation. These results were obtained by other researchers who worked on the machine translation task in SignSpeak, through translation experiments on the RWTH-PHOENIX-Weather corpus with (1) ground truth gloss annotations (2) ground truth gloss annotations with random errors, and (3) automatically recognized glosses (without our automatically extracted facial cues). The translation edit rate (TransER [75]) used in Tab. 2.4 is a translation error metric similar to the WER metric used in gloss recognition (see Eq. 2.25). In addition to the number of necessary insertions, deletions, and substitutions of words, the TransER involves the number of necessary shifts of words or groups of words to match the correct sentence in the target spoken language. Lower values of the TransER indicate better translation results. For reference, state-of-the-art machine translation TransERs for free texts in spoken languages, from English to German, English to Chinese, and English to Arabic, are of 52.34%, 63.74%, and 65.97%, respectively (machine translation of transcribed TED talks in English [76]).

The last row of Tab. 2.4 contains the translation result obtained on the basis of automatic gloss recognition with dominant hand cues only. Interestingly, the TransER in this last row (75.1%) is lower than the TransER in the row just above (75.8%) corresponding to the most noisy simulated gloss recognition WER (48.7%), even though this one is lower than the automatic gloss recognition WER (54.4%). This suggests that TransERs obtained from noisy gloss annotations are an upper bound for TransERs that can be obtained with actual gloss recognition results. Therefore, considering the machine translation TransERs in Tab. 2.4 and the automatic gloss recognition WERs in Tab. 2.3, we speculate that gloss recognition performed using a combination of dominant hand cues and our automatically extracted facial cues would lead to a TransER improvement (i.e., decrease) of more than 2% over a translation based on gloss recognition using dominant hand cues only, which is quite a significant increase in the translation quality. This speculation is further supported by the translation results presented next, which were obtained with a framework integrating gloss recognition and machine translation using facial cues extracted with our system.

Integrated framework with visible phoneme recognition

One shortcoming of the standard approach to sign language recognition is that it strongly relies on the hypothesis that the intermediary gloss form of a sentence in sign language is always effectively capturing its precise meaning. However, effectively capturing the multimodal nature of sign language with a comprehensive, universal annotation system actually remains an open question in the sign language research community [52]. Indeed, in natural, continuous sign language, a single sign (i.e., hand gesture) often has several different meanings, depending on variant executions, other signs used in the sentence, and subtle grammatical and semantic markers present in the other visual modalities (i.e., the facial expressions and the body posture). Additionally, in sign languages used in countries that have a strong oral education tradition, signs are often accompanied by mouthing, i.e., silently pronouncing the surrounding spoken language words with the lips while singing with the hands. Gloss annotations are powerful, but may require excessive efforts from the annotators to perfectly describe a sentence in sign language, and often linguists will produce non-exhaustive gloss annotations that suit their needs to study specific linguistic patterns.

As a consequence, sign language gloss corpora often lack the context details that would help obtain translation results as accurate as those obtained with spoken language text corpora, for which the annotation system, i.e., writing, is very natural and effective to capture meaning. More specifically, it is common that sign language sentences are glossed using so-called ID-glosses [77]. An ID-gloss designates a gloss annotation that has been associated to a sign (i.e., a hand gesture) independently of its context of execution and the other visual modalities. For instance, most of the RWTH-PHOENIX-Weather corpus contains the time boundaries and labels for the ID-glosses only (solely based on the signing hands yet with some pronunciation variants), because this corpus was originally developed for the recognition of hand-based features. During the creation of this corpus, very little emphasis was given on transcribing information coming from the other visual modalities, e.g., the facial expressions, which means, in particular, that signs that are identical in their hand component but differ in their facial expression component often received the same label.

The core idea of the integrated approach to sign language recognition (originally proposed in [40], to which we contributed) consists in recognizing mouthing in addition to recognizing ID-glosses, in order to provide machine translation with multimodal information that better describes a communication in sign language than information coming from the recognition of ID-glosses only. Indeed, recognizing mouthing consists in mapping visual facial cues to a target representation that is very close to the sentence in the target spoken language, which is why adding mouthing recognition to (ID-)gloss recognition toward machine translation is seen as an integrated approach, as opposed to the standard approach of (ID-)gloss recognition followed by machine translation.

The mouthing of a word can be phonetically divided into a sequence of atomic components, which are called phonemes (e.g., the word “map” would be described by the sequence *m-a-p*). Visually, however, some phonemes are indistinguishable from each other, such as *p* and *b*, which differ only in the aspiration. Since mouthing recognition is here based on visual cues only, the target representation of a mouthed word rather consists of a sequence of visemes, i.e., visually distinguishable phonemes (the phonemes *p* and *b* correspond to the same viseme *P*). In this setup, mouthing recognition is therefore implemented as a viseme recognition system, which seemingly performs automatic lip reading. To prepare the training of this viseme recognition system on the

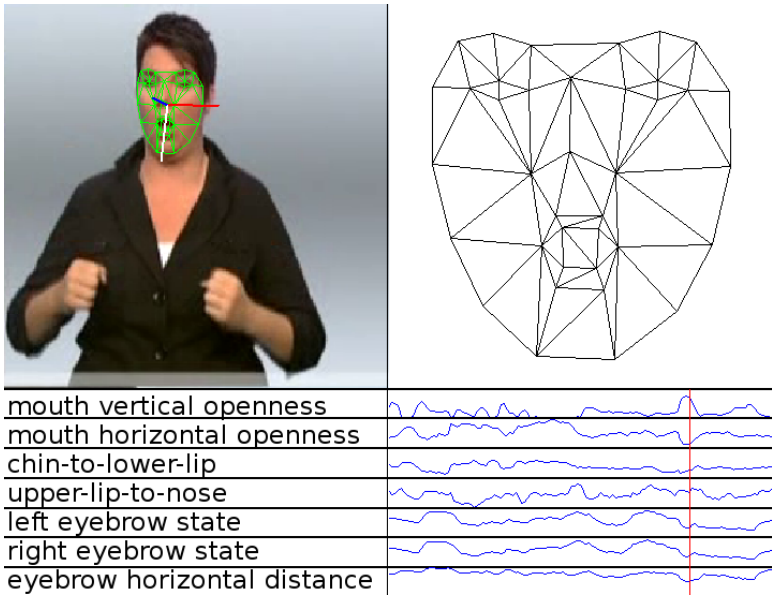


FIGURE 2.9: The facial cues we proposed for viseme recognition in [40]. These facial cues were obtained with a slightly different version of our sign language facial cue extraction system, based on the method proposed in Sect. 2.2.2. The bottom-right part of the figure shows the time evolution of the facial cue values throughout the processed video, with the vertical red bar indicating the values for the video frame shown in the top-left part of the figure.

RWTH-PHOENIX-Weather corpus, target viseme sequences were produced (not by us) on the basis of the target spoken language sentences in this corpus, and were associated to the ID-gloss labels[9].

We provided the visual cues used to train and test the viseme recognition system proposed in [40]. We used a slightly different version of our sign language facial cue extraction system, which version is however still based on the method we proposed in Sect. 2.2.2. We discarded the head pose facial cues, and we somewhat abused our concept of aperture facial cue, by defining new pairs of index subsets over to the AAM-aligned (Sect. 2.2.1) facial landmark points, notably for the calculation of new point set distance-based facial cues within the mouth region. Specifically, in addition to the mouth opening degree, our "aperture" facial cues here further include the distance between the mouth corners, the distance between the lower lip and the chin tip, and the distance between the upper lip and the nose tip. We also discarded the eye opening degrees, but kept the eyebrow raising degrees and included a new facial cue giving the distance between the eyebrows, as it was found empirically that these eyebrow-related facial cues helped in viseme recognition. Figure 2.9 gives an illustration of the facial cues we proposed for performing viseme recognition.

Table 2.5 shows the improvement obtained in machine translation (results from [40], using our automatically extracted facial cues described above), when using the viseme recognition-based integrated approach instead of the standard approach where machine translation solely uses ID-glosses. These results were obtained on a large subset of the RWTH-PHOENIX-Weather corpus

[9]Since not all signs are accompanied by mouthing, the ID-glosses where no lip movement could be observed were associated with a "silence" viseme representation, and the ones with lips movement irrelevant to mouthing with a "garbage" viseme representation.

TABLE 2.5: Translation results from [40], obtained on the RWTH-PHOENIX-Weather corpus, with the standard and integrated approaches to sign language recognition. The baseline (standard approach) machine translation results in the first row were obtained using the ground truth ID-glosses. In the second row, "oracle" TransER results are given, i.e., assuming that the viseme recognition was perfect (i.e., the target, ground truth viseme sequences were used). The third row shows the TransER results obtained using the actual output of the viseme recognition system, based on facial cues extracted with the method in Sect. 2.2.2.

Using visemes?	Viseme CER	ID-gloss WER	TransER
No	–	0.0% (ground truth)	66.5%
Yes	0.0% (ground truth)	0.0% (ground truth)	60.1%
Yes	32.2% (recognition)	0.0% (ground truth)	64.4%

originally annotated with ID-glosses only[10]. Tab 2.5 also includes the character error rate (CER) results obtained with the viseme recognition system that uses our facial cues. The CER is calculated in the same way as the WER (Eq. 2.25). It takes into account the number of character (viseme) deletions, insertions, and substitutions necessary to match the target viseme sequences. One can observe that, as speculated in the previous section, the inclusion of facial cues extracted with our system within the overall sign language recognition system chain gives a significant improvement in the translation quality, of about 2% TransER.

Note that the version of our facial cue extraction system proposed in [40] as part of an innovative viseme recognition system has also been used as is in follow-up work, again mostly as a basis for developing enhanced viseme recognition systems for integrated sign language recognition. This follow-up work, in which we did not directly participate, but where our facial cue extraction system was extensively used and acknowledged, includes [78, 79, 80] (as well as [81], to a lesser extent).

2.4 Conclusion

The idea of a cheap and noninvasive technology that effectively performs the automatic recognition of natural, continuous sign language arouses a lot of interest in the deaf and hard of hearing communities. It is also very appealing to hearing people who cannot or hardly can sign, but have recurrent interactions with deaf or hard of hearing people. Such a technology would not only provide unprecedented comfort and efficiency during the punctual exchanges between hearing and deaf people, but also, and more importantly, allow to create new institutional means of communication between hearing and deaf people, to help bridging the social gap that currently marginalizes these latter. The automatic recognition of sign language is however a complex scientific and technical problem. The multimodal nature of the communication in sign language, combined with the absence of a natural transcription mechanism of sign language, poses specific yet intertwined

[10]The difference between the baseline machine translation results in the first row of Tab. 2.5 and the baseline machine translation results in the first row of Tab. 2.4 comes from the fact that different subsets of the RWTH-PHOENIX-Weather corpus were used, as well as different machine translation techniques, at the respective times of publication of these results.

challenges in the fields of linguistic modeling, machine translation, and computer vision. The SignSpeak project, which is at the origin of most of the work presented in this chapter, was a attempt to tackle these challenges simultaneously and efficiently, through the close collaboration of research teams active in the above mentioned fields.

Eventually, the results obtained in SignSpeak shed light on the actual possibilities and remaining difficulties toward realizing this long-awaited automatic sign language recognition technology, laying the foundation for the development of a working end-to-end system prototype. The sign language facial cue extraction system we presented in this chapter is but one conceptual brick of this prototype, yet an important one. Indeed, our system offers a solution to effectively capture visual cues from the sign language communication modality consisting of the facial expressions, which are very present and informative in continuous sign language. Our system is robust against the difficulties typically encountered when tracking the face of a signer in a video, i.e., large off-image plane head rotations, extreme expressions, and occlusions of the face by the hands. Our system also implements the normalization of the facial cues with respect to the signer's identity, when this identity is known and given, and quickly adapts to new signers when the identity is unknown.

We have shown in the results section of this chapter how useful the facial cues extracted with our sytem are in automatic gloss recognition, as well as in machine translation. In particular, we showed their usefulness in an integrated approach to sign language recognition, via a viseme recognition system that is based on our facial cue extraction system and seemingly performs automatic lip reading to enrich the transcription of sign language. Another piece of work, which is not detailed in this chapter, but in which we also participated, is the design of a technique to enhance avatar animation of continuous sign language with facial expressions in general, and mouth patterns in particular. We refer the reader to [41] for more details about this technique. Note that sign language avatar animation is a problem closely related to sign language recognition, the two consisting of dual perspectives on the problem of effective sign language understanding and modeling. This is mostly why our sign language facial cue extraction system proved to be useful in the modeling part of this technique that generates sign language-specific facial expressions for an avatar.

Finally, it should be noted that deep learning with convolutional neural networks (CNNs) is the current trend for effectively implementing automatic gloss recognition, either to extract sign language communication cues from images, among which facial cues, or to directly infer the glosses from images in an end-to-end manner. In [81], a CNN was trained to infer mouth shape class probabilities from cropped face images. Incidentally, the cropped face images and the mouth shape class labels used to train the CNN in [81] were obtained on the basis of our AAM-based face alignment method and our point set distance-based facial cue extraction method, respectively. In [82], a CNN with recurrent units was proposed for use in continuous sign language recognition, which is able to distinguish between over a thousand glosses directly from a video of a signer performing a sentence. The authors of [82] also emphasized that gloss recognition with their approach gave better performance when full video frames were considered rather than cropped hand images only, as a CNN can effectively capture information from all sign language modalities at once.

Chapter 3

Computer vision system for objective visual pursuit assessment

Visual pursuit, i.e., the ability of a person to track a slowly moving stimulus, is a key clinical marker in most of the assessment scales for post-comatose states. In this chapter, we present our work on a computer vision-based system that helps clinicians make the bedside assessment of visual pursuit more accurate and less subject to experimenter bias. Our system, which uses cameras on a head-mounted device, is specifically designed to work with the moving handheld mirror stimulus, so as to follow the recommended and well-established clinical setup for visual pursuit assessment. During the clinical procedure, our system works alongside the clinician by tracking both the patient's pupil and the moving mirror. Our system then outputs a score indicative of the quality of visual pursuit. We give an evaluation of our system on healthy control subjects and actual post-comatose patients. This evaluation eventually shows the great potential of our system toward making visual pursuit assessment in a hospital environment an objective procedure. The content of this chapter is essentially based on two of our articles, one published in the proceedings of the 16th IEEE Winter Conference on Applications of Computer Vision [83], and the other one published in the Journal of Neurology [84]. All work about the computer vision-based system presented in this chapter is our own. The development of the head-mounted device, the clinical assessments, and the statistical evaluations were however made by others, and we make this explicit within the text wherever necessary.

3.1 Introduction

Disorders of consciousness (DOC) are neurological syndromes in which the patient's consciousness has been severly affected due to important brain damage. Different DOC states have been defined, including the state of coma, and the post-comatose states known as unresponsive wakefulness syndrome (UWS, previously called vegetative state) [85], and minimally conscious state (MCS) [86]. DOC states can be transitory, as patients in coma may evolve into UWS, then into MCS. Patients in coma show no sign of being awake, or of being aware of themselves or their surroundings. Patients in UWS are awake, in the sense that their eyes are open and that they may be capable of basic reflexes such as coughing and swallowing, but they do not show any sign of self or environmental awareness. Patients in MCS, however, are characterized by the presence of a number of reproducible, cognitively mediated behaviors, e.g., purposeful behaviors, which are distinguishable from reflex activity. A subcategorization has been proposed for MCS: MCS

plus (MCS+) and MCS minus (MCS-) [87]. Patients in MCS- are characterized by the fact that they only show lower-level non-reflex behaviors, such as pain localization or object localization. Patients in MCS+ are characterized by the specific presence of response to command. Also, when patients are able to functionally communicate and/or to functionally use objects, they are said to have emerged from MCS (EMCS) [86].

The appropriate clinical management and accurate diagnosis and prognosis of patients with DOC are difficult tasks that have engaged the efforts of medical doctors and researchers in neuroscience for many decades (going back to thehardware mid-1960s [88]). In some cases, the characterization of a DOC patient's state can be very challenging to make, especially the distinction between MCS and UWS, even for a skilled clinician. And yet, for ethical and medical reasons, it is of paramount importance to correctly recognize the signs of consciousness. For instance, it has been shown that patients in MCS are able to process auditory information, and to suffer from pain, unlike patients in UWS [89]. Furthermore, proper care of MCS patients can lead in some cases to full recovery of consciousness [90]. In the current clinical practice, bedside assessment of consciousness using behavioral scales is the gold standard to characterize the state of patients with DOC. Several scales have been designed, but the most popular and widely used one is the Coma Recovery Scale-Revised (CRS-R [91]). The CRS-R is among the few assesment scales showing strong evidence of reliability and validity for the assessment of DOC, based on a recent systematic review completed by the American Congress of Rehabilitation Medicine [92]. As compared to a diagnosis achieved solely by clinical consensus of the medical staff, the CRS-R allows to avoid 41% of misdiagnosis (i.e., erroneously considering patients in MCS as being in UWS) [93].

The CRS-R is divided into several assessment subscales (auditory function, motor function, visual function, etc.), and incorporates in its visual function subscale the assessment of visual pursuit, i.e., pursuit eye movement in direct response to a moving stimulus. Visual pursuit is a key response for the clinical assessment of patients with DOC. Indeed, it is one of the first signs appearing during recovery of consciousness, and it is a strong behavioral marker that, if present, is sufficient to diagnose MCS and discard UWS [86]. According to different studies assessing the evolution from UWS to MCS, around 45% of patients are diagnosed in MCS by establishing the presence of visual pursuit [94, 95]. Moreover, in the global population of MCS, the prevalence of visual pursuit is around 70% [96, 97]. In order to create the moving visual stimulus necessary to assess visual pursuit, the CRS-R Administration and Scoring Manual (available from the authors of the CRS-R by request) recommends to clinicians to move a handheld mirror in multiple trials right in front of the patient's face so that he/she might follow his/her own reflection. Following this recommendation, the use of this autoreferential stimulus was shown to be consistently reliable for declaring visual pursuit in MCS patients as compared to using other stimuli, e.g., a person or an object, with which some MCS patients showed no pursuit at all even though they were actually able to follow a moving mirror [97]. This result was also confirmed in [96], with additional insight being given on the influence of the mirror trajectories chosen during the clinical assessment procedure. Studies on healthy subjects also showed that using a mirror was more efficient than using an object, because it elicits a smoother visual pursuit. Finally, the use of a mirror was shown to decrease the probability of erroneously considering that the patient did not follow the stimulus [98].

While the research on visual pursuit assessment for patients with DOC has provided clinicians with precise and meaningful guidelines (notably with the use of the CRS-R), this assessment in practice only relies on subjective categorical estimates made by the clinician about the eye tracking ability of the patient. Indeed, the end result of visual pursuit assessment consists of a "follows" or "does not follow" statement, with no further details required, and based solely on the personal decision of the clinician doing the assessment. These estimates can obviously be biased and impact the overall diagnosis. For such a sensitive task, objective and quantitative measures are desirable as additional information that can be used by the clinician to refine the outcome of the assessment. Indeed, in general, medical sciences increasingly tend to incorporate objective measurement tools for supplementing the clinical assessment and decreasing the caveats of bedside assessment. For example in the assessment of DOC, electrophysiology and neuroimaging are used for improving the detection of command following [99, 100, 101], or of cerebral activity compatible with a residual consciousness [102, 103, 104]. Following this trend, the idea of using some form of eye tracking technology toward supplementing the assessment of visual pursuit is appealing.

Off-the-shelf eye tracking technologies cannot however be easily adapted to the requirements of bedside assessment of visual pursuit in DOC patients. As an illustration of this, in a first attempt to quantitatively measure visual pursuit in DOC patients, an off-the-shelf computerized eye tracking system was used, together with visual stimuli displayed on a computer monitor [105, 106]. In this preliminary work, visual pursuit was measured by on- and off-target fixation statistics obtained for the patient who was asked to follow the stimuli on the computer monitor. The use of such a system exhibits the following weaknesses:

1. It requires the patients to be seated, in order to face the monitor displaying the stimuli. This may not only introduce bias due to the lack of comfort, but also lead to the exclusion of some patients from the assessment, e.g., those who suffer from spasticity or from a lack of tonus.

2. The overall eye tracking system does not (and cannot, as it stands) conform to the recommended practice of using a mirror [97, 96]. This system uses suboptimal stimuli instead, e.g., a red dot or the richly colored image of a parrot.

3. Because eye tracking systems typically require a prior calibration stage, the authors of [105, 106] had to include an *a posteriori* correction stage of the results for each patient. Indeed, DOC patients are by definition much less collaborative and communicative than what would be required in a prior calibration stage of an off-the-shelf eye tracking system.

The work presented in this chapter is about the computer vision-based system we developed to assist clinicians in their bedside assessment of visual pursuit in patients with DOC [83, 84]. Specifically, using video data from two cameras on a head-mounted device, our system tracks the patient's pupil, as well as the mirror held by the clinician, then performs further processing on the obtained trajectories to give an objective and continuous measure of the visual pursuit ability of the patient. This system, which we designed in collaboration with DOC experts, works alongside the clinician, letting him/her perform the assessment procedure in the usual and recommended manner, i.e., by means of the moving mirror stimulus. Indeed, no changes are required in the

posture and behavior of either the clinician or the patient as compared to visual pursuit assessment performed without the presence of our system. We therefore contribute in two useful ways to the task of visual pursuit assessment for patients with DOC. First, our system helps clinicians enhance their subjective assessment of visual pursuit by providing an informative and fine-grained objective score relative to this assessment. Second, it does so while preserving the established clinical procedure that was validated as optimal and is widely used in clinical practice. To our knowledge and according to DOC experts, our system is the first to have both of these important characteristics. In the material and methods section of this chapter, we give a complete description of our system, notably the pupil and mirror tracking techniques used, and two different methods used for processing the obtained tracking trajectories toward extracting an objective score. In the results section of this chapter, we give an evaluation of our system on healthy control subjects as well as actual DOC patients. We also include side results that are indicative of the variability of bedside visual pursuit assessment. Specifically, these results show how reliable the bedside clinical assessment of visual pursuit is, as compared to a consensus by DOC experts on videos of the assessment [84].

To finish this introduction, we would like to put this chapter into the general perspective of the present thesis, which is about the automation of tasks of facial expression interpretation. One could wonder whether a system that measures the quality of visual pursuit is an instance of automating a face perception task, and in particular the interpretation of a facial expression. On the basis of our definition of facial expressions and their interpretations in Chap. 1, we argue that it is indeed. The eye is part of the face, and smoothly moving the pupil is a shared and temporary human behavior involving this specific face part, i.e., a facial expression which is observed by our system through pupil tracking. The complementary context information relevant to define an interpretation of this expression is here embodied by the expectation that there is a mirror smoothly moving right in front of the subject's face, and that the subject may follow this movement with his/her eyes. This context information is gathered on the fly by our system through mirror tracking. All in all, in the task of visual pursuit assessment, the facial behavior that is "a movement of the pupil" is particularly interpreted as "the ability to follow a moving mirror", or even "a marker of consciousness". This interpretation is made by the clinician during the procedure, and it is also made automatically by our system, working alongside the clinician.

3.2 Material and methods

3.2.1 System overview

Figure 3.1 depicts our vision-based system within a block diagram. Specifically, our system consists of all software modules within this diagram. The image acquisition part was provided by others and consists of a lightweight device that has to be fixed on the subject's head. We actually used two different prototypes for this head-mounted device, which we both describe in Sect. 3.2.2. In either one of the prototypes that we used, the head-mounted device has two cameras, which we refer to as the eye camera and the scene camera. The eye and scene cameras are connected to a laptop computer and provide video data to two distinct tracking modules, specific to one camera. One tracking module extracts the position of the pupil in the image reference frame of the eye

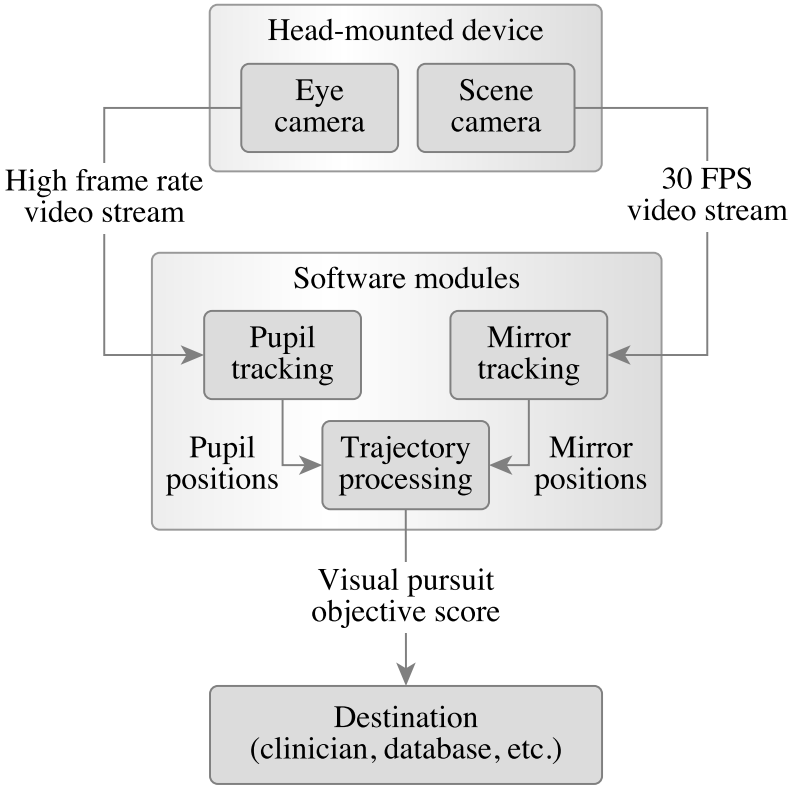


FIGURE 3.1: Overview of our computer vision system ("Software modules") within a block diagram.

camera (described in Sect. 3.2.3). The other tracking module extracts the position of the handheld mirror in the image reference frame of the scene camera (described in Sect. 3.2.4). A third module processes the synchronized pupil and mirror trajectories coming from the tracking modules, and outputs a continuous measure of the ability of the subject to follow the handheld mirror moved by the clinician. We actually propose two different methods for processing the pupil and mirror trajectories to output a measure of the visual pursuit ability. We describe both of these methods in Sect. 3.2.5. Finally, note that the video data acquired through visual pursuit assessment can be recorded on the laptop hard drive. Even though our system could, with a few adaptations, process video data at the time of the clinical assessment, we used this recording feature of our system to create experimental datasets and evaluate our methods in batch processing mode (Sect. 3.3).

3.2.2 Head-mounted device

When we started the project that eventually led to produce the content of this chapter, we used a cap-like prototype for the head-mounted device (Fig. 3.2), assembled at the Laboratory for Signal and Image Exploitation at the university of Liège. The scene camera of this prototype captures grayscale images of the scene as observed by the subject, at 752x480 pixel resolution and 30 frames per second (FPS). This scene camera uses an ultra wide angle fish eye lens with a horizontal field of view of 185° in order to cover the normal human field of vision and to ensure that the

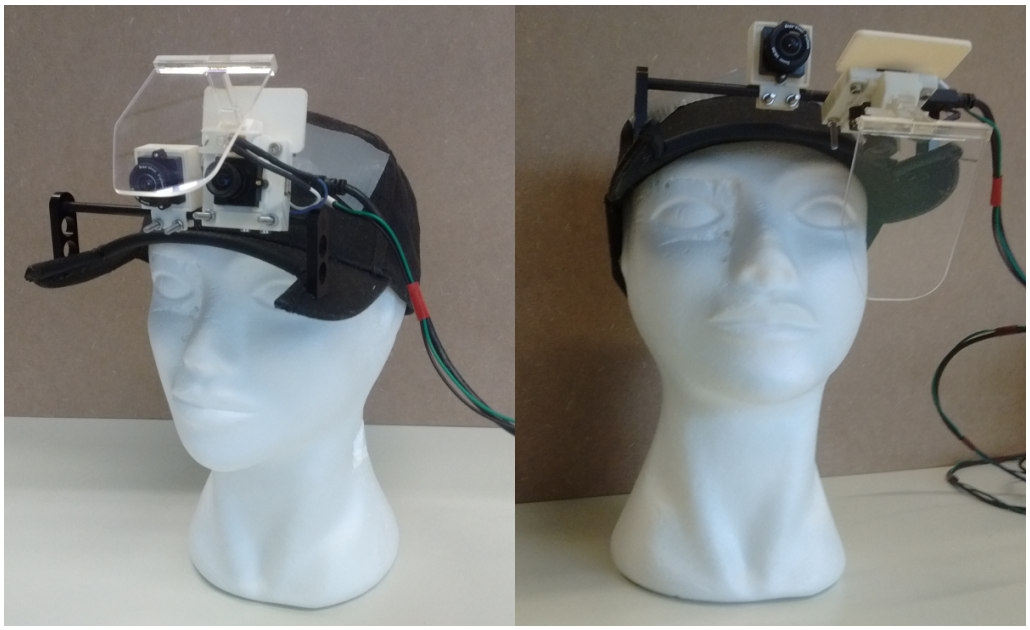


FIGURE 3.2: The first prototype we used for the head-mounted device. The beam-splitter can be raised in order to safely place the prototype on the patient's head (left). It is then lowered to enable the capture of close-up frontal images of the eye (right).

handheld mirror presented by the clinician to the subject is also visible by the scene camera. The eye camera of this prototype is sensitive to near infrared (IR) radiation and captures close-up grayscale frontal images of the eye of interest (left or right, easily selectable when needed). This eye camera is equipped with an IR illuminator and a sensor, and with a beam-splitter that is to be adjusted in front of the subject's eye. Obviously, the beam-splitter is reflective in the infrared and transparent in the visible to not disturb the subject's vision. Initially, the eye camera of this prototype was configured so as to capture images at 240x160 pixel resolution and 180 FPS. With this resolution and the adequate placement of the cap-like prototype and its beam-splitter, a pupil diameter of at least 20 pixels can be obtained within the image provided by the eye camera.

However useful and efficient in most cases, we found an intermittent issue in the cap-like prototype just described. With some agitated DOC patients lying in bed, this prototype could sometimes get displaced due to the patient rubbing the back of his/her head against the pillow. Such displacements could be large enough to remove the patient's pupil from the field of view of the eye camera, or even the patient's whole eye region in the worst cases. In such occurrences, we had to discard the recordings, readjust the cap-like prototype and its beam-splitter on the patient's head, and redo the assessment procedure all over again. After a few occurrences of the issue, we realized that it needed to be addressed. Our first solution was to increase the spatial resolution of the eye camera to 320x240 pixels, which allowed to get a larger field of view for the eye region, while still keeping the pupil diameter at around 20 pixels in the image. Set up in this way, the eye camera capture would not miss the pupil in case of a small displacement of the cap-like prototype. We also set a lower frame rate of 120 FPS for the eye camera, so that the acquisition chip could keep up with the higher eye image resolution. Care was taken in verifying that this decrease in

the eye camera frame rate would not cause difficulties in capturing pupil movements that could be part of a visual pursuit.

Even by increasing the pixel resolution of the eye camera, the head movements of a few DOC patients with a large or small skull structure could still lead the cap-like prototype to be displaced in problematic ways. Although this situation was fairly unfrequent, it led us to conclude that another, better-designed device would be advantageous for doing the rest of our experiments, as well as for demonstrating that a single adequate head-mounted device could be used with all DOC patients lying in bed. The head-mounted device we used from about the two thirds of our experimental data acquisition process onward was a glasses-like prototype (Fig. 3.3), adapted from a Drowsimeter R100 provided by Phasya S.A. (Angleur, Belgium). The scene camera of this prototype is exactly the same as the one used with the cap-like prototype (i.e, ultra wide angle fish eye lens, grayscale images, 752x480 pixel resolution, 30 FPS). The eye camera of this prototype is fairly similar to the one used with the cap-like prototype (i.e., close-up grayscale images, 320x240 pixel resolution, 120 FPS). One difference between the two prototypes is that the eye camera viewpoint of the glasses-like prototype is not exactly frontal, by design of the Drowsimeter R100. This difference is however mild enough to allow its systematic removal by applying a minor and consistent homography transformation to the pupil trajectory points obtained with the glasses-like prototype. Note that this simple post-processing step is peculiar to our change of prototype and our wish to homogenize the results presented in this chapter. It is not a method step that would be necessary to make our vision-based system work with new patients (cf. the calibration step typically required by common off-the-shelf eye tracking devices). Finally, note that, unlike with the cap-like prototype, the beam-splitter in the glasses-like prototype was fixed, increasing the standardization of the eye position within the captured images. Most of the remaining variations observed in the eye images of the glasses-like prototype are therefore simply due to morphological differences.

3.2.3 Pupil detection and tracking

Due to its many real-world applications, the automatic analysis of images of the human eye is a very prominent and thoroughly studied topic in computer vision. Various computer vision problems can be subsumed under the umbrella of automatic eye analysis, from iris recognition for biometric personal identification, to general-purpose eye gaze tracking in space, to name a few. As of today, some of these problems remain very challenging, especially when the envisioned application conditions are very general, e.g., eye tracking in the wild [107]. The eye analysis problem of interest in our application consists in detecting and tracking the center of the pupil in the image of one eye. This problem focuses on the most salient part of the eye image that is the dark and mostly homogeneous pupil region, and does not aim at inferring knowledge beyond the 2D realm of the eye image (unlike, e.g., the estimation of gaze direction in the 3D world). Moreover, the conditions defined for our specific application are quite well controlled, because of the quality of the eye camera of the head-mounted device. Indeed, by using either the cap-like prototype or the glasses-like prototype (Sect. 3.2.2), many of the difficulties that are usually encountered with less controlled image acquisition conditions could be alleviated in our application. The method

FIGURE 3.3: The second prototype we used for the head-mounted device, adapted from a Drowsimeter R100 provided by Phasya S.A. (Angleur, Belgium).

we propose for pupil detection and tracking is therefore fairly simple, because of the advantages offered by the acquisition system we used, which are as follows.

- The viewpoint of the eye region does not vary much, i.e., it is standardized, because the eye camera is head-mounted.
- The pupil appears approximately circular most of the time, i.e., at its peak salience, because the standardized viewpoint is frontal or near frontal.
- The pupil size appears relatively large, with a diameter of at least 20 pixels, because the standardized viewpoint gives close-up images of the eye.
- The pupil is one of the only few salient parts in the image, because the field of view of the eye camera is limited to the eye region.
- The pixel intensities of the eye parts have little inter-subject variation, in particular the intensities of the iris and pupil pixels, because the eye camera captures in the infrared.
- The pupil appears to move smoothly in most occurences, facilitating its tracking, because the frame rate of the eye camera is high.

Pupil detection

To detect the center of the pupil in an image I, we first remove the few specular reflections by inpainting the very bright areas in I. These bright areas are detected by adaptive thresholding, i.e., thresholding using the mean of each pixel's neighborhood, followed by morphological closing. Then, using a collection of synthetic, circular and concentric iris/pupil templates of varying iris

radii and pupil-to-iris ratios, we calculate a series of correlation images of these templates with I. If the largest value of a correlation image does not cross an empirical threshold value, this correlation image is discarded. If all correlation images are discarded, the pupil is assumed to be absent from I[1]. If there remains at least one correlation image, we define the coordinates of the detected pupil center $\mathbf{p}$ to be the x and y median values of the pixel positions corresponding to the largest values in the remaining correlation images. For further use during tracking, we also define the template T_0 as a circular region of I centered at $\mathbf{p}$ and with radius determined as follows. The radius r_0 of T_0 is set to be the median value of the iris radii in the synthetic templates corresponding to the remaining correlation images.

Pupil tracking

To track the center of the pupil in a video frame, we use a template matching approach that is based on the template update strategy proposed in [62]. This strategy was designed to bring a solution to the recurrent problem of object drift in template matching-based tracking methods. It is well-suited to robustly track near rigid objects with simple shape models – in our case, a 2D translation model – when such objects undergo minor changes in appearance throughout a video.

Let us assume that the pupil center $\mathbf{p}_{k-1}$ was successfully extracted from the video frame I_{k-1}. Let us also assume that we have two templates T_0 and T_k. T_0 is the original template obtained at time 0 (i.e., at detection time, see the previous section). T_k is an updated template estimated at time $k-1$ (i.e., at the most recent tracking time[2], see below). Using these two templates, we perform two sequential searches within the frame I_k to find the current pupil center $\mathbf{p}_k$. The first search is made by correlating T_k with a small region of I_k centered at the previous pupil center $\mathbf{p}_{k-1}$. The best match found by this first search gives us the provisional pupil center $\mathbf{p}_k$ for the frame I_k. The second search is then made by correlating T_0 with a small region of I_k centered at the (provisional) pupil center $\mathbf{p}_k$ found by the first search. The best match found by this second search gives us the candidate, drift-corrected pupil center $\mathbf{p}_k^*$ for the frame I_k. Drift correction is applied, i.e., $\mathbf{p}_k := \mathbf{p}_k^*$, if $||\mathbf{p}_k^* - \mathbf{p}_k|| \leq \epsilon$, where ϵ is a small empirical threshold. The template T_k is then updated to T_{k+1} according to the following rule: (1) if the drift correction was applied, then $T_{k+1} := I_k(\mathbf{p}_k, r_0)$, where the notation $I(\mathbf{p}, r_0)$ denotes the circular region of I centered at $\mathbf{p}$ with the radius r_0 found at detection time; (2) if the drift correction was not applied, then no update is made to the template, i.e., $T_{k+1} := T_k$. The updated template T_{k+1} will be used at the time of the first search in I_{k+1}.

Blinking or prolonged eye closure causes a temporary absence of the pupil in the image, which also causes our tracking method to fail. To detect such situations, we require that the best match of the first search, i.e., the search made within the image I_k using the updated template T_k, crosses a empirical threshold value. If it does not, the tracking procedure is stopped, and we perform pupil center detection in subsequent frames to try re-initiating the tracking procedure. Our method for pupil center detection and tracking performs well on our data, as can be seen in Fig. 3.4.

[1] As it is required to initialize pupil tracking, pupil detection that fails within a given video frame will be tried again as early as possible, i.e., within the next video frame.

[2] At time 1, i.e., just after detection, the convention is to choose $T_1 := T_0$.

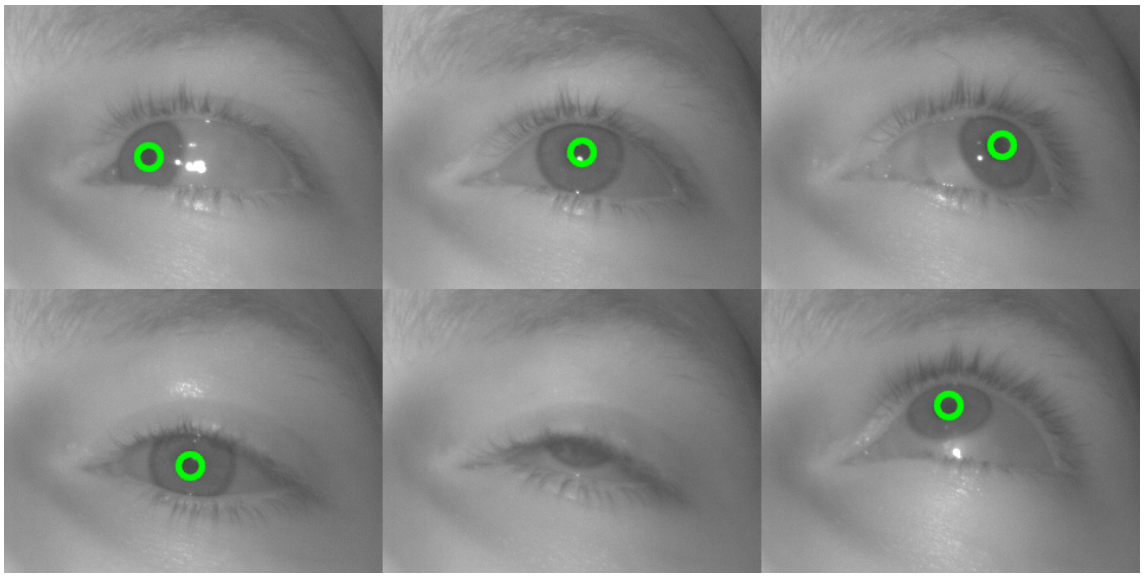


FIGURE 3.4: Snapshots of a video taken with the eye camera of the first head-mounted device (the cap-like prototype, described in Sect. 3.2.2), with superimposed results (as green circles) obtained with our method for pupil tracking.

3.2.4 Mirror tracking

In contrast with the problem of pupil tracking (Sect. 3.2.3), which is very frequently mentioned and addressed in the literature, the problem of tracking a handheld mirror facing the camera is not customary. Actually, we could not find any piece of literature or off-the-shelf technology dedicated to solving this specific problem. Tracking the non-reflective part of a mirror, i.e., its frame, can be recast as an instance of the general problem of tracking a 3D rigid object in a 2D image. This general problem can be solved by calling upon different standard approaches, e.g., fiducial-based tracking, model-based tracking, interest point-based methods, etc. [108]. However, it is a fact that a mirror is mostly a reflective surface, and that the reflected patterns present in the image indirectly contain information about the 3D pose of the mirror. Additionally, because a handheld mirror frame is typically thin and not prominent with respect to its reflective part, only focusing on tracking the mirror frame and ignoring the information coming from the mirror reflections is a suboptimal solution at best, and a hazard for the overall tracking robustness at worst.

Also note that, in our application, it is an absolute necessity to gather as accurate and robust information as possible about the mirror moved by the clinician, as observed by the patient, and that not much simplification can be made about the experimental environment. Indeed, to satisfactorily solve the handheld mirror tracking problem in our application, we have to assume difficult conditions inherent to a hospital environment, such as a cluttered and non-fixed scene with various moving objects, and significant variability in the spatial mirror pose. The only reasonably expectable conditions are that the mirror is present in the image, and that its movements appear smooth so as to comply with the visual pursuit assessment guidelines. In light of the challenges posed by this problem and the lack of applicable solutions found for it in the literature, we think that our solution, which is exposed next, may be innovative in itself, and not only in our application context. Of course, this problem of tracking a handheld mirror is somewhat specific and

seldom encountered in practice.

Our mirror tracking method is essentially 3D model-based, but it also incorporates key constraints relative to the mirror plane, which are estimated via a general geometric analysis of the patterns reflected by the mirror. For its model-based part, our method consists of a 3D shape model registration technique by 2D template matching, derived from the Lucas-Kanade algorithm [68]. The 3D shape model $\mathcal{S} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$ is a collection of N evenly and densely distributed 3D points that approximately represent the frontal surface of the mirror frame. Specifically, $\mathcal{S}$ models the non-reflective part of the mirror object as seen from the front, i.e., with the principal axis of the camera being normal to the reflective surface of the mirror. Such a model can be obtained either from a single depth image of the real mirror object in a frontal pose, or by hand-crafting, which is easy enough if the real mirror object's frame shape is pretty regular and its dimensions have been measured. The frontal pose of $\mathcal{S}$ is considered to be the reference pose, with rotation and translation denoted as $\{\mathbf{R}_0, \mathbf{t}_0\}$. We use the perspective camera model for projecting $\mathcal{S}$ onto the image plane of the camera. This camera model transforms a 3D point $\mathbf{X}_i = [X_i; Y_i; Z_i]$ into its 2D projection $\mathbf{x}_i = [x_i; y_i]$ in the image, via

$$\mathbf{x}_i = \mathbf{K}\,(\mathbf{R}\mathbf{X}_i \,+\, \mathbf{t}), \qquad \mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.1}$$

where $\mathbf{R}$ and $\mathbf{t}$ represent the rotation and translation in 3D, respectively, and $\mathbf{K}$ is the calibration matrix with intrinsic camera parameters f_x, f_y, c_x, and c_y. These parameters, all expressed in pixels, represent the focal lengths in the x and y axes, and the image coordinates of the center of camera, respectively. They were determined via a prior calibration procedure, as usually done. Also note that, in our application, the fish eye lens of the scene camera produces strong radial distortion, and the calibration procedure therefore also included the estimation of the distortion coefficients (not shown in Eq. 3.1). In practice, mirror tracking was performed on images where the distortion had been removed.

The template T used in our mirror tracking method is a 2D image of the mirror frame as seen from the front. Similarly to the 3D shape model $\mathcal{S}$, the template T can be obtained either from a single grayscale image of the real mirror object in a frontal pose, or by hand-crafting, which is easy enough if the real mirror object's frame appearance has simple textural patterns (e.g., all dark, or regularly arranged dark and light regions). The relationship between the template T and the shape model $\mathcal{S}$ is so that the spatial domain of T, denoted $\{\mathbf{u}_i\}$, corresponds to the projection of the $\{\mathbf{X}_i\}$ with the perspective model in Eq. 3.1, when $\mathcal{S}$ is in the reference pose $\{\mathbf{R}_0, \mathbf{t}_0\}$, i.e.,

$$\mathbf{u}_i = \mathbf{K}(\mathbf{R}_0\mathbf{X}_i + \mathbf{t}_0) \qquad \forall i \in \{1, \ldots, N\}. \tag{3.2}$$

For ease of explanation here, we present the template T as being constant (i.e., having fixed pixel values). In practice, the template is actually parameterized to allow global linear variation of its pixel values, in order to account for global camera gain and exposure bias. Using the 3D shape model $\mathcal{S}$, the perspective model in Eq. 3.1, and the 2D template T with its spatial domain as in Eq. 3.2, we can define the first term of the objective function that will have to be minimized in

order to retrieve the mirror pose from an image I, as

$$E(I, \mathbf{R}, \mathbf{t}) = \sum_{i=1}^{N} \rho(I(\mathbf{K}\ (\mathbf{R}\mathbf{X}_i\ +\ \mathbf{t})) - T(\mathbf{u}_i)), \tag{3.3}$$

where ρ is the Huber loss function [68], which is quadratic for small values of its argument, and linear for large values thereof. Given a pose $\{\mathbf{R}, \mathbf{t}\}$, the loss term in Eq. 3.3 is a robust M-estimator of the residuals between the image and the template. Indeed, it weights down the residuals that are likely to come from an occlusion, e.g., the hand of the clinician, which, as such, should not contribute too much to the minimization procedure.

To express the constraints about the reflective plane of the mirror, we use a method inspired by the work in [109], where it was shown how the pose of a camera can be estimated provided that the rigid motion between a number of virtual views induced by planar mirror reflections is known. In our application, the problem is simplified by making the assumption that the mirror pose and, therefore, its plane normal $\mathbf{n}_{k-1}$ and scalar Euclidean distance to the origin d_{k-1} are known *a priori* at frame I_{k-1}. We also make two other assumptions, namely, (1) that the scene camera is fixed in the world reference frame, and (2) that the 3D environment being reflected by the mirror is mostly static between two consecutive frames. The differences between the projected reflections in two consecutive frames I_{k-1} and I_k can be thought of as coming from the change of viewpoint of a virtual camera of center $\mathbf{C}^*_{k-1} = 2d_{k-1}\mathbf{n}_{k-1}$, which is symmetric to the real camera of center $\mathbf{C} = \mathbf{0}$ with respect to the moving mirror plane. To retrieve the virtual camera rotation $\mathbf{R}^*_k$ and translation $\mathbf{t}^*_k$, we use the essential matrix method [110]. This method exploits the known calibration matrix $\mathbf{K}$ and an estimated fundamental matrix $\mathbf{F}_k$ embodying the epipolar geometry that explains the correspondences, between the projected reflections in frames I_{k-1} and I_k, of the image keypoints. From the new virtual camera center, obtained by $\mathbf{C}^*_k = \mathbf{R}^*_k\mathbf{C}^*_{k-1} + \mathbf{t}^*_k$, the estimation of the new mirror plane normal is $\mathbf{n}_k = \mathbf{C}^*_k\ /\ ||\mathbf{C}^*_k||$, with Euclidean distance to the origin $d_k = -\langle \mathbf{n}_k, \mathbf{C}^*_k\ /\ 2\rangle$.

Incorporating a regularizing term that penalizes the distance of the 3D points of the shape model $\mathcal{S}$ to the reflective mirror plane $\{\mathbf{n}_k, d_k\}$ estimated at frame I_k, the complete minimization problem for finding the mirror pose $\{\mathbf{R}_k, \mathbf{t}_k\}$ at frame I_k is therefore

$$\underset{\mathbf{R},\mathbf{t}}{\operatorname{argmin}}\, E(I_k, \mathbf{R}, \mathbf{t}) + C\sum_{i=1}^{N}(\langle \mathbf{R}\mathbf{X}_i + \mathbf{t}, \mathbf{n}_k\rangle + d_k)^2, \tag{3.4}$$

where $E(I_k, \mathbf{R}, \mathbf{t})$ comes from Eq. 3.3, and C is an empirical constant multiplier balancing the soft constraint. After a coarse initialization of the mirror pose at the beginning of a scene video, the Gauss-Newton algorithm derived from the optimization problem in Eq. 3.4 continuously extracts the 3D pose of the mirror robustly, i.e., in the presence of extreme projective deformation, clutter, and occlusions. Figure 3.5 illustrates the effectiveness of our mirror tracking method. Note that, as we explain in the next section, in our application, we only need to track the 2D position of the mirror in the image of the scene camera. Tracking the full 3D mirror pose merely consists of a means to robustly obtain this image position of the mirror. Therefore, the mirror tracking module of our system outputs the image mirror position for each frame I_k, as $\mathbf{m}_k := \mathbf{K}\mathbf{t}_k$.

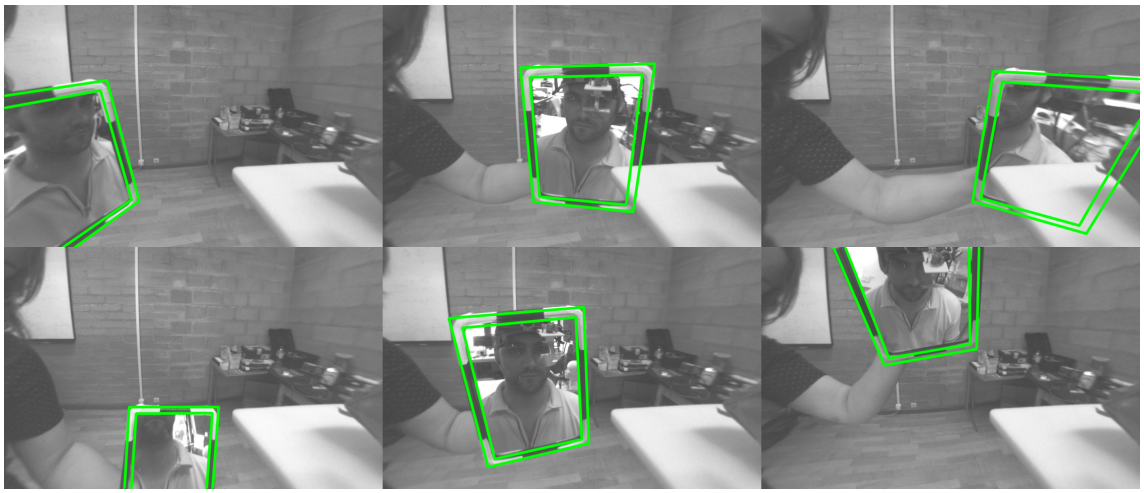


FIGURE 3.5: Snapshots of a video taken with the scene camera of the first head-mounted device (the cap-like prototype, described in Sect. 3.2.2), with superimposed results (as green lines) obtained with our method for mirror tracking.

3.2.5 Trajectory processing

Because the pupil and mirror tracking modules were designed to be robust, they do not output strong outliers. By using Kalman filtering [111], statistical noise and inaccuracies are efficiently removed from the extracted pupil and mirror positions. Also, in case of blinking or prolonged eye closure[3], we use the Kalman filter to smoothly predict the missing pupil positions due to the temporary loss of tracking. Likewise we predict the missing mirror positions due to the lower frame rate of the scene camera, as compared to the high frame rate of the eye camera. As a result, the pupil and mirror trajectories $\{\mathbf{p}_i\}$ and $\{\mathbf{m}_i\}$ consist of two same-size, ordered sets of synchronized image positions, which are faithful of the movement of the pupil and mirror as observed by the eye and scene cameras, respectively, during the course of the visual pursuit assessment procedure.

The viewpoints of the eye and scene cameras of the head-mounted device are approximately symmetrical with respect to the frontal plane of the subject's face. Therefore, in the presence of visual pursuit, the pupil trajectory $\{\mathbf{p}_i\}$ should be fairly similar to the mirror trajectory $\{\mathbf{m}_i\}$[4]. The trajectory processing module of our system has the task of measuring the similarity between these trajectories, so as to provide an objective score that is indicative of the presence or absence of visual pursuit. We developed two different methods that can each provide an objective score for visual pursuit assessment. The first one is based on a correlation analysis on time-matched segments from $\{\mathbf{p}_i\}$ and $\{\mathbf{m}_i\}$. The second one is base on machine learning, and involves the

[3] In the clinical assessment of visual pursuit, prolonged eye closure is a matter of about one second, at most. Indeed, if the clinician detects a longer period of time for an eye closure during the assessment, he/she will systematically stop the procedure and try awakening the patient, before redoing the assessment in case of successful awakening.

[4] Actually, in the presence of visual pursuit, the pupil trajectory in the image domain of the eye camera should be similar to the *reflected* mirror trajectory in the image domain of the scene camera, i.e., the mirror trajectory that is "flipped" with respect to the vertical axis of the scene image coordinate system. We therefore systematically flip the images coming from the scene camera before they are passed to the mirror tracking module, to remove the reflection component from the similarity.

independent classification of time-matched segments from $\{\mathbf{p}_i\}$ and $\{\mathbf{m}_i\}$. We developed the second method to overcome some of the limitations we found about the first method, as our results with both methods will show in Sect. 3.3. Before describing these methods further, we find it necessary to give the details of the clinical procedure of visual pursuit assessment, as recommended by the CRS-R protocol. Indeed, the detailed specification of this procedure is the ground for implementing our trajectory processing module, especially with the second, machine learning-based method.

Clinical procedure and trajectory segments of interest

The clinician must first hold a planar mirror at about 15 centimeters right in front of the subject's face, which is called the reference frontal pose of the mirror, and verbally encourage the subject to fixate the mirror. The exact mirror shape and dimensions are not specified by the CRS-R, but it should obviously be big enough for the subject to see a full reflection of his/her face when held at the prescribed distance, and small and light enough to be effectively manipulated by the clinician. The clinician must then move the mirror slowly from its reference frontal pose, one time back and forth in each of the leftward, rightward, upward, and downward directions, i.e., at 45 degrees to the right and left of the vertical midline of the subject's face, and 45 degrees above and below the horizontal midline of the subject's face. This must be done while keeping the mirror at a constant distance from the subject's face and ensuring that the subject might see and follow his/her own reflection. The exact order of these four movements is to be chosen by the clinician. In general practice, the clinician tries to make them as random as possible to avoid evaluation bias. This series of four movements is then repeated, again in a random order, so that a total of eight visual pursuit trials, two in each of the leftward, rightward, upward, and downward directions, are performed. The presence of visual pursuit is declared by the clinician if the subject can follow the mirror without loss of fixation on at least two complete trials, i.e., from 0 to 45 degrees, irrespective of the directions of the successful trials.

For each of the eight trials, it is only when the mirror is moved from 0 to 45 degrees, i.e., from the reference frontal pose to a shifted pose, that the clinician determines whether the trial is successful or not by observing the patient's eyes. The return move from 45 to 0 degrees, i.e., to the reference frontal pose of the mirror, only prepares the next trial and should not be part of the overall assessment of visual pursuit. Therefore, even though the tracking modules of our system extract the pupil and mirror positions without interruption throughout the procedure, only time-matched segments of the trajectories $\{\mathbf{p}_i\}$ and $\{\mathbf{m}_i\}$ should be considered to evaluate their similarity. These segments correspond to the eight mirror move parts going from 0 to 45 degrees, two in each of the four directions. We did not include in our system a way to automatically find the boundaries of the segments of interest in $\{\mathbf{p}_i\}$ and $\{\mathbf{m}_i\}$. These boundaries were manually set by visual inspection of the mirror videos, which is not very time consuming as it can be done in one pass over a video played at normal speed. However, a fully automatic system should ultimately include the automatic detection of such boundaries. A proper method for doing so could be dynamic programming for sequence segmentation [112], applied to the mirror trajectory points, with the segment homogeneity model being the constant speed of displacement over the x- and y-axes.

Correlation-based objective score

Because of the choice of viewpoints for the eye and scene cameras, the pupil and mirror trajectories are given in image reference frames that share the same orientation, but have different origins and scales in the horizontal and vertical directions. In case of smooth visual pursuit, the pupil and mirror trajectories should therefore be approximately equivalent, up to a linear transformation. In statistics, the Pearson correlation coefficient [113] is a method of choice to measure the linear correlation between two variables. The calculation of our first objective score of the visual pursuit ability is thus based on the sample Pearson correlation coefficient, and essentially quantifies the linearity of the relationship between the pupil and mirror trajectory points.

We found empirically that it gives better results to quantify the linear relationship between the trajectories $\{\mathbf{p}_i\}$ and $\{\mathbf{m}_i\}$ separately in each of the horizontal and vertical directions, and then to average the results. We also ignore the negative correlations obtained for either one of the x- and y-coordinate samples, by thresholding them to zero before averaging (the Pearson correlation coefficient has values in $[-1, 1]$). We denote by $\widehat{p}_x$ and $\widehat{p}_y$ the samples of the x and y image positions of the pupil, and by $\widehat{m}_x$ and $\widehat{m}_y$ the samples of the x and y image positions of the mirror. These pupil and mirror samples are formed by concatenating the trajectory points from the time-matched segments of interest in $\{\mathbf{p}_i\}$ and $\{\mathbf{m}_i\}$, respectively. Denoting by r_{ab} the sample Pearson correlation coefficient between two samples a and b, the correlation-based objective score of the visual pursuit ability, which we coined the "confidence score", or C-score for short, is calculated as

$$\frac{\max(r_{\widehat{p}_x\widehat{m}_x},\ 0) + \max(r_{\widehat{p}_y\widehat{m}_y},\ 0)}{2}. \tag{3.5}$$

The values of the C-score in Eq. 3.5 are in $[0, 1]$, with the value 0 representing perfect confidence in the absence of visual pursuit, and the value 1 representing perfect confidence in its presence. A precise discrimination threshold for declaring the presence of visual pursuit could be obtained by a receiver operating characteristic (ROC) curve analysis of this score on experimental data. We found however that arbitrarily setting the discrimination threshold to 0.25 (somehow representing the proportion of visual pursuit needed to reach the CRS-R criteria) works well enough in practice. As an illustration, Fig. 3.6 shows the evolution of the confidence score (C-score) as the pupil and mirror trajectories are extracted over a visual pursuit assessment with a successful outcome, as declared by the clinician.

Machine learning-based objective score

As it will be shown in the results section of this chapter, some difficult cases with DOC patients led us to conclude that a linear correlation model is not sufficient to automatically assess visual pursuit reliably in all cases. In the most difficult practical cases, the interpretation made by the clinician has to be subtle and to account for faint purposeful movements of the patient's eyes, for each of the eight trials of the procedure. Figure 3.7 illustrates the variability of the trajectory patterns in trials that are considered by DOC experts to be successful. We take a supervised machine learning approach to design our second method for the extraction of an objective score of the visual pursuit ability. In doing so, we wish to capture from expertly labeled data the full complexity of visual pursuit assessment, as performed by a skilled clinician.

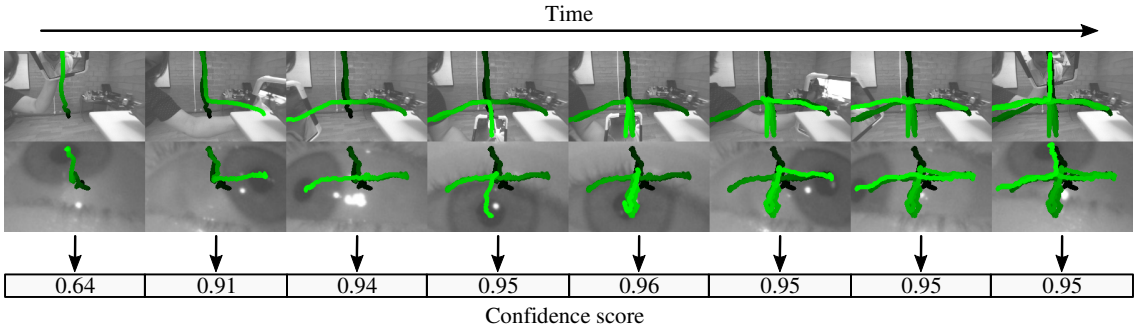


FIGURE 3.6: Time-lapse image sequence of a visual pursuit assessment with a successful outcome, illustrating the production of the pupil and mirror trajectories, along with the derived confidence score. The trajectories are drawn with various shades of green to visualize the progress in time (the brighter the green is, the more recently the trajectory point was extracted). The pupil and mirror trajectories look similar because of the presence of visual pursuit. This is corroborated by the evolution of the confidence score, which quickly reaches a value close to 1.

Let $\mathcal{C} = \{success, failure\}$ be the classes representing the possible outcomes of any one trial of a visual pursuit assessment, i.e., of a mirror movement in either one of the rightward, leftward, upward, or downward directions. Let $\mathcal{X} = \{\{\mathbf{p}_j, \mathbf{m}_j\} \,|\, j \in \{1, \ldots, N\}\}$ be the set of all possible time-matched trajectory segments of interest that can be obtained with the pupil and mirror tracking modules of our system. It is assumed that every such segment corresponds to one trial of a visual pursuit assessment, and that it has been resampled in time so as to contain exactly N couples of a pupil position $\mathbf{p}_j$ and a mirror position $\mathbf{m}_j$. Given a machine learning method and a training set $\mathcal{D} \subset \mathcal{X} \times \mathcal{C}$, we can learn a classifier $\hat{c} : \mathcal{X} \to \mathcal{C}$, i.e., a classification rule that can notably be used to predict the success or failure of each of the eight trials of a visual pursuit assessment. Given an effective machine learning method, the eventual performance of the classifier, i.e., the quality of its predictions, will largely depend on the quality of the training set $\mathcal{D}$ and the appropriateness of the functional form assumed for $\hat{c}(\cdot)$ within the learning method. We leave the details of the training set to the results section, and explain here the machine learning method we used, as well as the functional form we chose for $\hat{c}(\cdot)$ within this machine learning method.

We chose an artificial neural network approach (ANN) to learn our trial-level visual pursuit classifier $\hat{c}(\cdot)$. Using the ANN framework introduced in [114], we designed a sequential neural network architecture composed of four fully connected linear layers interspersed by three rectified linear units layers (ReLU) [115], and a log-softmax layer at its end. By incorporating the transfer functions that are the ReLU and the log-softmax, we wish to capture the hypothesized nonlinear nature of the subtle decisions made by the clinician at the trial level in difficult patient cases. The log-softmax layer was also chosen so that we can append a negative log likelihood (NLL) criterion to the end of the network to prepare the learning stage. The input and output sizes of the first three linear layers is $4N$. This means in particular that the input size of the network as a whole is $4N$. This size corresponds to the total number of xy coordinates in the $\{\mathbf{p}_j \,|\, j \in \{1, \ldots, N\}\}$ and $\{\mathbf{m}_j \,|\, j \in \{1, \ldots, N\}\}$ points within a time-matched trajectory input segment. The input and output sizes of the last linear layer are $4N$ and 2, respectively, to make the classifier learn over a

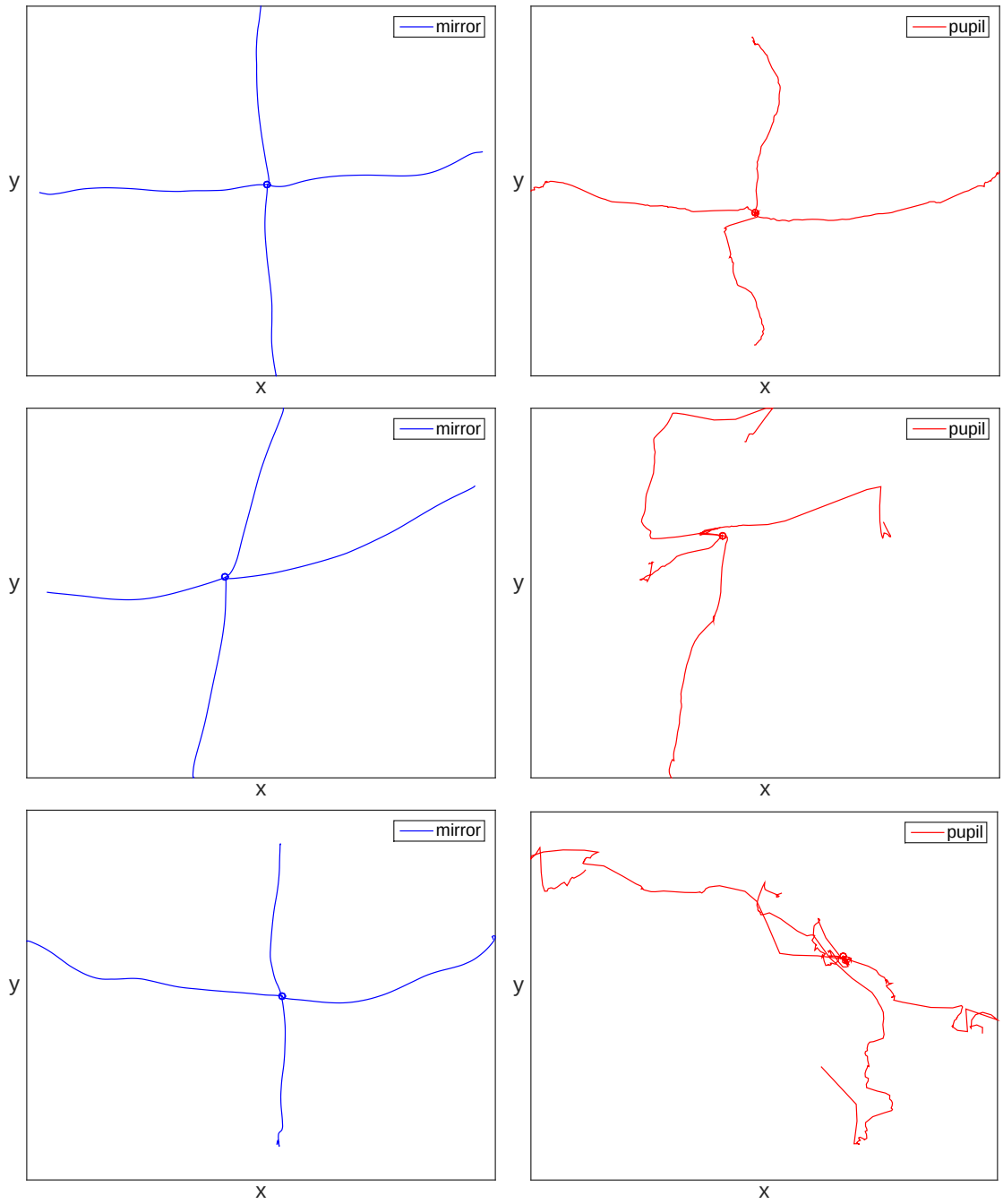


FIGURE 3.7: Mirror (left) and pupil (right) trajectories extracted by the respective tracking modules of our system, during the visual pursuit assessment of three different DOC patients at bedside. For each of the three assessments, only the first four consecutive trials are depicted. The left column shows the mirror movements in the leftward, rightward, upward, and downward directions. The right column shows the corresponding pupil movements, which were all labeled as successful by DOC experts, i.e., they follow the mirror movements. The first row illustrates a simple case, where the mirror and pupil trajectories look very similar. The second row illustrates a more difficult case, where the pupil trajectory is less regular and more dissimilar to the mirror trajectory. The third row illustrates a difficult case, where the pupil trajectory seems erratic. A linear similarity model may fail to recognize the presence of visual pursuit in such a difficult case.

2-class output space (through the log-softmax transfer function and the NLL criterion). With the neural network architecture (and the criterion) thus designed, the parameters of the neural network can be optimized by feeding it the inputs and targets from the training set $\mathcal{D}$. The number of network parameters to optimize is a function of the resampling size N of the trajectory segments. In this work, we use a resampling size of 100 trajectory points, leading to 482,002 network parameters to optimize. We use the classic stochastic gradient descent optimization procedure to learn our trial-level visual pursuit classifier $\hat{c}(\cdot)$ based on this artificial neural network architecture.

Once the trial-level classifier $\hat{c}(\cdot)$ has been learned, it is simple to use it to derive a global objective score for any new and complete visual pursuit assessment, i.e., a global score for a series of eight successive trials not in the training set $\mathcal{D}$. Indeed, provided that the eight time-matched pupil and mirror trajectory segments of interest $\{\{\mathbf{p}_j, \mathbf{m}_j\}_i \,|\, i \in \{1, \ldots, 8\}\}$ of a particular assessment are given, their outcome can be predicted individually by the trial-level classifier $\hat{c}(\{\mathbf{p}_j, \mathbf{m}_j\}_i)$, so as to produce a series of eight class labels $\{\hat{c}_i \,|\, i \in \{1, \ldots, 8\}\}$. By matching the class $success$ with the numeric value 1 and the class $failure$ with the numeric value 0, a scalar objective score in $\{0, 0.125, 0.25, 0.5, 0.625, 0.75, 0.875, 1\}$ can be calculated by averaging the elements of $\{\hat{c}_i\}$. To follow the recommendations of the CRS-R, if this machine learning-based score of the visual pursuit ability, or M-score for short, reaches the value 0.25 (i.e., two successful trials), then the presence of visual pursuit is declared by our system. We believe that the M-score may be more faithful of the clinician's cognitive process than the C-score, notably because the calculation of the M-score involves summarizing trial-level decisions by means of averaging, in the same way as the clinician does.

3.3 Experimental evaluation

3.3.1 Subject enrollment

To conduct the experimental evaluation of our system, healthy control subjects and chronic DOC patients were enrolled (by others, notably the clinician who did the visual pursuit assessments presented in this results section). For information, age and gender were not taken into consideration to enroll either the healthy control subjects or DOC patients.

The cohort of healthy control subjects consisted of 23 volunteers, between the age of 23 and 48 years, 13 of which were male. All of them provided written informed consent, and none of them was ever diagnosed with visual function impairment or neurological or psychiatric disorder.

The cohort of DOC patients consisted of 31 people, 12 in UWS, 11 in MCS-, 3 in MCS+, 5 in EMCS, 13 traumatic, 20 male, mean age: 40.23 $\pm$ 13.19 years, mean time since onset: 4.55 $\pm$ 4.84 years. The individual demographic and clinical data are included in Tab. 3.1. These patients were recruited during a one-week hospitalization at the university hospital of Liège. They were sent there by their treating physician and/or their family to be subjected to several clinical examinations. Written informed consent was obtained from each of the DOC patients' surrogate decision makers, in accordance with the research protocol approved by the university hospital of Liège. The criteria for the inclusion of a patient in our experiments were (1) to be at least 18 years old, and (2) to have suffered a severe brain injury leading to a prolonged DOC syndrome, as diagnosed by the CRS-R. The exclusion criteria were (1) a time shorter than three months since

the occurrence of the brain injury, and (2) the presence of a premorbid neurological or psychiatric disorder. The patients were not included on the basis of the integrity of their visual function as, most of the time, clinicians do not know *a priori* the visual function state of a patient before assessing visual pursuit.

3.3.2 Clinical assessments and data acquisition

The tests of our system with the healthy control subjects were all conducted in the same laboratory environment at the Montefiore institute of the university of Liège. The tests of our system with the DOC patients were all conducted in their respective rooms at the university hospital of Liège. The healthy control subjects were seated casually, and the DOC patients could either sit in a chair or lie in bed in their favorite, most comfortable position. Before proceeding to the visual pursuit assessment with a healthy control subject or a DOC patient, the head-mounted device was placed on the subject, and the laptop computer connected to the eye and scene cameras of the head-mounted device was placed on a table nearby, out of view of the subject being tested. These preparatory steps are very simple, and could be performed within a few seconds in all tests. We participated in these preparatory steps for most tests of our system with the DOC patients and the healthy control subjects.

After the preparatory steps were made, visual pursuit was assessed the clinician, who was the same experienced research neuropsychologist in all tests. Also, the handheld mirror moved by the clinician in all tests was the one shown in Fig. 3.5. The clinician followed the same procedure with either a healthy control subject or a DOC patient, i.e., she applied the procedure described in Sect. 3.2.5, in accordance with the CRS-R administration guidelines. Specifically, visual pursuit was considered by the clinician to be present when a smooth pursuit eye movement was observed on two occasions, in any of the leftward, rightward, upward, or downward directions, out of the eight trials prescribed by the CRS-R procedure[5]. For each test of our system, while the clinician applied the visual pursuit assessment procedure, eye and scene videos were recorded. These videos were later processed offline with the pupil and mirror tracking modules of our system, as well as the trajectory processing module, so as to output an objective score indicative of the presence of visual pursuit in each of the tested subjects.

In order to test the sensitivity of our system to the eye behavior, we conducted additional tests with the healthy control subjects. We used a procedure similar to the one described in the CRS-R for visual pursuit assessment, but with fundamentally different instructions given to the subjects. For 17 of the healthy control subjects, we organized another test where they were verbally encouraged by the clinician to focus their gaze on a fixed point and *not* to try following the moving mirror. Also, for 10 of the healthy control subjects, we organized another test where they were verbally encouraged to perform random eye movements and to *ignore* as best as possible the moving mirror. The general experimental setup remained unchanged for these additional tests, notably the rest of the CRS-R procedure for visual pursuit assessment was applied, the same research neuropsychologist performed all of the assessments, and the same handheld mirror was used. Also

[5]Incidentally, after a test of our system for the visual pursuit assessment of a DOC patient, the other items of the CRS-R were almost always assessed by the clinician, with our system removed, in order to make the complete clinical diagnosis for the patient.

TABLE 3.1: Demographic and clinical data of the cohort of DOC patients, and their visual pursuit (VP) assessment outcomes by human experts and by our automatic system. Gdr: patient gender; Age: patient age; TSO: time since onset of the DOC syndrome; Diag: DOC diagnosis; Clin-VP: VP assessment outcome by the clinician at bedside; Off-VP: VP assessment outcome by the clinician on video; Cons-VP (GS): VP assessment outcome by a consensus by DOC experts on video (**gold standard**); C-sc-VP: automatic VP assessment outcome with our system and the C-score; M-sc-VP: automatic VP assessment outcome with our system and the M-score. Equivalence with the **gold standard** is shown in green. Difference with the **gold standard** is shown in red. Only M-sc-VP is perfectly equivalent to the gold standard.

					Human expert decision			Automatic decision	
	Gdr	Age	TSO	Diag	Clin-VP	Off-VP	Cons-VP (GS)	C-sc-VP	M-sc-VP
1	F	28	4m	UWS	No	No	**No**	No	No
2	M	41	1y	UWS	No	No	**No**	No	No
3	M	73	9y	UWS	No	No	**No**	Yes	No
4	M	58	2y	UWS	No	No	**No**	No	No
5	F	50	6m	UWS	No	No	**No**	No	No
6	M	37	11m	UWS	No	No	**No**	Yes	No
7	F	41	2y	UWS	No	No	**No**	Yes	No
8	F	36	1y	UWS	No	No	**No**	No	No
9	M	33	14.5y	UWS	No	No	**No**	No	No
10	M	22	1y	UWS	No	No	**No**	Yes	No
11	M	23	1y	UWS	No	No	**No**	Yes	No
12	F	42	9m	UWS	No	Yes	**No**	No	No
13	F	40	3.5y	MCS-	Yes	Yes	**Yes**	Yes	Yes
14	M	45	13y	MCS-	Yes	Yes	**Yes**	Yes	Yes
15	M	54	6m	MCS-	No	No	**No**	No	No
16	F	25	11m	MCS-	Yes	Yes	**No**	Yes	No
17	M	26	12y	MCS-	Yes	No	**No**	Yes	No
18	M	25	1.5y	MCS-	Yes	Yes	**Yes**	Yes	Yes
19	M	34	12y	MCS-	Yes	Yes	**Yes**	Yes	Yes
20	M	34	12y	MCS-	No	No	**No**	No	No
21	F	62	2y	MCS-	Yes	Yes	**Yes**	Yes	Yes
22	F	42	7m	MCS-	Yes	Yes	**Yes**	Yes	Yes
23	M	55	7y	MCS-	Yes	Yes	**No**	Yes	No
24	F	31	13y	MCS+	Yes	Yes	**Yes**	Yes	Yes
25	M	60	2y	MCS+	No	No	**No**	No	No
26	M	30	4y	MCS+	No	No	**No**	No	No
27	M	24	5m	EMCS	Yes	Yes	**Yes**	Yes	Yes
28	M	49	1y	EMCS	Yes	Yes	**Yes**	Yes	Yes
29	M	33	11y	EMCS	Yes	Yes	**Yes**	Yes	Yes
30	F	37	6y	EMCS	Yes	Yes	**Yes**	Yes	Yes
31	M	58	4y	EMCS	Yes	Yes	**Yes**	Yes	Yes

note that the age and gender of the healthy control subjects were not taken into consideration to conduct these additional tests.

Each one of the visual pursuit assessment tests of our system conducted with either a healthy control subject or a DOC patient was performed exactly once, as well as the additional tests. Our experimental dataset is composed of 81 tests in total: 23 healthy control subjects who were encouraged to follow the moving mirror, 17 healthy control subjects who were encouraged to keep a fixed gaze, 10 healthy control subjects who were encouraged to do random eye movements, and 31 DOC patients who were encouraged to follow the moving mirror.

3.3.3 Assessment outcomes and gold standard

Since the motivation for our work is to provide an objective score of the visual pursuit ability to help clinicians enhance their bedside assessment of visual pursuit, it means in particular that the clinical decisions made during the construction of our experimental dataset may be subjective and biased. As such, these clinical decisions should not be considered as the perfect ground truth, or gold standard, to evaluate our system on the experimental video data. Providing accurate and non-biased ground truth about the visual pursuit assessment of a DOC patient is however a hard task. Assessing the simultaneity and similarity of the mirror and pupil movements altogether is a perception task for which the clinician doing the assessment is ideally located with respect to the patient. Having more DOC experts being present as observers during the assessment is not a suitable solution, because (1) their point of view is probably not ideal, if only because they do not move the handheld mirror themselves, (2) their mere presence may add even more bias to the assessment procedure, disturbing the patient and/or the clinician who moves the mirror, and (3) such a solution requires more research/medical staff for each visual pursuit assessment.

We propose to use as a gold standard the ground truth decisions made via the offline inspection by DOC experts of the video data acquired by our system. This solution has several advantages. Assuming that we give an appropriate presentation of the eye and scene video data, the DOC experts have access to accurate visual information about the degree of simultaneity and similarity between the pupil and mirror movements. Also, since the video data are to be visually inspected offline, the integrity of the clinical procedure at the time of visual pursuit assessment is preserved, and as many viewings as needed of a single assessment can be made by a DOC expert for him/her to make a decision. Finally, the overall decision-making process for producing the ground truth of a visual pursuit assessment can be organized by consensus on the basis of the individual decisions made by each of the DOC experts.

To implement the production of our video-based, consensus-based gold standard for our experimental dataset, we created an anonymous video dataset including all of the visual pursuit tests performed with the cohorts of DOC patients and healthy control subjects. We composed each video of this anonymous dataset so as to display the sequence of eye movements as recorded by the eye camera, side-by-side with a synthetic depiction of the corresponding, synchronized sequence of eight mirror movements as seen by the subject. We designed this synthetic depiction to ensure that the unnecessary details captured by the scene camera (such as the clinician's body movements and the subject's face in the mirror reflection) could not perturb the visual inspection or reveal the subject group or identity. We used the 3D mirror pose tracking feature of our

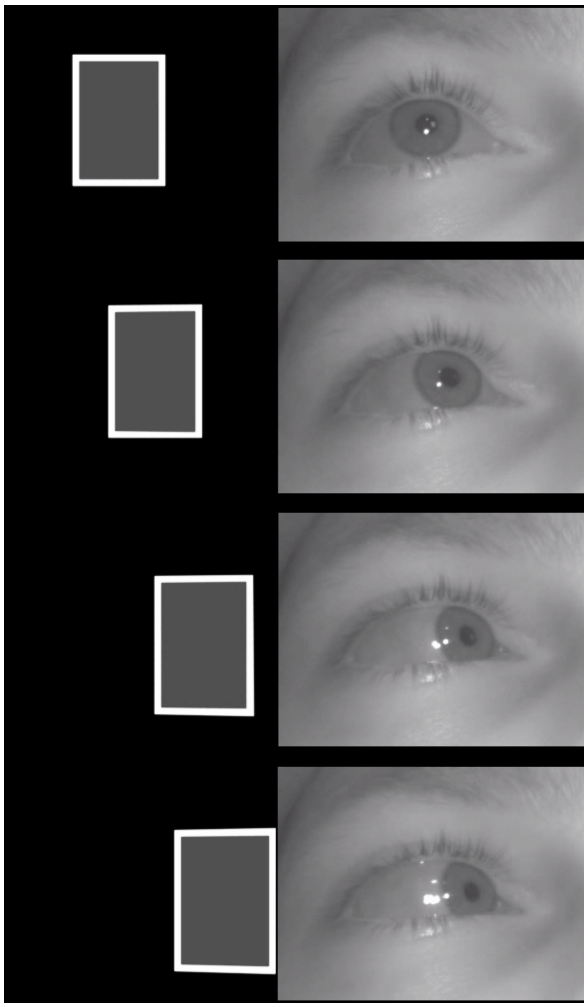


FIGURE 3.8: Time-lapse image sequence from a video of the anonymous dataset we created to produce gold standard decisions via a consensus by DOC experts. The synthetic depiction of the mirror movements is shown on the left side, and the corresponding eye images acquired by the eye camera are shown on the right side.

mirror tracking module (Sect. 3.2.4) to create this synthetic depiction of the mirror movements. Figure 3.8 shows a few images of a video of this anonymous dataset.

The videos of the anonymous dataset were presented for individual and independent visual inspection to three researchers with great experience in the clinical assessment of patients with DOC. These three DOC experts were the neuropsychologist who did the original clinical assessment, another neuropsychologist, and a neurologist. For each video, they could each score the eight trial movements prescribed by the CRS-R, labeling them as successful or not. After their individual scoring, the trials for which a unanimous decision was not obtained were discussed between the three DOC experts until a final decision was reached by consensus. In the healthy control subject tests, five trials out of the 400 performed had to be discussed to reach a consensus (1.25%). In the DOC patient tests, 28 trials out of 248 had to be discussed (11%). For the other trials, the three DOC experts had indicated the same appreciation without concerting. The gold standard for the presence of visual pursuit in each test was determined according to the CRS-R criteria, i.e., two or more successful trials out of the eight performed trials, as decided by the consensus. Also, the

gold standard objective score was set to be the proportion of successful trials, as decided by the consensus. For instance, if the test subject followed the mirror two times out of the eight trials according to the consensus, then the gold standard score was set to 0.25 ($= 2/8$), i.e., the minimum objective score to declare the presence of visual pursuit. Table 3.1 includes the gold standard, consensus-based visual pursuit assessment decisions (Cons-VP (GS)) made for the DOC patients in our experimental dataset. This table also includes the other human expert decisions made for the patients, i.e., the clinical decisions made at bedside (Clin-VP), and the individual decisions made by the clinician who performed the original assessment, after her visual inspection of the anonymous dataset (Off-VP).

Note that we also used the consensus-based trial-level decisions to assign class labels to the time-matched trajectory segments extracted from the video data of our experimental dataset, and so create the training set $\mathcal{D}$ needed to learn the trial-level classifier used in the derivation of the M-score (Sect. 3.2.5). We actually created as many training sets and performed as many training stages as the number of tested subjects (healthy control subjects as well as DOC patients), so as to perform a leave-one-subject-out cross-validation. Using some data augmentation heuristics, the size of each of the training sets could count several thousands of labeled trajectory segments. Each learned trial-level classifier was then used on the eight trajectory segments associated with the left-out subject, to derive its M-score.

3.3.4 Hypotheses and statistical analyses

With the clinician who performed the visual pursuit assessment in all tests of our system, we discussed ways to effectively evaluate the similarity between the various assessment outcomes collected from different sources: the clinician, the consensus by DOC experts on video (gold standard), and the C-score and M-score automatically provided by our system. All outcomes could be compared at the global level (i.e., the presence or absence of visual pursuit, based on the CRS-R criteria of at least two successful trials), and most outcomes could be compared at the trial level (i.e., the success or failure in following a single mirror movement). This discussion led to hypotheses that were formally defined by the clinician (who also performed all statistical analyses below), as follows.

- A high congruence is expected, both at the global level and the trial level, between the consensus by DOC experts on video (gold standard) and the clinical assessment of visual pursuit. This hypothesis is tested using the Cohen's kappa coefficient (measure of the inter-rater agreement for categorical items).

- A high congruence is expected at the global level between the consensus by DOC experts on video (gold standard) and the decisions obtained via both the C-score and the M-score. This hypothesis is tested using the Cohen's kappa coefficient.

- A high congruence is expected at the trial level between the consensus by DOC experts on video (gold standard) and the M-score trial-level decisions. This hypothesis is tested using the Cohen's kappa coefficient.

- A significant positive correlation is expected between the gold standard objective score (proportion of successful trials, according to the consensus by DOC experts on video), and both of the C-score and the M-score. This hypothesis is tested using the Spearman correlation.

The interpretation of the Cohen's kappa coefficient was done according to the recommendations in [116], i.e., the agreement was classified as poor (< 0), slight ($\in [0, 0.2]$), fair ($\in [0.21, 0.4]$), moderate ($\in [0.41, 0.6]$), substantial ($\in [0.61, 0.8]$), or almost perfect ($\in [0.81, 1]$).

Additionally, the sensitivity and the specificity of both of the-global level C- and M-score-based decision were calculated, again with respect to the gold standard expert consensus. In the present study, the sensitivity of the decision based on a score consists of the reliability in using this score to declare the absence of visual pursuit in a subject (true positive rate in a test for a disease). Conversely, the specificity of a score-based decision consists of the reliability in using this score to declare the presence of visual pursuit (true negative rate in a test for a disease).

All statistical analyses were performed separately for the tests with DOC patients and for the tests with the healthy control subjects (including the additional tests with fixed and random gaze instructions). The obtained results were considered to be significant at $p < 0.05$.

3.3.5 Results

Healthy control subjects

Before testing the hypotheses given in the previous section, we give here some preliminary results that are indicative of the effectiveness of the C-score with healthy control subjects[6]. Figure 3.9 presents, as box plots, the distributions of the C-score obtained with our system for the test groups defined for these subjects. For participants of the first group (CS1, 23 tests), which were instructed to follow the mirror, the median score is 0.92, the maximum score is 0.96, and the minimum score is 0.79, which corresponds to the only outlier of the group. For participants of the second group (CS2, 17 tests), which were instructed to keep a fixed gaze, the median score is 0.01, the minimum score is 0.0, and the maximum score is 0.25, which corresponds to the only outlier of the group. For participants of the third group (CS3, 10 tests), which were instructed to do random eye movements, the median and minimum scores are both 0.0 and the maximum score is 0.31, which corresponds to the only outlier of the group. Overall, the C-scores for the three healthy control subject test groups are as expected, according to the instructions given by the clinician, i.e., close to the maximum value of 1.0 for CS1 and close to the minimum value of 0.0 for CS2 and CS3. After visual inspection of the videos corresponding to the outliers in groups CS2 and CS3, we observed that these subjects had difficulties to ignore the moving mirror, showing brief, self-restrained intentions to follow it. On the other hand, the outlier subject in group CS1 performed a visual pursuit with pronounced saccadic eye movements, for an unknown reason. We also show in Fig. 3.10 the time evolution of the average C-score over a visual pursuit assessment, for each of the three groups CS1, CS2, and CS3. This figure shows that the C-score quickly discriminates

[6]Note that these preliminary results with healthy control subjects were obtained by calculating the C-score on the full length of the pupil and mirror trajectories extracted from the videos of the assessments, and not solely on the 0 to 45 degrees trajectory segments of interest (see Sect. 3.2.5), which explains the mild numerical discrepancies between these preliminary results and the other results presented for the healthy control subjects in this section.

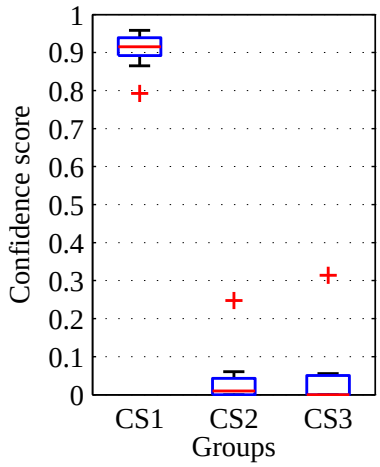


FIGURE 3.9: Box plots of the C-score distributions obtained with our system for the healthy control subject test groups CS1 (tracking gaze instruction), CS2 (fixed gaze instruction), and CS3 (random gaze instruction).

between the presence and absence of visual pursuit in the tests designed with the healthy control subjects. Indeed, the average C-score reaches above 0.9 for the CS1 group, which was instructed to follow the mirror, and below 0.1 for the CS2 and CS3 groups, which were instructed to not follow it, when barely half of the total assessment has been performed.

We now expose the statistical analyses performed to test the hypotheses given in Sect. 3.3.4, on the assessment outcomes with the healthy control subjects. At the trial level, an almost perfect agreement was observed between the decisions made by the research neuropsychologist during the assessment and the video-based decisions made by the same neuropsychologist, according to the kappa statistic (κ = 0.98, based on the 400 trials done with the subjects); a disagreement was observed in four trials (1%). An almost perfect agreement was also observed between the decisions made by the research neuropsychologist during the assessment and the video-based decisions made by the consensus by DOC experts (κ = 0.98); a disagreement was also observed in four trials (1%). At the global level, an almost perfect agreement was observed between the decisions made by the research neuropsychologist during the assessment and the video-based decisions made by the same neuropsychologist (κ = 0.96, based on 50 tests); a disagreement was observed in one test (2%). An almost perfect agreement was also observed between the decisions made by the research neuropsychologist during the assessment and the video-based decisions made by the consensus by DOC experts (κ = 0.96); a disagreement was observed in one test (2%).

As for the objective scores automatically provided by our system, an almost perfect agreement was observed between the global video-based decisions made by the consensus by DOC experts (gold standard) and the decisions made via the C-score (κ = 0.92, based on 50 tests); a disagreement was observed in two subjects (4%). The proportion of succeeded trials based on the consensus by DOC experts (gold standard objective score) and the C-score correlated significantly (Spearman r = 0.891, p < 0.001; see Fig. 3.11, part A). The sensitivity of the C-score was 96.1%, and the specificity 95.8%. Regarding the M-score, a perfect agreement was observed with the video-based decisions made by the consensus by DOC experts (gold standard) at the trial level (κ

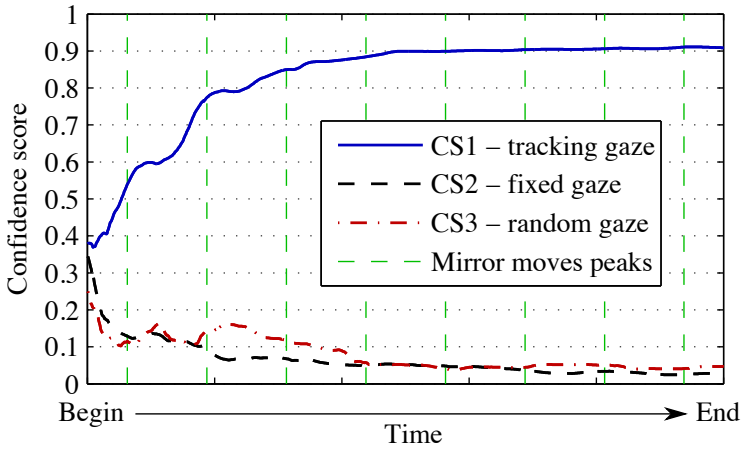


FIGURE 3.10: Time evolution, over the assessment procedure, of the average confidence score (C-score) in each of the healthy control subject test groups. The eight green vertical dashed lines correspond to the approximate moments when the clinician reaches 45 degrees in a trial mirror movement, in either of the leftward, rightward, upward, or downward directions.

= 1, 0% disagreement), as well as at the global level (κ = 1, 0% disagreement). The M-score significantly and perfectly correlated with the proportion of succeeded trials based on the consensus by DOC experts, i.e., the gold standard objective score (Spearman r = 1, p > 0.001; see Fig. 3.11, part B). The sensitivity and the specificity of the M-score both reached 100%.

DOC patients

In the following, we expose the statistical analyses performed to test the hypotheses given in Sect. 3.3.4, on the assessment outcomes with the DOC patients. At the trial level, an almost perfect agreement was observed between the decisions made by the research neuropsychologist during the assessment at bedside and the video-based decisions made by the same neuropsychologist, according to the kappa statistic (κ = 0.864, based on the 248 trials done with the patients); a disagreement was observed in 14 trials (5.7%). An almost perfect agreement was also observed between the decisions made by the research neuropsychologist during the assessment at bedside and the video-based decisions made by the consensus by DOC experts (κ = 0.859); a disagreement was observed in 14 trials (5.7%). The kappa's relative to the different diagnostic subgroups were also calculated (see Tab. 3.2), except in the UWS subgroup, as these patients do not show a visual pursuit at bedside, by definition, so that no reliable statistical indices could be calculated. Moreover, no reliable visual pursuit was detected on video by the consensus by DOC experts. At the global level, an almost perfect agreement was observed between the decisions made by the research neuropsychologist during the assessment at bedside and the video-based decisions made by the same neuropsychologist (κ = 0.871, based on 31 patients); a disagreement was observed in two patients (6.5%). A substantial agreement was observed between the decisions made by the research neuropsychologist during the assessment at bedside and the video-based decisions made by the consensus by DOC experts (κ = 0.805); a disagreement was observed in three patients (9.7%). All global-level decisions made by human experts with DOC patients are also included in Tab. 3.1.

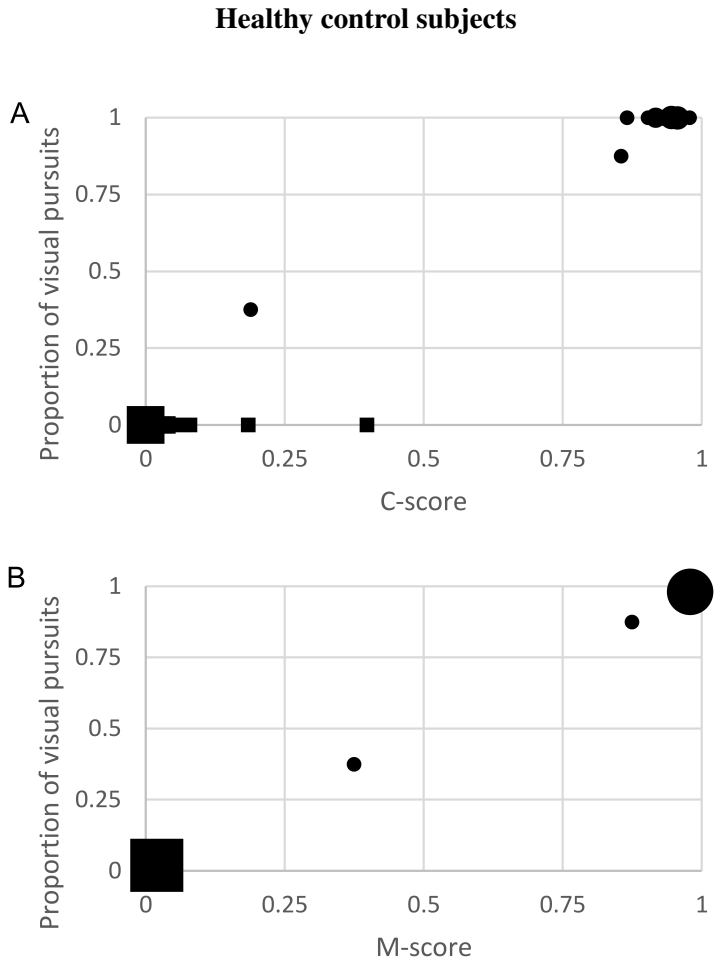


FIGURE 3.11: Scatterplots representing the correlation, for healthy control subjects, between the objective scores (C- and M-) provided by our system and the gold standard objective score based on the consensus by DOC experts. Part A: correlation between the C-score and the consensus by DOC experts. Part B: correlation between the M-score and the consensus by DOC experts. Dots represent the tests where visual pursuit was declared, according to the consensus by DOC experts. Squares represent the tests where the absence of visual pursuit was declared. The difference in size of squares or dots represents the amount of tests with similar results. Figure 3.12 gives the corresponding plots for DOC patients.

TABLE 3.2: Kappa, sensitivity, and specificity related to relevant comparisons between two measures of the visual pursuit ability, among which (1) the *bedside* assessment, (2) the *video* scoring by the clinician who did the bedside assessment, (3) the *consensus* by three DOC experts on video, and (4) the *M-score* provided by our system. The consensus by DOC experts on video is here considered as the gold standard, and therefore as our reference measure. The sensitivity and specificity for the *bedside* vs. *video* comparison are not reported, because no reference is available is these cases (the *consensus* gold standard is not involved).

Kappa	**MCS-**	**MCS+**	**EMCS**
Bedside vs. video	0.815	0.864	0.944
Bedside vs. consensus	0.789	0.864	1
M-score vs. consensus	0.902	1	0.894
Sensitivity	**MCS-**	**MCS+**	**EMCS**
Bedside vs. consensus	97%	80%	100%
M-score vs. consensus	94%	100%	92%
Specificity	**MCS-**	**MCS+**	**EMCS**
Bedside vs. consensus	85%	100%	100%
M-score vs. consensus	96%	100%	100%

As for the objective scores automatically provided by our system, a moderate agreement was observed between the global video-based decisions made by the consensus by DOC experts (gold standard) and the decisions made via the C-score (κ = 0.516, based on 31 patient tests); a disagreement was observed in eight patients (25.8%). The proportion of succeeded trials based on the consensus by DOC experts (gold standard objective score) and the C-score correlated significantly (Spearman r = 0.83, p < 0.001; see Fig. 3.12, part A). The sensitivity of the C-score was 100%, while the specificity was 60%. Regarding the M-score, an almost perfect agreement was observed with the video-based decisions made by the consensus by DOC experts (gold standard) at the trial level (κ = 0.907); a disagreement was observed in nine trials (3.6% disagreement). A perfect agreement was observed with the consensus-based gold standard at the global, patient level (κ = 1, 0% disagreement). The M-score significantly correlated with the proportions of successful trials based on the consensus by DOC experts, i.e., the gold standard objective score (Spearman r = 0.913, p < 0.001; see Fig. 3.12, part B). At the global level, the sensitivity and the specificity of the M-score both reached 100%. Kappa, sensitivity, and specificity were also calculated for each diagnosis subgroup (see Tab. 3.2). Again, the UWS subgroup was not included as the absence of visual pursuit is a necessary criterion for this state, thus, no reliable statistical indices could be calculated. Moreover, no reliable visual pursuit was detected by the M-score. All global-level automatic decisions made via either the C-score or the M-score provided by our system with DOC patients are also included in Tab. 3.1.

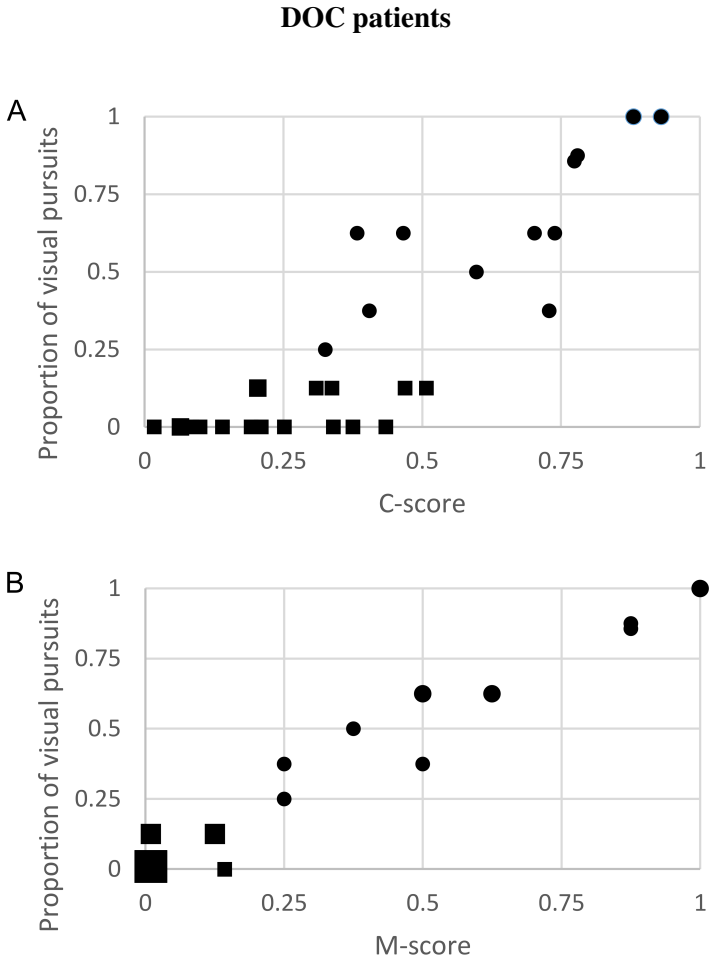


FIGURE 3.12: Scatterplots representing the correlation, for DOC patients, between the objective scores (C- and M-) provided by our system and the gold standard objective score based on the consensus by DOC experts. Same types of plots as in Fig. 3.11, but for DOC patients.

3.4 Conclusion

Visual pursuit is one of the first signs of recovery of consciousness in patients with DOC [86], and is actually sufficient in itself to differentiate between UWS, i.e., unconscious, and MCS, i.e., conscious patients. The CRS-R, which is currently the gold standard behavioral scale for the assessment of consciousness [91, 97], provides precise guidelines for the assessment of visual pursuit. In particular, the CRS-R recommends that a clinician move a handheld mirror in multiple trials in front of the patient's face. It was shown that this autoreferential stimulus was especially effective to detect visual pursuit in DOC patients [97, 96]. However, the assessment of visual pursuit by a single, yet experimented clinician remains a subjective process. In delicate, complex cases of visual pursuit assessment, the sensitivity of the CRS-R procedure to subjectivity may cause errors, as we showed in the results section of this chapter. Indeed, in the assessments of our DOC patient cohort, around 6% of the trials were scored differently by the clinician (an experienced research neuropsychologist) as compared to when the corresponding videos were scored afterwards by three DOC experts (including the clinician herself) who had to reach an agreement. These trial-level errors resulted in up to 10% error in detecting overall visual pursuit in the patient cohort, according to the consensus by three experts on video. This is of great importance, because such errors may lead to wrongly diagnose UWS in DOC patients, therefore influencing the course of their treatment, including the development of a rehabilitation plan, the management of pain, and the end-of-life decisions [117, 118, 119]. Involving several observers with experience in DOC to assess a patient is not easy to implement in the clinical practice, or even via an offline video-based assessment procedure, notably because it is time-consuming and requires a lot of human resources. For all these reasons, an objective and "plug-and-play" measurement tool of the visual pursuit ability is of particular interest.

In this chapter, we presented our computer vision-based system designed to assist clinicians in their assessment of visual pursuit in patients with DOC. The first objective score we proposed with our system, i.e., the C-score, is based on a correlation analysis on the trajectories of the patient's pupil and the handheld mirror stimulus. The C-score misclassified 26% of the patients in our cohort, according to the consensus by three DOC experts on video. The main limitation of the C-score was its lack of specificity, i.e., the moderate reliability in using this score to declare the presence of visual pursuit in a subject (true negative rate in a test for a disease), especially in the patient group. In healthy control subjects, a high C-score value was observed in a subject who was actually not following the stimulus, having been instructed to keep a fixed gaze. This suggests that the C-score may be affected by unrelated and/or tiny eye movements, leading to an increase in the C-score value despite the absence of visual pursuit. The second objective score we proposed with our system, i.e., the M-score, is based on machine learning, and involves the classification of each trial of the assessment as successful or not, on the basis of the pupil and mirror trajectories for this trial. The M-score correctly classified all of the patients and healthy control subjects, according to the consensus by experts on video. This suggests that our system can provide a measure of the visual pursuit ability with the same performance as three DOC experts examining a patient. When testing the M-score at the trial level (i.e., comparing the trial-level classifier predictions to the trial-level decisions made by the consensus by experts), the results were very encouraging in

all of the different diagnostic subgroups, with the Cohen's kappa coefficient ranging from 0.815 to 1, and the sensitivity and specificity ranging from 80% to 100%. The subgroup of patients in UWS was not included in this statistical analysis, as UWS patients did not show visual pursuit. It is however important to note that neither the video-based scoring by experts nor the M-score detected any sustained visual pursuit that would have changed the diagnosis from UWS to MCS in the DOC patient cohort.

Our results suggest that the system we presented in this chapter could be of great interest for clinicians. First, jointly used with a head-mounted device, our system is adapted for bedside assessment without any constraint for the patient, and it also allows the clinician to assess visual pursuit with a handheld mirror, as recommended by the CRS-R. Second, and more importantly, the fact that we could obtain meaningful results from our DOC patient cohort suggests that our system could be an effective tool to supplement classical bedside assessment in this population in an objective way. Incidentally, our system also reliably detected the few trials for which healthy control subjects did not perfectly comply with the instruction to not follow the moving mirror. Indeed, these particular trials were detected during the video scoring by experts, as well as by the M-score we proposed with our system.

We believe that, overall, our system is a good first step toward objectively assessing visual pursuit. However, it is important to note that, even though we considered in our work that a consensus by three DOC experts best approaches the "true" objective measure for an assessment, this consensus is still essentially based on subjective observations. Also, it is worth mentioning that using close-up videos of the eye reduces the information that is available at bedside about the patient. The clinician actually perceives global information about the patient's face during the assessment, such as a head movements and other facial expressions that may influence his/her judgment. Yet it is unknown whether or to what extent such information, which is not used by our system, could be beneficial to the objective assessment of visual pursuit. Finally, a possible caveat of visual pursuit assessment with our system and a head-mounted device may be the change of appearance of the subject's face caused by the wearing of this head-mounted device. Indeed, the effectiveness of the mirror stimulus is presumed to be due to its autoreferential aspect [97]. It should be noted that, for one patient, after a test with our system where no visual pursuit could be observed, a visual pursuit could be observed in another assessment with the head-mounted device removed. We cannot exclude that this inconsistency may have been caused by the change of appearance of the patient due to the wearing of the head-mounted device, even though an alternative explanation for this might be the vigilance fluctuation, or tiredness of the patient.

Chapter 4

Hierarchical vs. flat classification in vision-based recognition problems

In contrast to the standard, flat classification approach, it is proposed in the hierarchical approach to incorporate a semantic class hierarchy within the learning stage of a classifier, so as to enforce a fine-grained notion of semantic similarity between the classes. The objective in doing so is to guide the learning process to discover an overall better classifier than one obtained with flat classification. In this chapter, we present the empirical study we conducted on hierarchical vs. flat classification, used in the vision-based recognition problems of facial expression recognition and 3D shape recognition. Through our experiments on these problems, we found that, unexpectedly, there was little to no improvement in the recognition performance by using hierarchical classification with visual features provided by off-the-shelf feature extraction methods. Through additional experiments we conducted in a simulation, we found general conditions about feature representations and semantic class hierarchies that should be met in order to get the benefits of using hierarchical classification. Everything presented in this chapter is our own work, but for a few, explicitly stated exceptions. Our work was also published in its entirety in a special issue of the journal "Machine Vision and Applications" [120], as an extended version of our early work published in the proceedings of the "16th International Conference on Computer Analysis of Images and Patterns" [121]. The few differences between the content of this chapter and our publications in [120] and [121] consist of mild variations in our formulation of the concepts, and the emphasis we give in this chapter to the problem of facial expression recognition.

4.1 Introduction

In supervised classification, an algorithmic classification rule is to be learned automatically, on the basis of class-labeled data exemplars. The learned classification rule, i.e., classifier, can then be used to automatically infer the class labels of new data. The learning stage of a classifier is often referred to as training, and is preceded by the choosing of a training method suited to exploit class-labeled data exemplars. The inference stage of a classifier may be referred to as testing, when the goal is to evaluate the performance of the classifier on new data. Before going through these two stages, it is of course essential to specify the classes of the particular classification problem of interest. In the standard approach to supervised classification, the specification of the problem classes is only focused on their number. Specifying the semantic relationships between the classes is not part of the standard formulation of a classification problem. In fact, the standard approach

implicitly presupposes that any two problem classes share the same degree of semantic similarity or dissimilarity. A standard binary, resp. multiclass, problem formulation only indicates the presence of two, resp. several, possible classes that are equally dissimilar from each other. Also, a standard multilabel problem formulation only indicates that any possible combination of several class labels may be associated with a single data instance. Consequently, in the standard approach, any possibly useful semantic relationship between the problem classes has to be somehow discovered from the labeled data that are available for training. It is indeed assumed that any semantic information about two problem classes, such as their similarity, or their validity of co-occurrence to describe a data instance, is implicitly present in the labeled data. Most of the theoretical and practical contributions in supervised classification have been dedicated to this standard classification approach [122].

In many classification problems, however, the semantic relationships between classes can be explicitly specified before the learning stage, often with little additional effort. For instance, a vision-based classification problem involving the "bee", "ant", and "hammer" classes could be given the explicit information that an ant and a bee are visually more similar to each other than an ant and a hammer, or a bee and hammer. One simple way of doing so is to add an "insect" superclass to the class specification, which superclass includes the "ant" and "bee" classes, but excludes the "hammer" class. By using a training method designed to exploit such prior hierarchical semantic information, it is intuitive that the overall performance of the resulting classifier would be improved, in particular if the semantic relationships between classes are not well represented implicitly via the available training exemplars. Motivated by this intuition, a new, hierarchical classification approach has emerged for dealing with the classification of data deemed to be inherently semantically hierarchical. The training methods developed following this new approach all exploit prior hierarchical semantic information given in the form of superclasses specified at different semantic hierarchy levels [122]. The development of hierarchical classification methods also benefited from the advances made in the field of machine learning generalized to arbitrary output spaces, also known as the structured output classification approach, of which the hierarchical approach is actually a special case [123]. In contrast to the hierarchical classification approach, the standard classification approach is called "flat", because it implicitly considers that the problem classes all belong to the same unique hierarchy level, i.e., the classes are the leaves of an unspecified semantic class hierarchy in a standard classification problem formulation.

In several application domains, the explicit specification and use of a semantic class hierarchy has been shown to be key to improve the classification performance, e.g., in text categorization [124], protein function prediction [125], and music genre classification [126]. For the classification of visual content, using prior hierarchical information intuitively seems particularly appropriate as it reflects the natural way in which humans organize and recognize the objects they see, according to neurophysiological studies of the visual cortex [127, 128, 129]. In practice, some results suggest that there is indeed a gain in performance by using hierarchical classification in vision-based recognition problems, e.g., in 3D object shape recognition [130], and automatic annotation of medical images [131]. A line of research analogous to the hierarchical classification of visual content is the automatic construction of class hierarchies from image databases, to effectively organize them. The methods used in this research line exploit either the image content [132,

133], or the image tag labels (when these happen to be available within the database) [134, 135], or both, as proposed in [136], where a "semantivisual" hierarchy that is semantically meaningful and close to the visual content is learned. Overall, many research results motivate to further investigate the potential of incorporating prior hierarchical information within classification problems based on visual content. Our motivation in this chapter is therefore to conduct such an investigation, and we fairly compare hierarchical and flat classification performance empirically in vision-based recognition problems typically solved using the standard, flat classification approach.

One such problem is the recognition of the facial muscle contraction patterns, which is of particular interest in this thesis about the automation of tasks of facial expression interpretation. Indeed, this problem is often simply called facial expression recognition, because recognizing facial muscle contraction patterns is often considered equivalent to recognizing facial expressions themselves objectively, without interpreting them [36]. In some applications, this problem is posed as an upstream stage toward further automating a task of facial expression interpretation, e.g., typically, emotion recognition [137]. This problem is also involved in applications that seemingly do not deal with facial expression interpretation, e.g., animated face avatars for video-conferencing systems, with the purpose of assuring the anonymity of the user while *a piori* preserving facial communication cues [59]. In our understanding, however, and regardless of our following study on hierarchical vs. flat classification performance in vision-based recognition problems, the so-called problem of facial expression recognition is actually already associated with a task of facial expression interpretation. For one thing, facial expressions are not limited to the action of the facial muscles, as head and eye movements may be part of a facial expression according to our discussion and definition in Chap. 1. Also, recognizing a facial muscle contraction pattern is visually ambiguous and relies to some extent on evidence that is not distinctly visible in the face. Even among trained human experts, visually determining which exact muscles are in action in a person's face, and with which exact intensity, may not result in a perfect agreement, notably because separating the muscle-based facial dynamics from the static face characteristics requires some degree of familiarity of the observer with the people under observation. Nevertheless, we use the conventional term of "facial expression recognition" in this chapter to refer to the vision-based problem associated with the face perception task of recognizing the facial muscle contraction patterns, which patterns we also call "facial expressions", for simplicity.

To prepare our hierachical vs. flat classification experiment in the problem of facial expression recognition, we consider that the various muscle contractions composing a facial expression can be legitimately organized in a hierarchical fashion. We therefore specify a semantic class hierarchy for organizing these contractions, and we choose two different, sound hierarchical classification methods that can exploit our proposed class hierarchy, as well as their two respective flat counterpart methods that cannot exploit this hierarchy. We feed all methods with the same face features, obtained with off-the-shelf face feature extraction methods. We fairly compare the classification performance of all methods using appropriate measures designed for evaluating both hierarchical and flat classification performance. The results of this experiment lead us to conclude that, contrary to our expectations, the hierarchical approach provides no performance improvement in our problem of facial expression recognition [120, 121]. We then consider a second, fairly

different, yet popular vision-based recognition problem, to continue our empirical study on hierarchical vs. flat classification of visual content. This second problem is 3D shape recognition, i.e., the classification of an object from its 3D shape data. Such data can be obtained from either a depth image of the object, or a computer aided-design (CAD) shape model of the object. We fairly compare hierarchical vs. flat classification performance in this problem, on the basis of the same hierarchical and flat classification methods, and same evaluation measures as we used in our problem of facial expression recognition. It is noteworthy that, compared to facial expression recognition, the classes of our problem of 3D shape recognition may be more suited to be organized in a hierarchical fashion, and we use for this problem a semantic class hierarchy that was previously proposed in the literature [138, 130]. We feed our hierarchical and flat classification methods with features obtained using several popular 3D shape descriptors, to increase the chance to observe a performance improvement with hierarchical classification. However, our conclusion from the results of this second experiment is that, as in our problem of facial expression recognition, hierarchical classification unexpectedly provides no significant performance improvement in the resolution of our problem of 3D shape recognition [120, 121]. Despite all the care taken, we could not showcase the superiority of the hierarchical approach in either of our experiments on the classification of visual content. We conjecture that, overall, these relatively poor hierarchical classification results may be caused by the inadequacy of the off-the-shelf face and 3D shape features we used in our experiments, and that richer visual feature representations may be necessary to exploit the prior hierarchical information we provided to our hierarchical classification methods [120, 121].

In the last part of our empirical study, we seek to explain why hierarchical classification could not outperform flat classification in our problems of facial expression recognition and 3D shape recognition. More generally, we wish to find the conditions in which the hierarchical classification approach could consistently provide a better performance than the flat classification approach. We wish in particular to test our hypothesis about the importance to use feature representations that are rich and well-suited to hierarchical classification. We therefore design a simulation framework where rich hierarchical feature representations can be constructed, and we again conduct the comparative evaluation of our hierarchical and flat classification methods, this time in artificial classification problems generated with our simulation framework. In more detail, we simulate problems about which we can control aspects that are key to hierarchical classification, such as the true, underlying hierarchical nature of the phenomenon being measured, the amount of noise in the features extracted from the measurements, and the adequacy of the hierarchical semantic information that the observer perceives about the phenomenon. From the results of this simulation experiment, we conclude that using rich hierarchical feature representations is indeed crucial for obtaining a performance gain with the hierarchical classification approach, and furthermore that the specification of class hierarchies with semantic errors may seriously hinder this gain even though proper hierarchical feature representations are used [120].

4.2 Hierarchical classification

4.2.1 Framework and terminology

Recently, a necessary effort to unify the hierarchical classification framework was made by [122]. We follow their terminology, which is summarized next.

Formally, a semantic class hierarchy, or *class taxonomy* $\{\mathcal{C}, \prec\}$, consists of a finite set of semantic concepts $\mathcal{C} = \{c_i \mid i = 1, \ldots, n\}$ and a partial order relationship $\prec$ organizing these concepts either in a tree or in a directed acyclic graph (DAG). The partial order relationship can be seen as an embodiment of the "IS-A" or "PART-OF" relationship between the semantic concepts under consideration. A classification problem defined over such a taxonomy is called a *hierarchical classification problem*: its classes and superclasses correspond to the leaf nodes and interior nodes of the tree (or DAG), respectively, and all classes and superclasses are considered in the classification problem. In contrast, a *flat classification problem* only considers the leaf nodes of such a taxonomy as its classes, and does not consider the interior nodes corresponding to the superclasses. A flat classification problem therefore deals with classes that are virtually at the same single semantic level, whence the term "flat" for such problem.

In the definition of a (supervised) hierarchical classification problem, every data instance is labeled with a subset of the class taxonomy. Such subset must satisfy the partial order relationship $\prec$, which means that the inclusion of any node within the taxonomic label implies the inclusion of its parent node—or at least one of its parent nodes, if a DAG structure is used—and the parent node(s) of it (them), up to reaching the root node. Consequently, in addition to which structure (between tree or DAG) is used in the problem definition, a hierarchical classification problem can also be defined according to two more properties: (1) whether the problem uses *single-path* labeling or *multiple-path* labeling, i.e., whether or not every data instance is labeled with no more than one single path in the taxonomy structure, and (2) whether the problem uses *full depth* labeling or *partial depth* labeling, i.e. whether or not every data instance is labeled with at least one leaf node, i.e., has a label that covers all hierarchy levels of the taxonomy structure according to the partial order relationship. In all cases, and without loss of generality, any taxonomic label $\mathbf{y}$ for a data instance can be formalized using an indicator vector notation, i.e., $\mathbf{y} \in \mathcal{Y} \subset \{0, 1\}^n$, where the i^{th} component of $\mathbf{y}$ takes value 1 if the data instance belongs to the class or superclass $c_i \in \mathcal{C}$, and 0 otherwise, and where all elements in $\mathcal{Y}$ satisfy $\prec$. It is noteworthy that this notation is equally convenient to deal with flat classification labels.

The real and simulation problems considered in this study are all defined using tree taxonomies with full depth labeling. For the facial expression recognition problem, we define multiple-path labeling (see Sect. 4.3.1), whereas for the 3D shape recognition problem and for our simulation problems we define single-path labeling (see Sect. 4.3.2 and Sect. 4.4.2).

Because they do not penalize structural errors, evaluation measures commonly used in the standard flat classification approach may not be appropriate when comparing hierarchical methods to each other, or flat methods to hierarchical methods. In particular, standard evaluation measures do usually not consider that misclassification occuring at different levels of the taxonomy should be treated in different ways, to reflect a policy where a higher penalty is given to errors made at higher semantic levels. In this study, we adopt the following hierarchical evaluation measures

proposed by [139], also recommended by [122]: hierarchical precision (*hP*), hierarchical recall (*hR*), and hierarchical F-measure (*hF*). They are defined as

$$hP = \frac{\sum_i |\hat{P}_i \cap \hat{T}_i|}{\sum_i |\hat{P}_i|} \quad , \quad hR = \frac{\sum_i |\hat{P}_i \cap \hat{T}_i|}{\sum_i |\hat{T}_i|} \quad , \quad hF = \frac{2\, hP\, hR}{hP + hR} \quad , \tag{4.1}$$

where $\hat{P}_i$ is the set of the most specific class(es) predicted for a test data instance i and all its (their) ancestor classes, and $\hat{T}_i$ is the set of the true most specific class(es) of a test data instance i and all its (their) ancestor classes. These hierarchical evaluation measures are extensions of the standard precision, recall and F-measure, and reduce to them as special cases for flat classification problems.

4.2.2 Hierarchical classification methods

We present in this section the two hierarchical classification methods we use to study the possible benefits of the hierarchical approach for facial expression recognition, as well as for one other real and several simulated classification problems. Since we want to evaluate these possible benefits in comparison to the standard, flat classification approach, we choose hierarchical classification methods that have flat methods as clear counterparts. Thus, our hierarchical methods are sound generalizations to standard, flat classification methods. Our hierarchical methods of interest are said to be *global*, because for each problem they produce a single classifier that deals with complete taxonomic labels. Indeed, our hierarchical methods consider the class taxonomy at once, in both the training and testing stages. This contrasts with other strategies where multiple classifiers are trained to deal locally with one taxonomy node or one taxonomy level, i.e., so-called *local* hierarchical methods. The compounds of classifiers typically produced by such local methods correspond to a complex classification rule that does not necessarily respect the pre-established partial order relationship in its output [122]. Additionally, because nothing prevents fundamentally different classification methods to be used at the local level, such local hierarchical methods lack a clear flat counterpart against which to be compared. Overall, local hierarchical classifiers are more difficult to analyze and the local approach to hierarchical classification is therefore not examined in this study.

Structured output k-nearest neighbors

For our first hierarchical classification method of interest, we modify the standard k-nearest neighbors (kNN) method to allow it to cope with outputs having general structure, e.g., taxonomic labels defined using a pre-established class taxonomy $\{\mathcal{C}, \prec\}$. We call the resulting classification method structured output k-nearest neighbors (SkNN).

Let $\mathcal{D} \subset \mathcal{X} \times \mathcal{Y}$ be the training set of a hierarchical classification problem, where $\mathcal{X}$ is the set of all possible data instances, and $\mathcal{Y}$ is the set of all possible indicator vectors corresponding to the taxonomic labels defined by $\{\mathcal{C}, \prec\}$. The SkNN classifier is trained in the same way as the standard kNN classifier, i.e., by projecting the training data instances into a feature space using a feature map $\boldsymbol{\phi}(\mathbf{x}) \,|\, \mathbf{x} \in \mathcal{X}$, and keeping the record of their corresponding structured

labels (in the hierarchical case, their taxonomic labels). Then, given the k nearest neighbors[1] $\mathcal{N} = \{(\mathbf{x}_i, \mathbf{y}_i) \,|\, i \in \{1, \ldots, k\}\} \subset \mathcal{D}$ to a test data instance $\mathbf{x} \in \mathcal{X}$, those nearest neighbors being found according to a distance metric $\rho(\boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}_i))$, the SkNN classification rule that gives the structured label (in the hierarchical case, the taxonomic label) $\hat{\mathbf{y}}$ as a prediction for the test data instance $\mathbf{x}$ is

$$\hat{\mathbf{y}}(\mathbf{x}; \mathcal{N}) = \underset{\mathbf{y} \in \mathcal{Y}}{\operatorname{argmax}} \left\langle \sum_{i=1}^{k} \omega_i \frac{\mathbf{y}_i}{||\mathbf{y}_i||}, \frac{\mathbf{y}}{||\mathbf{y}||} \right\rangle, \tag{4.2}$$

where ω_i are weights attributed to the elements of $\mathcal{N}$. In our experiments, we choose these weights so as to reflect the distances of the nearest neighbors to the test data instance, i.e., $\omega_i = 1/\rho(\boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}_i))$. Also, we choose to use the L^2 norm for our distance metric ρ.

Structured output support vector machine

Our second hierarchical classification method is a customization of the structured output support vector machine (SSVM, proposed in [123]). SSVM extends the standard support vector machine (SVM) in order to cope with arbitrary output spaces with non-trivial structure, and defines the relationship between a test data instance $\mathbf{x} \in \mathcal{X}$ and its prediction $\hat{\mathbf{y}} \in \mathcal{Y}$ on the basis of a joint score maximization,

$$\hat{\mathbf{y}}(\mathbf{x}; \mathbf{w}) = \underset{\mathbf{y} \in \mathcal{Y}}{\operatorname{argmax}} \big\langle \mathbf{w},\ \boldsymbol{\psi}(\mathbf{x}, \mathbf{y}) \big\rangle, \tag{4.3}$$

where $\mathbf{w} \in \mathbb{R}^d$ is a learned parameter vector, and where the user-defined joint feature map $\boldsymbol{\psi} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ projects any couple $(\mathbf{x}, \mathbf{y})$ to its real-valued vectorial representation in a joint feature space. The role of the linear function $\big\langle \mathbf{w},\ \boldsymbol{\psi}(\mathbf{x}, \mathbf{y}) \big\rangle$ is to produce a joint score that reflects how well a data instance $\mathbf{x}$ matches a structured label $\mathbf{y}$, i.e., the higher the joint score, the better the match.

With this formulation of the inference part in SSVM, the data and labels can virtually represent anything, provided that an appropriate encoding of the structured output domain $\mathcal{Y}$ is used, e.g., an indicator vector formulation. It comes however at the cost that the inference part in SSVM is an optimization problem in itself (similarly to the inference with SkNN in Eq. 4.2), which makes it a quite complex task to learn an SSVM classifier. Learning an SSVM classifier consists of finding the parameter vector $\mathbf{w}$ that separates the training joint feature representations from the origin of the joint feature space by the largest margin. In the straightforward, naive formulation of the learning optimization problem, it comes that the loss function $\Delta(\mathbf{y}_i, \hat{\mathbf{y}}(\mathbf{x}_i; \mathbf{w}))$ that measures the fit quality of $\mathbf{w}$ for a training example $(\mathbf{x}_i, \mathbf{y}_i)$ is non-convex, and is composed with the argmax inference problem in Eq. 4.3 (the second argument of this loss function). Hence, the authors of [123] propose the use of convex surrogates for the loss. The learning problem for SSVM therefore becomes the minimization, with respect to $\mathbf{w}$, of the following convex objective function,

$$\frac{1}{2}||\mathbf{w}||_2^2 + \frac{C}{n} \sum_{i=1}^{n} L_i(\mathbf{w}), \tag{4.4}$$

[1] In this chapter, the notation $\mathbf{x}_i$ is not used to designate the i^{th} component of some vector $\mathbf{x}$, but rather the i^{th} element of an ordered set $\mathcal{X}$. The i^{th} component of a vector $\mathbf{x}$ would instead be written as x_i.

where C is a parameter which, as in the usual SVM soft-margin approach, balances the allowed misclassification rate; where n is the number of couples in the training set $\mathcal{D} \subset \mathcal{X} \times \mathcal{Y}$, and where $L_i(\mathbf{w}) \approx \Delta(\mathbf{y}_i, \hat{\mathbf{y}}(\mathbf{x}_i; \mathbf{w}))$ is a convex approximation of the original loss, i.e., a surrogate loss, which is chosen to have a bounding property: $\Delta(\mathbf{y}_i, \hat{\mathbf{y}}(\mathbf{x}_i; \mathbf{w})) \leq L_i(\mathbf{w})$. Good surrogates should be a tight upper bound of the original loss so that minimizing them has the same effect as minimizing the original loss. Standard construction methods that give good surrogates do exist, e.g., the margin rescaling method, where $L_i(\mathbf{w}) = \sup_{\mathbf{y} \in \mathcal{Y}} \Delta(\mathbf{y}_i, \mathbf{y}) + \langle \mathbf{w},\, \boldsymbol{\psi}(\mathbf{x}_i, \mathbf{y}) \rangle - \langle \mathbf{w},\, \boldsymbol{\psi}(\mathbf{x}_i, \mathbf{y}_i) \rangle$, which we use in this study.

For our study, we use the SVM$^{\text{struct}}$ implementation provided by the authors of [123], along with its MATLAB code wrapper [140]. This implementation is a customizable SSVM framework which allows to define the various necessary components needed to get actually working SSVM learning and inference algorithms. In particular, in our customized hierarchical SSVM, we define the joint feature map that projects any couple $(\mathbf{x}, \mathbf{y})$ in the joint feature space, as

$$\boldsymbol{\psi} : \quad \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d, \quad (\mathbf{x}, \mathbf{y}) \mapsto \boldsymbol{\phi}(\mathbf{x}) \otimes \frac{\mathbf{y}}{||\mathbf{y}||}, \tag{4.5}$$

where $\mathcal{Y}$ is here the set of all possible indicator vectors corresponding to the taxonomic labels defined by a pre-established taxonomy $\{\mathcal{C}, \prec\}$, and $\boldsymbol{\phi}$ is a another feature map only concerned with transforming the data in $\mathcal{X}$, much like the feature map we use for SkNN.

Solving the inference problem

The inference argmax problems associated with our SkNN and SSVM hierarchical classification methods, given in Eq. 4.2 and Eq. 4.3, respectively, can be solved by exhaustively searching the set $\mathcal{Y}$ for the optimal taxonomic label. We adopt this strategy in our study, since it is time-efficient enough in classification problems that do not involve a very large number of classes. For problems with a very large number of classes, or when high time efficiency is a crucial matter, other strategies would however be better-suited for solving these inference argmax problems, e.g., strategies that resort to greedy approximations, but this aspect is out of the scope of the present study.

4.3 Real vision-based classification problems

4.3.1 Facial expression recognition

The problem

For this experiment, we define a facial expression as an observable pattern of facial muscle contraction. We use the facial action coding system (FACS, proposed by [36]), which gives a very detailed description of a facial expression in terms of action units (AUs). AUs represent the atomic movements that can be performed independently, though not always spontaneously, within the facial muscles. Each AU is associated with the action of one muscle, or one group of muscles. The FACS describes more than a hundred AUs, which can be noted with an optional intensity marker going from A (trace) to E (maximum). A valid FACS code can be for instance `[1+2+5+26]`, where we have in this case the presence of AU1 (inner eyebrow raiser), AU2 (outer eyebrow

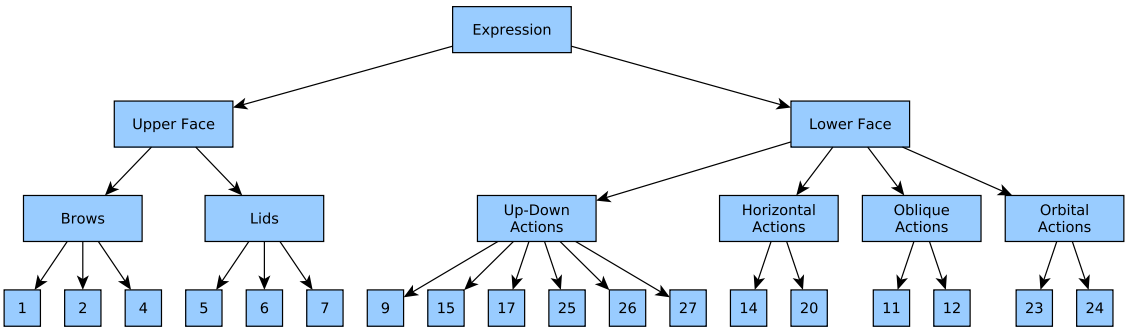


FIGURE 4.1: Our facial expression taxonomy. The leaves correspond to FACS action units.

raiser), AU5 (upper lid raiser), and AU26 (jaw drop). FACS codes can also be used to describe basic emotions conveyed by the facial expression. For instance, for most people with no particular mental condition, the FACS code `[1+2+5B+26]`, as described above but with the additional intensity marker B (slight) for AU5, can be confidently associated to the prototypical emotion of surprise, as illustrated by the rightmost face image in the bottom row of Fig. 4.2.

In order to use the hierarchical approach in our problem of facial expression recognition, we must specify a class taxonomy. FACS codes typically involve multiple AUs, each of which can be taxonomized according to the area of the face where the action takes place, and the type of local deformation the action applies on the face. We choose to focus here on only 18 specific AUs, and to ignore the possible intensity markers. These choices are respectively motivated by the fact that those 18 AUs are the most frequent ones within the facial expression dataset used in this experiment (see below, the CK+ dataset, proposed by [141]), and the fact that the intensity markers are not consistently provided within this dataset because they are optional. We therefore propose the class taxonomy in Fig. 4.1 for a hierarchical semantic description of the facial expression limited to our 18 AUs of interest. This tree taxonomy is inspired by how those AUs are usually grouped when presented in the literature [36]. As their names suggest, up-down actions, horizontal actions, and oblique actions group AUs for which the deformation movement in the frontal face is mostly vertical (e.g., AU26: jaw drop), horizontal (e.g., AU20: lip stretcher), or oblique (e.g., AU12 lip corner puller), respectively. Orbital actions group AUs for which the deformation is seemingly radial with respect to a fixed point, e.g., AU24: lip pressor, which closes the mouth and puckers the lips, seemingly bringing them closer to the centroid point of the mouth region.

The Extended Cohn-Kanade Dataset (CK+)

The CK+ dataset proposed by [141] consists of FACS-annotated video recordings of 123 subjects between the age of 18 to 50 years. 69% of these subjects are female, and 31% male. Also, 81% of these subjects are Euro-American, 13% Afro-American, and 6% other groups. Every subject was instructed to display several facial expressions from a predefined series of 23 different facial expressions. In total, 593 videos of 10 to 60 frames were recorded and manually annotated with a facial expression label in the form of a FACS code. All videos start with an onset neutral facial expression and end with the peak of the facial expression that the subject was asked to display. Figure 4.2 show some examples of peak facial expression from the CK+dataset. Additionally,

FIGURE 4.2: Examples of facial expressions present in the CK+ dataset.

landmark point annotations are provided for all frames of all videos: 68 fiducial points have been marked on the face, sketching the most salient parts of the face shape. For each video, these landmark points were obtained by (1) manually annotating a few key frames with the points, and then (2) applying an automatic face alignment method (a version of the active appearance models, proposed by [65]) to track the points in the remaining frames. Finally, the CK+ dataset comes with a standard evaluation strategy for comparing different facial expression recognition methods: the authors propose to use a leave-one-subject-out strategy, therefore resulting in a cross-validation with 123 folds partitioning the set of 593 videos.

We used the CK+ dataset in this experiment because it is one of the standard benchmark datasets for reporting the performance of methods designed for facial expression recognition (in a controlled environment, as opposed to "in the wild" benchmark datasets). The CK+ dataset is well-designed in its acquisition protocol: it is composed of clean video recordings, with many subjects and many different facial expressions. This dataset is also quite rich in the metadata it provides, including FACS codes for each video, 68 landmark points for each video frame, and even annotations about the prototypical emotions for some videos, not used in this experiment. Although as much as 30 AUs are present in the FACS annotations of the CK+ dataset, we only consider those AUs that are present in at least 30 occurrences within this dataset, totaling 18 AUs, so that our classifiers have enough training data to capture what is relevant to the recognition of those AUs. We emphasize that our goal in this experiment is not to propose a specific method that would outperform all current, state-of-the-art methods for facial expression recognition. Instead, our goal is to investigate how using the hierarchical approach compares to using the flat approach for the classification of facial expressions, by applying general-purpose hierarchical methods and their flat counterparts to a well-known facial expression recognition benchmark dataset.

Face features

In this experiment, we use face features that are close to the similarity normalized shape features (SPTS) and canonical normalized appearance features (CAPP) proposed in [141]. The extraction of these features is essentially based on the landmark point annotations provided for the video frames of the CK+ dataset.

Let $\mathbf{x}$ be a facial expression video from the set $\mathcal{X}$ of videos in the CK+ dataset. Our face features $\boldsymbol{\phi}(\mathbf{x})$ for this video consist of a 636-dimensional real-valued vector, i.e., $\boldsymbol{\phi}(\mathbf{x}) = [\mathbf{f}_s; \mathbf{f}_a]$, where $\mathbf{f}_s$ has 136 components and encodes information about the face shape, and where $\mathbf{f}_a$ has 500 components and encodes information about the face appearance (i.e, gray-level texture information). We wish to avoid mixing our facial expression recognition problem with an unwanted identity factor related to the static morphological differences between people. Therefore, the features $[\mathbf{f}_s; \mathbf{f}_a]$ are "identity normalized", by the apt subtraction of all information about the onset, neutral facial expression at the beginning of each video, from the information about the peak facial expression displayed in the end of the video. This strategy was also used in [141].

In more detail, our shape and appearance face features $\mathbf{f}_s$ and $\mathbf{f}_a$ are extracted as follows. First, we arbitrarily fix a reference face shape $\mathbf{s}_0$, composed of 68 landmark points in 2D, i.e., 136 real-valued coordinates. Likewise, we fix a reference grid $\mathcal{S}_0$ composed of 500 pixel locations which are organized to densely span the convex hull defined by the reference shape $\mathbf{s}_0$. Both $\mathbf{s}_0$ and $\mathcal{S}_0$ remain constant in the extraction of face features from any video of the CK+ dataset. Then, for extracting the face features specific to a video, we begin by retrieving the first and last video frames, $\mathbf{I}_n$ and $\mathbf{I}_p$, respectively, as well as their corresponding annotations of 68 landmark points, i.e., the face shapes $\mathbf{s}_n$ and $\mathbf{s}_p$, respectively. $\mathbf{I}_n$ and $\mathbf{s}_n$ thus contain information about the onset, neutral facial expression for the video, and $\mathbf{I}_p$ and $\mathbf{s}_p$ contain information about the peak facial expression for the video. Considering the landmark points as the control points of a geometric warping transformation, both of the $\{\mathbf{s}_0, \mathbf{s}_n\}$ and $\{\mathbf{s}_0, \mathbf{s}_p\}$ shape couples are used to define warping functions that project the set of pixel locations $\mathcal{S}_0$ to the pixel location domains of $\mathbf{I}_n$ and $\mathbf{I}_p$, respectively. Considering the images $\mathbf{I}_n$ and $\mathbf{I}_p$ as two functions taking a set of 2D pixel locations and returning their corresponding pixel values, the "shape-free" difference in facial appearance between the last and first frames of the video can then be calculated, as

$$\Delta\mathbf{a} = \mathbf{I}_p(\mathbf{W}(\mathcal{S}_0; \mathbf{s}_0, \mathbf{s}_p)) - \mathbf{I}_n(\mathbf{W}(\mathcal{S}_0; \mathbf{s}_0, \mathbf{s}_n)), \tag{4.6}$$

where $\mathbf{W}(\mathcal{S}; \mathbf{s}_1, \mathbf{s}_2)$ is a warping function defined by the shape couple $\{\mathbf{s}_1, \mathbf{s}_2\}$, which projects the pixel locations $\mathcal{S}$ from the domain of $\mathbf{s}_1$ into the domain of $\mathbf{s}_2$. Next, both the neutral $\mathbf{s}_n$ and peak $\mathbf{s}_p$ face shapes are aligned to the reference shape $\mathbf{s}_0$, according to the 2D similarity, i.e., adjusting their global scale, 2D rotation and 2D translation. The similarity normalized difference in the face shape between the last and first frames of the video is therefore calculated, as

$$\Delta\mathbf{s} = \mathbf{N}(\mathbf{s}_p, \mathbf{s}_0) - \mathbf{N}(\mathbf{s}_n, \mathbf{s}_0), \tag{4.7}$$

where $\mathbf{N}(\mathbf{s}_1, \mathbf{s}_2)$ is a transform that optimally aligns $\mathbf{s}_1$ to $\mathbf{s}_2$ according to the 2D similarity. The shape and appearance differences $\Delta\mathbf{s}$ and $\Delta\mathbf{a}$ are then independently normalized to have zero mean and unit length, giving $\Delta\tilde{\mathbf{s}}$ and $\Delta\tilde{\mathbf{a}}$, respectively. These normalized differences are then combined into a single vector $[\Delta\tilde{\mathbf{s}}; \Delta\tilde{\mathbf{a}}]$, which is further normalized in the same way, i.e., to have zero mean and unit length. This normalized combined vector corresponds to the face features $[\mathbf{f}_s; \mathbf{f}_a] = \boldsymbol{\phi}(\mathbf{x})$ for a video $\mathbf{x}$ of the CK+ dataset.

Results

Our two hierarchical classification methods, i.e., SkNN and SSVM, that deal with taxonomic labels, are compared to their flat counterparts that deal with non-taxonomic, leaf-level labels. These flat counterparts are the standard, multiclass, multilabel kNN, and a multiclass, multilabel version of the standard SVM (multiclass kernel-based SVM, or MKSVM, proposed by [142]), respectively. In our problem of facial expression recognition, a taxonomic label normally includes several paths reaching the leaf nodes in the class taxonomy defined in Fig. 4.1. This means that we use here multiple-path, full depth labeling for a taxonomic label. A non-taxonomic label for a facial expression normally includes several leaf nodes of the class taxonomy, but never its interior nodes.

For each of the hierarchical and flat methods evaluated here, we consider the variation of a core parameter, the tuning of which can have a large influence on the results. For SkNN and kNN, this parameter is the number of neighbors k that are considered during the testing stage. Considering more neighbors means considering more alternatives to assign the output label. Fixed parameters for kNN and SkNN are the distance measure ρ (which is the L^2 norm) and the weights given to the votes of the nearest neighbors (which is the inverse of the distance measure). For SSVM and MKSVM, the core parameter we consider is the training parameter C, which, in the soft-margin approach, balances the allowed misclassification rate during the training procedure. Large values of C makes the optimization favor smaller-margin hyperplanes (more training data instances get to be correctly classified). All other SSVM and MKSVM parameters are fixed, like, for SSVM, the joint feature map $\boldsymbol{\psi}$ (see Eq. 4.5), and the way to construct the surrogate loss (which is the margin rescaling method).

Figure 4.3 depicts the evaluation curves obtained by applying our hierarchical and flat classification methods to our facial expression recognition problem. The curves show the evolution of the hierarchical F-measure (*hF*) against the core parameters chosen for our methods, which parameters we make vary within an appropriate range of values. Because we want to compare the hierarchical classification approach to the flat classification approach, and not two hierarchical methods against each other, the performance curves are grouped accordingly in pairs of a hierarchical method and its flat counterpart, i.e., SkNN vs. kNN, and SSVM vs. MKSVM. It is noteworthy that the evaluation measure *hF* considers all leaf and interior nodes of the class taxonomy presented in Fig. 4.1, to characterize the performance of the hierarchical methods and the flat methods alike, even though the existence of the interior nodes is not considered at all in the formulation of the training and inference parts of the flat methods. In that way, we wish to determine whether there is a gain in performance given by the hierarchical methods with respect to the flat methods in our experiment. Such gain would necessarily come from the incorporation of the information about the full class taxonomy within our hierarchical classification methods.

We observe that the hierarchical approach does not outperform the flat approach with either of the hierarchical methods in our problem of facial expression recognition. Instead, the hierarchical and flat approaches have a very similar performance. A confirmation of this is given in Tab. 4.1, where for each hierarchical and flat method the best performance in hierarchical recall (*hR*), hierarchical precision *hP*, and hierarchical F-measure *hF* are detailed. These best performance values are associated with the highest points of the performance curves presented in Fig. 4.3. The overall

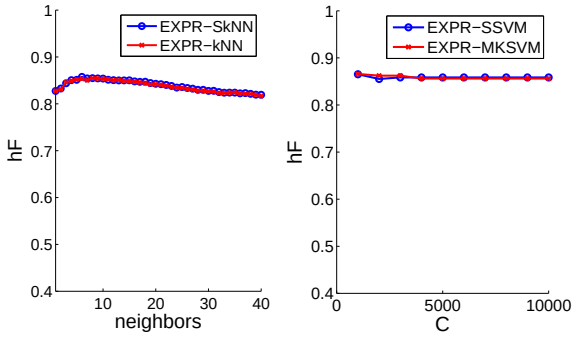


FIGURE 4.3: Results of facial expression recognition. Blue and red curves show *hF* for hierarchical and flat classification respectively, against the number of neighbors k for SkNN vs. kNN (left), and the training parameter C for SSVM vs. MKSVM (right).

TABLE 4.1: Best *hF* performance from Fig. 4.3, along with the corresponding *hP* and *hR* performance obtained in our facial expression recognition problem.

Classifier	hP	hR	hF
SkNN	**83.63%**	**88.00%**	**85.76%**
kNN	83.12%	87.98%	85.48%
SSVM	85.22%	**87.87%**	86.52%
MKSVM	**85.68%**	87.54%	**86.60%**

similarity of these performance results allows us to conclude that the hierarchical approach gives no gain in performance over the flat approach in this experiment.

The results reported here are illustrative. We actually designed multiple other experimental cases, where we changed various aspects pertaining to the evaluation of hierarchical vs. flat classification in our facial expression recognition problem, but still reached the same conclusion. As stated in the introduction of this chapter, however, the hierarchical approach to classification has been shown to improve the classification performance in several application domains, including domains involving the classification of visual content. We thus consider the possibility that our facial expression recognition problem might not be well-conditioned to showcase the superiority of the hierarchical approach. Therefore, we decide to give a round of experimentation to another vision-based classification problem that is more likely to benefit from the hierarchical approach, namely 3D shape recognition. Our experiment on this problem is detailed in the next section.

4.3.2 3D shape recognition

The problem

The problem of 3D shape recognition consists in determining the category of an object on the sole basis of its shape in the 3D world, without considering its appearance, e.g., its color or gray-level texture, or any other kind of perceptual evidence. Most of the time in the definition of this problem, the objects under consideration either have a rigid shape by nature, e.g., a car (bottom row of Fig. 4.5, second image from the right), or are given a prototypical rigid shape if they are deformable

by nature, e.g., a dog [in a normal standing posture] (bottom row of Fig. 4.5, rightmost image). Indeed, the possible dynamic shape deformations of an object are essentially disregarded in the problem of 3D shape recognition. Instead, the focus is given to effectively describing the static shape similarities of objects that belong to the same category, toward effectively distinguishing objects from different categories. The intuition behind 3D shape recognition is that an adequate shape description of an object in space provides the most important cues toward categorizing many of the objects one could encounter in the real world. Therefore, except for some object categorizations (e.g., a color-based categorization), it is assumed that an adequate description of the 3D shape carries all the necessary information relevant to the recognition of most real-world objects. It is also assumed that the raw visual data acquired by modern depth sensing devices, i.e., the data commonly called 3D point clouds, hold all of the information necessary to construct adequate 3D shape descriptions, much like it is assumed that conventional digital images acquired by modern cameras hold all of the information necessary to construct adequate image features for solving recognition problems based on 2D images.

We think that the problem of 3D shape recognition is well-suited to continue our investigation of hierarchical vs. flat classification of visual content. Indeed, real-world objects can be quite naturally organized in class taxonomies according to an "IS-A" partial order relationship that notably embodies shared semantic attributes about the shape, e.g., a dog and a cow are both quadrupeds, and a human and an ostrich are both bipeds, and both quadrupeds and bipeds are animals with legs. It is intuitive that incorporating an adequate object class taxonomy within the problem of 3D shape recognition should yield a classification performance gain. In this second experiment, we therefore provide our hierarchical methods with prior taxonomic information about 3D objects. Both our hierarchical and flat methods use the same features, in the form of shape descriptions extracted from the 3D point clouds of the objects, but only our hierarchical methods learn to infer the full taxonomic label of a 3D object, as a single path to a leaf node in a tree-based object class taxonomy (Fig. 4.4). Some parts of this second experiment are the work of our former colleagues from the Intelligent and Interactive Systems group at the university of Innsbruck, namely, the very idea to use hierarchical classification in the problem of 3D shape recognition, the choice of the taxonomic dataset of 3D objects for this problem, the generation of the 3D point clouds from the CAD models in this dataset, and the choice and application of the methods used to extract shape descriptions from these point clouds.

The Princeton Shape Benchmark (PSB)

The PSB dataset proposed by [138] is one of the largest and most heterogeneous datasets of 3D objects, and also one of the most challenging for 3D shape recognition [143]. It is composed of 1,814 CAD models which give the surfaces of a wide variety of natural and man-made 3D objects. These objects are categorized into 161 object classes, as the leaf nodes of a tree-based class taxonomy with 36 interior nodes organized in up to four semantic levels above the leaf level (Fig. 4.4). The taxonomy proposed by the authors of the PSB dataset was constructed by considering (1) shared shape-related attributes, e.g., round tables belong to the same category, because they are round, and (2) shared semantic attributes, e.g., "Furniture" is a superclass of both "Table" and "Bed", because these objects share a property about how humans use them.

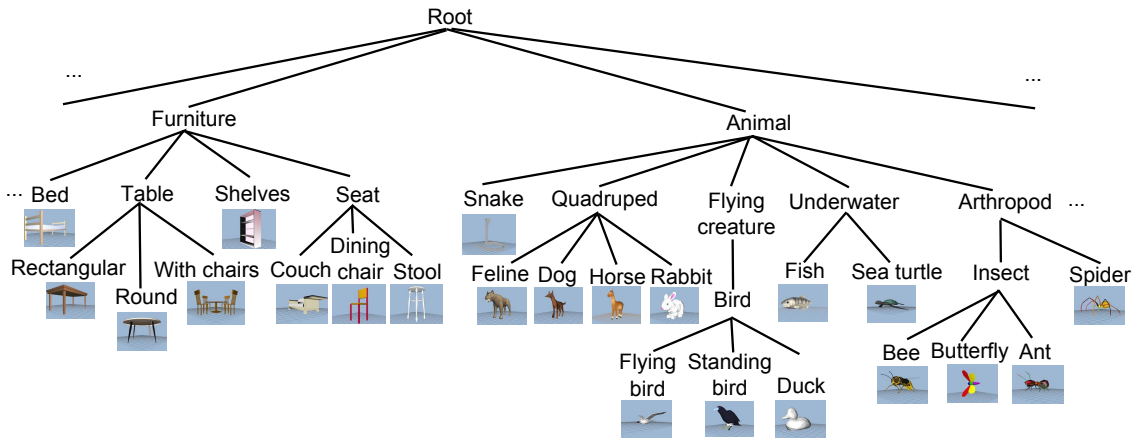


FIGURE 4.4: The "Furniture" and "Animal" sub-trees of the Princeton Shape Benchmark, with snapshots of some of the models that belong to the leaves (classes) of those sub-trees.

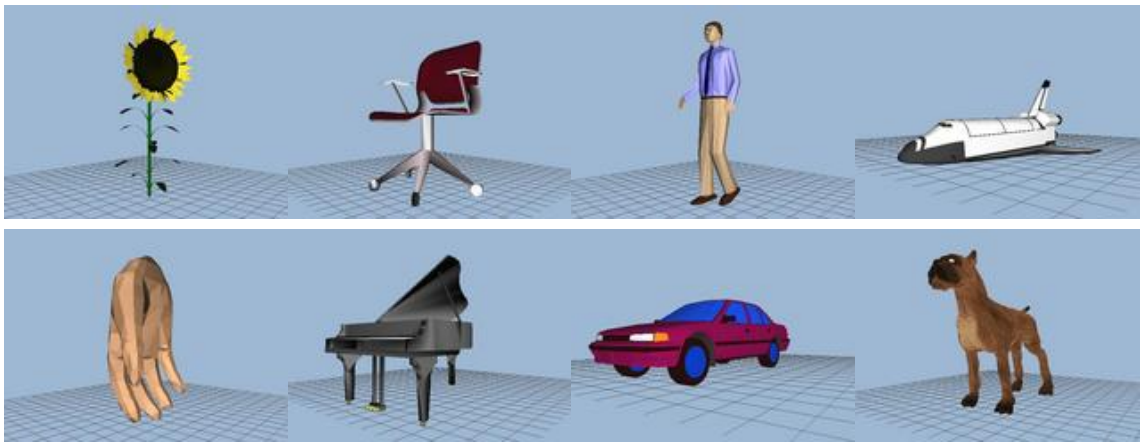


FIGURE 4.5: Examples of 3D object models present in the Princeton Shape Benchmark.

The CAD models encode the polygonal geometry of the objects they describe, as a set of vertices and edges. Textured depictions of some of these CAD models are shown in Fig. 4.5 (colors and shading are here added for visual comfort, but are not parts of the CAD models). Even though the CAD models proposed in the PSB dataset are obviously not raw sensing data acquired from the real world via depth sensing devices (e.g., a laser scanner), they represent good approximations to their real-world equivalents. Such models are actually widely used to present new solutions to the problem of 3D shape recognition, as the performance of such solutions in real-world scenarios can be reliably extrapolated from their performance on CAD models. Finally, the PSB dataset comes with an evaluation strategy proposed by the authors to perform the benchmark of 3D shape recognition methods. The set of 1,814 CAD models is divided into two splits, one for training the classification methods, the other for testing the obtained classifiers. This dataset subdivision was carefully designed to best represent all object categories in both the training and test splits, with an appropriate number of same-category CAD models in both splits.

3D shape features

We mentioned above that adequate 3D shape descriptions should allow to effectively recognize the category of many real-world objects. Standard methods to build such descriptions exist, and are commonly called 3D shape descriptors. In 3D computer vision research, the design and development of 3D shape descriptors is a very active and prolific line of work. The common objective is to develop a general-purpose method that captures what is essential to describe a 3D point cloud, i.e., a method that gives, for any point cloud, a geometric description that is resistant to noise, compact against redundancy, and invariant to specific similarity transformations. However, the effectiveness of the various 3D shape descriptors proposed in the literature was most often demonstrated within the specific context of experimentation in which they were designed. There is therefore no guarantee that such 3D shape descriptors give adequate shape descriptions for our experiment of hierarchical vs. flat classification in the problem of 3D shape recognition on the PSB dataset.

Of the many 3D shape descriptors available, the following five were chosen for this experiment: ensemble of shape functions (ESF [144]), viewpoint feature histogram (VFH [145]), intrinsic spin images (ISI [146]), signature of histograms of orientations (SHOT [147]), and unique shape contexts (USC [148]). We do not, in this document, describe the detail of how these methods work, but invite the reader to follow our bibliography pointers to find out more about them. The most important aspects to remember here are the reasons why these particular 3D shape descriptors were chosen: (1) they are quite different in their design, and should give quite different 3D shape descriptions to use as features in our experiment, and (2) there is an easy access to their straightforward implementation. Indeed, each of these five 3D shape descriptors are available in the widely used and well-documented point cloud library (PCL) [149]. By applying our hierarchical and flat classification methods to features coming from five different 3D shape descriptors implemented in the sound and consistent PCL framework, we wish to multiply the variants of this experiment, and therefore enhance the quality of our comparative study about the hierarchical and flat approaches for the classification of visual content.

Since the PCL framework expects a 3D point cloud as input for any of our 3D shape descriptors of interest (ESF, VFH, ISI, SHOT, and USC), each CAD object model of the PSB dataset was transformed into a point cloud. To do so, the surface of each CAD model was first triangulated using its vertices and edges, then a total of 5,000 3D points were randomly sampled from its full triangulated 3D surface. The probability to sample a point from a particular triangle of the surface was made proportional to the area of this triangle. In this way, the 5,000 3D points obtained from each CAD model were homogeneously distributed across its 3D surface. Given such a point cloud, our five 3D shape descriptors of interest calculate shape descriptions of different lengths, some with only a few dimensions, other very high-dimensional. To facilitate the design of our experimental variants with these five descriptors of interest, five linear kernel matrices were calculated, each encoding the PSB object point cloud similarities according to the shape descriptions calculated with one of the five particular descriptors. For any experimental variant involving a particular descriptor, we used the rows of the corresponding kernel matrix as 3D shape features with which to feed our classification methods. More formally, the feature map $\boldsymbol{\phi}$ used in an experimental variant of our problem of 3D shape recognition is as follows. If $\mathbf{x}$ is a CAD model-generated point cloud from the (orderded) set $\mathcal{X}$ of all PSB CAD model-generated point

clouds, then $\boldsymbol{\phi}(\mathbf{x})$ is the real-valued vector of the dot products between the shape description of $\mathbf{x}$ and the shape descriptions of all the elements in $\mathcal{X}$, such shape descriptions coming from the application of the same particular shape descriptor to the elements in $\mathcal{X}$.

Results

Using our five 3D shape descriptors of interest, i.e., ESF, VFH, ISI, SHOT, and USC, we conduct five comparative evaluations of the hierarchical and flat classification approaches for solving the problem of 3D shape recognition defined over the PSB dataset. Each comparative evaluation involves the results given by our two hierarchical classification methods of interest, i.e., SkNN and SSVM for the prediction of taxonomic labels, as well as the results given by their flat counterparts dealing with non-taxonomic labels, i.e., standard multiclass kNN, and a version of the standard multiclass SVM called MKSVM [142]. In our problem of 3D shape recognition, a taxonomic label corresponds to a path from the root to a leaf node within the PSB tree taxonomy (an excerpt from which is shown in Fig. 4.4), i.e., we use single-path, full depth labeling. In the flat classification approach, a non-taxonomic label here corresponds to a scalar class label, i.e., it corresponds to one of the leaf nodes of the PSB tree taxonomy (the interior nodes of the tree-based class taxonomy are therefore not considered).

Figure 4.6 shows our results for the problem of 3D shape recognition. As in our experiment about facial expression recognition, we plot here the hierarchical F-measure (*hF*) against the core parameters for our hierarchical and flat classification methods, i.e., the number of neighbors k for SkNN and kNN, and the C parameter for SSVM and MKSVM. We also group the evaluation curves in pairs according to (1) the 3D shape descriptor used to obtain our 3D shape features, i.e., either of ESF, VFH, ISI, SHOT, or USC, and (2) the interest (and fairness) of comparing the performance of a hierarchical classification method specifically with those of its respective flat counterpart, i.e., we compare SkNN vs. kNN, and SSVM vs. MKSVM. We also wish to remind the reader that, regardless of whether it is a hierarchical or a flat method that is used in the testing stage, the evaluation measure *hF* takes into account all leaf and interior nodes of the PSB tree-based class taxonomy. Indeed, even though the prediction made by a flat method for some test data instance is a scalar class label, this class label corresponds to a leaf node within the PSB class taxonomy and implicitly defines a unique path that includes all its ancestor nodes up to the root. In that sense, a flat method could be seen as being able to predict taxonomic labels, but what is of interest to us is that our flat methods are oblivious to the full class taxonomy in both their training and testing stages, whereas our hierarchical methods are given prior information about the full class taxonomy, which should intuitively help them achieve a better classification performance.

By visual inspection of the curves in Fig. 4.6, there seems to be, with some of our descriptors of interest, a consistent yet rather slight performance improvement by using the hierarchical approach in our problem of 3D shape recognition. Indeed, when based on the VFH, ESF, and ISI descriptors, the classification seems to benefit a little from the incorporation of prior hierarchical information. This is further illustrated in Table 4.2 which gives the detail about the best *hF*, *hR*, and *hP* values obtained for each of our experimental variants. However, for the classification based on the SHOT and USC descriptors, the results are mixed: either the hierarchical approach or the flat approach performs slightly better, depending on the descriptor and method that is used. We

TABLE 4.2: Best *hF* performance from Fig. 4.6, along with the corresponding *hP* and *hR* performance obtained in our 3D shape recognition problem using the 3D shape descriptors ESF, VFH, ISI, SHOT, and USC.

Descriptor	Classifier	hP	hR	hF
ESF	SkNN	**32.23%**	**34.40%**	**33.28%**
	kNN	32.00%	34.22%	33.07%
	SSVM	**49.72%**	**49.92%**	**49.82%**
	MKSVM	47.78%	47.45%	47.61%
VFH	SkNN	**20.38%**	**23.07%**	**21.64%**
	kNN	19.60%	21.42%	20.47%
	SSVM	**23.47%**	**23.62%**	**23.55%**
	MKSVM	21.84%	21.84%	21.84%
ISI	SkNN	**27.24%**	**29.07%**	**28.12%**
	kNN	26.42%	27.79%	27.09%
	SSVM	**31.15%**	**33.58%**	**32.32%**
	MKSVM	31.01%	32.23%	31.61%
SHOT	SkNN	**34.36%**	34.95%	34.65%
	kNN	33.99%	**35.48%**	**34.72%**
	SSVM	33.43%	36.35%	34.83%
	MKSVM	**35.79%**	**36.67%**	**36.22%**
USC	SkNN	40.26%	41.08%	40.67%
	kNN	**40.78%**	**41.18%**	**40.98%**
	SSVM	**37.58%**	**40.88%**	**39.16%**
	MKSVM	37.56%	39.41%	38.46%

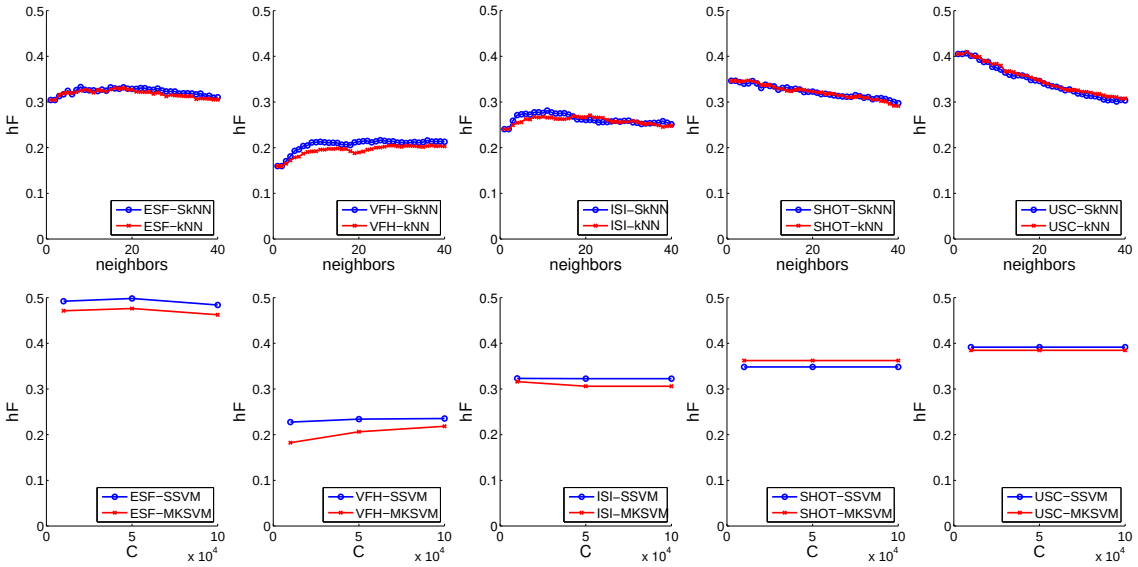


FIGURE 4.6: Results of 3D shape recognition. Blue and red curves show *hF* for hierarchical and flat classification respectively, against the number of neighbors k for SkNN vs. kNN in the first row, and the training parameter C for SSVM vs. MKSVM in the second row. Each column corresponds to the use of a particular 3D shape descriptor, between ESF, VFH, ISI, SHOT, and USC.

therefore conclude that the hierarchical classification approach does not give better results than the flat classification approach in solving our 3D shape recognition problem defined over the PSB dataset, with either of the five 3D shape descriptors, but for a few cases where this improvement is hardly significant. It is important to note that the absolute classification performance obtained in using this or that 3D shape descriptor is of little interest to us, because in this experiment we do not aim at determining which 3D shape descriptor is best designed in general, or the most useful for solving the problem of 3D shape recognition over the PSB dataset in particular. Instead, we are interested in the relative gain in classification performance that the hierarchical approach may offer in comparison to the flat approach. The only interest for us in considering the absolute classification performance is to empirically validate the numerous experimental variants we designed for our problem of 3D shape recognition, this absolute performance being well above the chance level in all variants, considering that we deal with a classification problem with 161 classes.

4.4 Simulation framework

The experimental results presented in Sect. 4.3 for our real-world vision-based problems leave us with an uneasy, and actually unexpected conclusion about the hierarchical classification approach. After systematically applying two different hierarchical classification methods and their flat counterparts to two different vision-based classification problems, and using sound feature extraction methods of noticeably different natures in that experimentation, we still fail to showcase the superiority of the hierarchical approach over the flat approach for solving classification problems based on visual features. However, as stated in Sect. 4.1, such superiority (1) has been clearly demonstrated in general in other research fields, such as text categorization and protein function

prediction, and (2) can be reasonably expected as the use of a hierarchical prior for classifying visual content echoes the findings made in neurophysiological studies of the visual cortex, which emphasize the natural hierarchical organization made by humans to recognize the objects they see.

In order to decide what conclusions can be drawn from the the present study, we make the hypothesis that the features extracted from the raw visual data for solving our real-world vision-based classification problems lack the necessary information to exploit a hierarchical prior. Our argument is that the feature extraction methods we use in this study are commonly used in 2D and 3D computer vision for general purpose, and have not been specifically designed with the idea to be jointly used with a hierarchical prior. We have the intuition that for a hierarchical prior to be useful, the features should somehow be able to capture hierarchical information so that it may be matched to the hierarchical prior by sound hierarchical classification methods. Based on this hypothesis, we ask the following three questions about the features we use, in the form of guesses on why these features might prevent the hierarchical methods from outperforming the flat methods in our real-world problem scenarios:

1. Do the features fail to capture *any* hierarchical information?
2. Do the features capture hierarchical information, but *different* from the hierarchical prior?
3. Do the features capture hierarchical information similar to the hierarchical prior, but with a destructive amount of *noise*?

In order to answer these rather general questions, we believe that it is a good strategy to not focus on a specific vision-based classification problem but instead consider a more general approach. To do so, we design a simulation framework that generates abstract, artificial classification problems, the complexity of which can be controlled through the manipulation of aspects that we think are key to consider in the hierarchical classification approach. From the results obtained with our hierarchical and flat classification methods applied to these artificial problems, we wish to draw useful insights about the conditions in which the hierarchical approach can offer a real gain in performance over the flat approach, for classification problems in general, and vision-based classification problems in particular.

4.4.1 Abstraction of the classification problem

To build a meaningful simulation framework, we need to have a clear view and definition of the concepts at work in the hierarchical and flat classification approaches (Fig. 4.7). Given some *phenomenon* of interest, the repeated *manifestation* of this phenomenon is measured by a *sensor* on the one hand, and a semantic *interpretation* of the possible states of the phenomenon is made by an observer on the other hand. We are interested in phenomena that have a natural hierarchical relationship between their states, i.e., an *underlying taxonomy*[2]. Being aware to some extent of the hierarchical nature of the phenomenon, the observer may organize his/her semantic interpretation in a hierarchical manner, i.e., define a *perceived taxonomy*. This perceived taxonomy does however not necessarily correspond perfectly to the natural underlying taxonomy.

[2]It is arguable whether or not there exists such a thing as an "underlying taxonomy" for a phenomenon. Taxonomies may be thought of as always being arbitrary, their value lying in their usefulness, not in some underlying, self-evident truth.

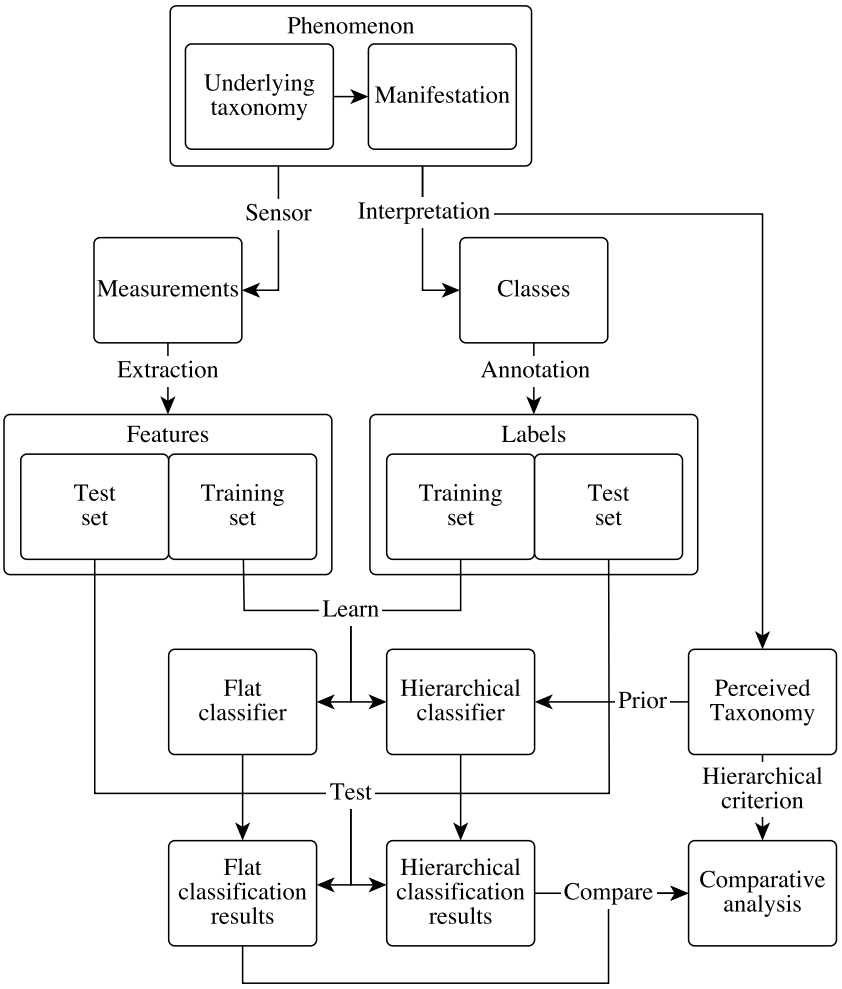


FIGURE 4.7: Our schematic view of the hierarchical and flat classification approaches used in our simulation framework.

The semantic *classes* provided by the interpretation of the observer are then used to associate *labels* to a collection of *measurements* obtained by the sensor. This *annotation* procedure yields a labeled dataset. In parallel, *features* are associated to the measurements via a feature *extraction* method, with the primary goal of capturing the essential information present in the measurements, in preparation of a task of supervised classification. If it is assumed that the measurements contain information about the underlying taxonomy of the phenomenon[3], then it is also assumed that a *high-level* feature extraction should capture at least part of this essential information, and therefore provide a rich, high-level feature representation, whereas a *low-level* feature extraction should fail to capture any part of the underlying taxonomic information.

A set of labeled features is therefore available for training a classifier in a supervised manner, i.e., for *learning* a classification rule that can associate class labels with new measurements of the same phenomenon, on the basis of their features obtained with the same feature extraction method as the one used to obtain the labeled features. In order to *test* the generalization capability of a classifier, the set of labeled features is split into a *training set* and a *test set*. The training of a *hierarchical classifier* differs from the training of a *flat classifier* by the fact that the method used for training the hierarchical classifier is given the perceived taxonomy as a *prior*, whereas the method used for training the flat classifier does not make use of a hierarchical prior about the classes.

Given the *hierarchical classification results* and the *flat classification results* obtained by testing the hierarchical and flat classifiers, respectively, we want to *compare* these results. Specifically, we want to make a fair *comparative analysis* of these results in order to get a good notion about which one of the hierarchical or flat approaches to classification performs best on our test set. Whether or not of a hierarchical prior is used in learning, all other things remaining equal, the classification performance of the hierarchical and flat classifiers can be fairly compared on the basis of the perceived taxonomy which, in this case, is used as a *hierarchical criterion* for penalizing the misclassification of the elements of the test set (e.g., using hierarchical evaluation measures, see Sect. 4.2.1). Indeed, regardless of the type of classifier being used, the superclass labels in the taxonomy used as a criterion come as a byproduct of the specific class labels that are predicted by the classifier, and these superclass labels can be used to emphasize serious hierarchical errors (according to the given hierarchical criterion) made by either a hierarchical or a flat classifier.

4.4.2 Artificial datasets with taxonomies

Following on what has been discussed in Sect. 4.4.1, we are interested in simulating the existence of labeled datasets obtained by measuring and interpreting the manifestations of phenomena having underlying taxonomies, which we call hierarchical phenomena for short. To simulate such underlying taxonomies, we consider perfect k-ary trees, where all leaf nodes belong to the same semantic level L (the root node belonging to the level 1), and all interior nodes have degree k, i.e., have k children. For such trees, the total number of nodes is given by $\frac{k^L-1}{k-1}$, and the number of leaf nodes is k^{L-1}. In our view, a path from the root to a leaf node of such a taxonomy corresponds to a state of the hierarchical phenomenon that is being measured by the sensor and interpreted

[3]The measurements may comply particularly well to a specific taxonomic model, that would be the best, i.e., the most useful taxonomic approximation of the nature of the phenomenon.

TABLE 4.3: The seven k-ary tree-based underlying taxonomies of the phenomena under consideration in our simulation experiment.

Tree type	k (children per node)	L (depth)	# nodes (total)	# leaves (classes)
Binary trees	2	3	7	4
	2	4	15	8
	2	5	31	16
	2	6	63	32
	2	7	127	64
Ternary trees	3	3	13	9
	3	4	40	27
	3	5	121	81
Quadtrees	4	3	21	16
	4	4	85	64

by the observer. Note that we do not consider here hierarchical phenomena for which a single manifestation can simultaneously correspond to multiple paths in the underlying tree taxonomy. We consider 10 different phenomena with underlying taxonomies in the form of k-ary trees (see Table 4.3). For each hierarchical phenomenon, we assume that (1) the observer is able to establish the existence of all the different states, and make those states correspond to semantic classes of his/her interpretation, and that (2) exactly 200 manifestations per state are measured by the sensor and correctly class-labeled by the observer. We therefore simulate the existence of 10 different labeled datasets which are perfectly class-balanced, and which are obtained from measuring and interpreting the states of 10 different phenomena having underlying taxonomies. Although only underlying taxonomies and class labels were actually generated so far, and not the data corresponding to the measurements, which remain abstract, we refer to these artificial labeled datasets as generated datasets in the following, for simplicity.

In our view, the observer has also established a perceived taxonomy embodying his/her interpretation of the hierarchical relationships between the semantic classes. For each of the 10 generated datasets, we consider a first experimental simulation condition where the perceived taxonomy perfectly matches the actual underlying taxonomy of the phenomenon associated with the generated dataset. We then consider a second experimental simulation condition where the perceived taxonomy does not perfectly match the underlying taxonomy, with varying degrees of disparity. The artificial classification problems resulting from the application of this second simulation condition mimic real classification problems where the arbitrarily chosen (perceived) taxonomies do not optimally reflect the hierarchical nature (underlying taxonomy) of the phenomenon under consideration. In order to test this second simulation condition, we focus on just one of our 10 generated datasets, i.e., the dataset associated with the underlying binary tree taxonomy having 7 levels (127 nodes, 64 leaves).

4.4.3 Artificial high-level features

We assume that the abstract measurements in our generated datasets somehow reliably encode the underlying hierarchical nature of the phenomenon of interest, which we believe to be a reasonable hypothesis that applies in most practical cases (e.g., digital images, CAD model-based point clouds, etc.). We also assume that the ideal features extracted from these measurements should decode and capture the hierarchical nature of the phenomenon, i.e., in our terminology they should be high-level features. We therefore focus on simulating the varying quality of the high-level features extracted from the measurements, by considering these features as random variables with a Gaussian distribution, centered on their ideal value and with a variance indicative of the "noise degree" caused by the imperfect feature extraction procedure. For each generated dataset, we simulate the extraction of high-level features with different degrees of noise. More precisely, the choice of a generated dataset and the specification of a noise degree for the high-level features yields a specific couple of training and testing stages for the artificial classification problem defined over the generated dataset. These training and testing stages use noisy high-level features and the class labels in the dataset, as well as the perceived taxonomy as a prior, in the case where a hierarchical classifier is to be trained.

More formally, let $\mathcal{X}$ be the collection of (abstract) measurements in a dataset generated by measuring and annotating the manifestations of a phenomenon having an underlying taxonomy. Let $\mathcal{Y}^* \subset \{0,1\}^n$ be the set of indicator vectors, defined over this underlying taxonomy, that represent all of the possible states of the hierarchical phenomenon of interest. The simulated extraction of a noisy feature vector $\boldsymbol{\phi}(\mathbf{x}) \in \mathbb{R}^n$ from a measurement $\mathbf{x} \in \mathcal{X}$ is made on the basis of the state $\mathbf{y}^* \in \mathcal{Y}^*$ associated to $\mathbf{x}$. Indeed, in our view, $\mathbf{y}^*$ is the ideal representation of a state of the phenomenon, therefore $\mathbf{y}^*$ also corresponds to the best possible high-level features $\boldsymbol{\phi}(\mathbf{x})$ that can be extracted from a measurement $\mathbf{x}$ of the phenomenon in this state. Specifically, the high-level feature vector $\boldsymbol{\phi}(\mathbf{x})$ for a measurement $\mathbf{x} \in \mathcal{X}$ with a state $\mathbf{y}^* \in \mathcal{Y}^*$ is so that

$$\phi_i(\mathbf{x}) \sim \mathcal{N}(y_i^*, \sigma^2), \quad \forall i \in \{1, \ldots, n\}, \tag{4.8}$$

where the variance σ^2 of the Gaussian distribution represents the noise degree, shared by all features in $\boldsymbol{\phi}(\mathbf{x})$. For each of our 10 generated datasets (Sect. 4.4.2), we consider 51 progressive degrees of noise (and therefore 51 experimental cases for our artificial classification problems), by choosing σ^2 in $\{0, 0.05, \ldots, 2.5\}$.

With our method to simulate the feature extraction procedure, each scalar feature $\phi_i(\mathbf{x})$ is a random variable related to one of the n nodes of the underlying taxonomy (see Fig. 4.8, left and center). Such features are therefore high-level and capture hierarchical information with some amount of noise. In our first simulation condition, where the perceived taxonomy is equivalent to the underlying taxonomy, i.e., where a taxonomic label $\mathbf{y} \in \mathcal{Y}$ is equal to $\mathbf{y}^* \in \mathcal{Y}^*$, these features are very discriminative for all classes and superclasses of the hierarchical classification problem, disregarding the chosen noise degree. However in our second simulation condition, where the perceived and underlying taxonomies differ, these features may be less discriminative for the superclasses of the perceived taxonomy, since these superclasses do not ideally embody the true hierarchical nature of the phenomenon (see Fig. 4.8, right).

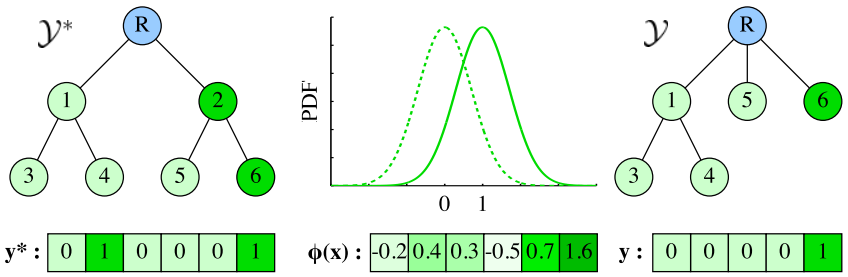


FIGURE 4.8: Left: an underlying taxonomy $\mathcal{Y}^*$ and a representation $\mathbf{y}^*$ of a state in this taxonomy. Center: a feature vector $\boldsymbol{\phi}(\mathbf{x})$ for a measurement $\mathbf{x}$, generated from its associated state $\mathbf{y}^*$ with a noise degree $\sigma^2 = 0.5$. Right: a label $\mathbf{y}$ for the measurement $\mathbf{x}$, defined over a perceived taxonomy $\mathcal{Y}$ obtained from $\mathcal{Y}^*$ by the elimination of the interior node 2.

4.4.4 Results

We conduct multiple evaluations of hierarchical vs. flat classification performance in our artificial classification problems. For each artificial classification problem, we split its generated dataset into a training set and a test set of the same size, i.e., 100 examples per class are attributed for both the training and test sets in each problem. In our first simulation condition, where the perceived taxonomy is equivalent to the underlying taxonomy, each comparative evaluation involves the results obtained with our hierarchical classification methods, i.e., SkNN and SSVM for the prediction of taxonomic labels, as well as the results obtained with their flat counterparts dealing with non-taxonomic labels, i.e., kNN and MKSVM. In our second simulation condition, however, where the perceived taxonomy differs from the underlying taxonomy, we only focus on comparing the performance of SkNN vs. kNN in the problem related to the phenomenon with the underlying 7-level binary tree taxonomy. Indeed, we do not want to overwhelm the reader with excessive experimentation, and, in anticipation of our analysis of the results, this single experimental case for the second simulation condition is insightful enough to allow us to draw useful conclusions in this study. For the same reasons, and unlike in our real vision-based problems, we do not consider a range of values for the core parameters of our classification methods in this simulation experiment, but instead fix those parameters: the number of neighbors $k := 10$ for SkNN and kNN, and the training parameter $C := 100$ for SSVM and MKSVM. Those values are empirically found to be near-optimal for our methods when used to solve our artificial classification problems. We consider of greater interest to show the evolution of the classification performance with respect to the varying degree of noise σ^2 chosen to simulate the quality of the features. Besides, it is noteworthy that our artificial classification problems are very similar by nature, but for some variations in their associated underlying taxonomies. They can therefore be effectively compared as the variants of a single meta-problem.

Figures 4.9 and 4.10 show the results obtained for all our experimental cases in the first simulation condition, for SkNN vs. kNN, and SSVM vs. MKSVM, respectively. In this simulation condition, where the underlying taxonomy is perfectly perceived by the observer and used as a prior for training hierarchical classifiers, we can see that the hierarchical approach to classification outperforms the flat approach in all experimental cases where the noise degree σ^2 is non-zero. The performance gain is even more pronounced when the number of classes is larger, i.e., when deeper

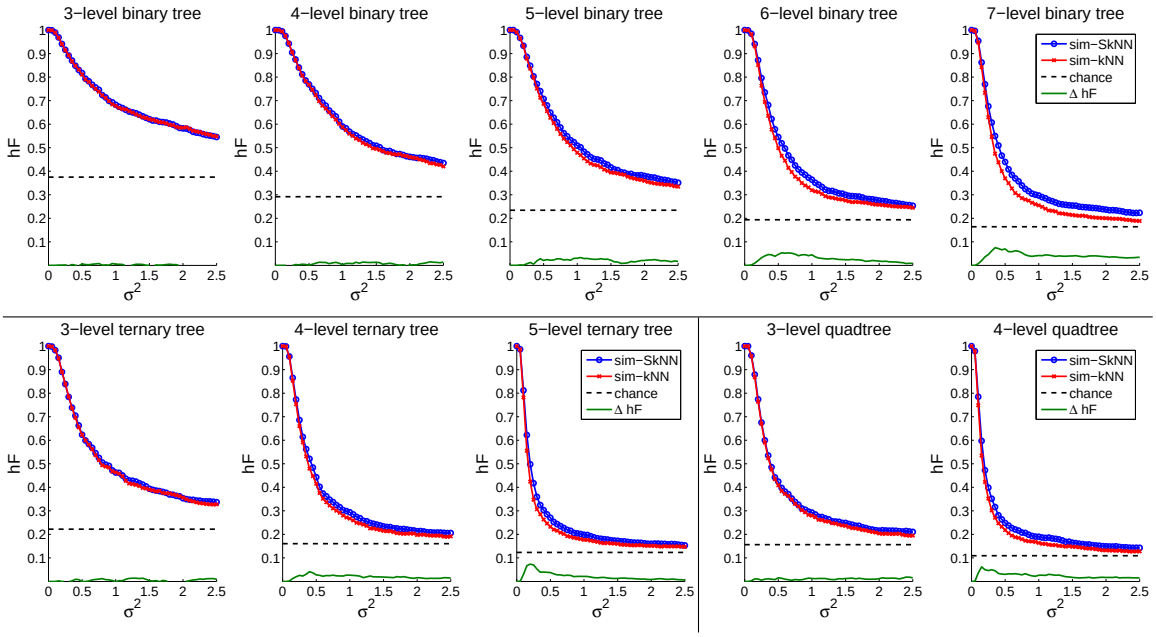


FIGURE 4.9: SkNN vs. kNN performance in the first simulation condition, using binary (top row) and ternary/quad trees (bottom row) for the taxonomies. Blue and red curves show *hF* for the hierarchical and flat classification, respectively, against the degree of noise in the features. Dashed black lines show the chance level for *hF*. Green curves show ΔhF, i.e., the performance gain in *hF* by using the hierarchical approach.

trees or larger tree degrees are used, with up to 13.31% hF gain for the 7-level binary tree taxonomy (64 classes), 11.99% hF gain for the 5-level ternary tree taxonomy (81 classes), and 11.56% hF gain for the 4-level quadtree taxonomy (64 classes). Table 4.4 gives quantitative results for all of our experimental cases in the first simulation condition. It can be noticed that the median hierarchical gain over the range of noise degrees globally increases with the number of taxonomy levels. From these results, we conclude that, when high-level features are used, i.e., rich feature representations that can capture the true hierarchical nature of the phenomenon, the hierarchical approach to classification outperforms the flat approach, even in the presence of strong noise in the high-level features.

In our second simulation condition, the true underlying taxonomy of the phenomenon is not identical to the perceived taxonomy that is used as a prior for training the hierarchical classifiers and as a hierarchical criterion for calculating the hierarchical F-measure. We simulate two perceptual errors that the observer could make in defining a perceived taxonomy for organizing the problem classes he/she has interpreted for the states of the phenomenon: (1) ignoring or missing some of the true hierarchical relationships between the states of the phenomenon, and (2) creating hierarchical relationships that do not exist between the states of the phenomenon. In practice, defining a perceived taxonomy containing the first, resp. second, error type corresponds to removing, resp. swapping, some of the interior nodes in the true underlying taxonomy of the phenomenon. We believe that both these perceptual error types typically coexist in practice, but we decide to simulate them independently in this experiment.

Figures 4.11 and 4.12 show the results obtained for SkNN vs. kNN using progressive interior

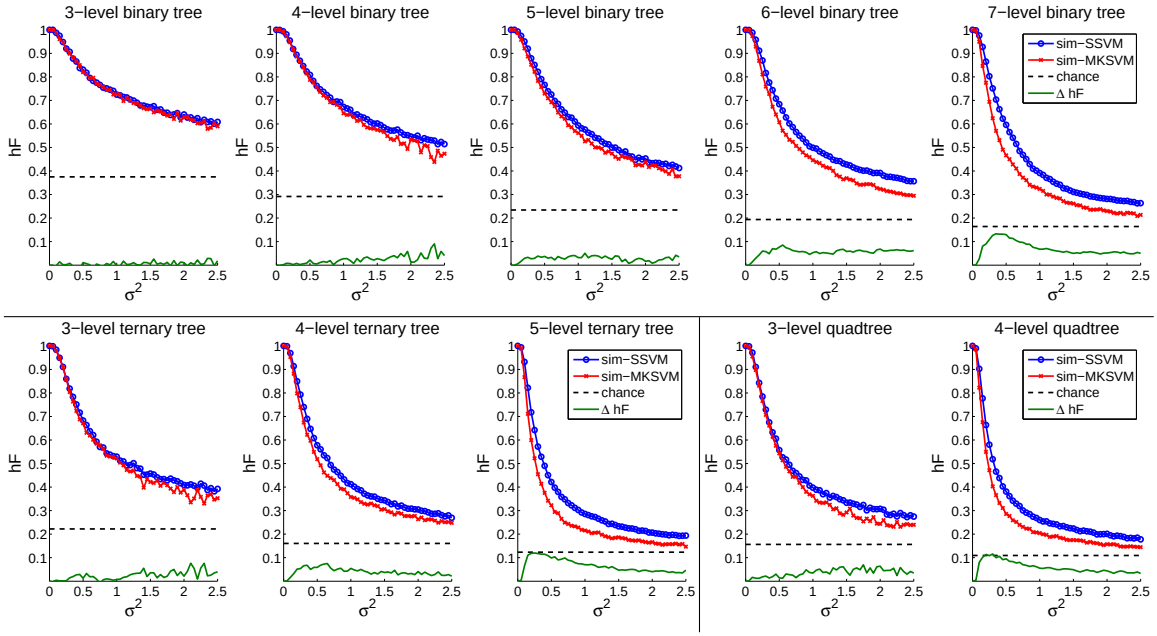


FIGURE 4.10: SSVM vs. MKSVM performance in the first simulation condition, using binary (top row) and ternary/quad trees (bottom row) for the taxonomies. Blue and red curves show *hF* for the hierarchical and flat classification, respectively, against the degree of noise in the features. Dashed black lines show the chance level for *hF*. Green curves show ΔhF, i.e., the performance gain in *hF* by using the hierarchical approach.

TABLE 4.4: Median and maximal ΔhF, i.e. performance gains in *hF* with the hierarchical approach, in our results for the first simulation condition shown in Fig. 4.9 and 4.10, for SkNN vs. kNN, and SSVM vs. MKSVM respectively.

		SkNN vs. kNN		SSVM vs. MKSVM	
Tree type	L	Med(ΔhF)	Max(ΔhF)	Med(ΔhF)	Max(ΔhF)
Binary trees	3	0.10%	0.80%	0.40%	2.90%
	4	0.73%	1.49%	2.15%	9.06%
	5	2.10%	3.32%	2.98%	5.03%
	6	2.48%	5.35%	5.92%	8.54%
	7	3.97%	7.52%	5.70%	13.31%
Ternary trees	3	0.35%	1.39%	2.08%	7.64%
	4	1.95%	4.16%	3.93%	7.50%
	5	1.48%	7.36%	5.86%	11.99%
Quadtrees	3	1.12%	1.86%	3.52%	6.98%
	4	2.21%	6.22%	5.08%	11.56%

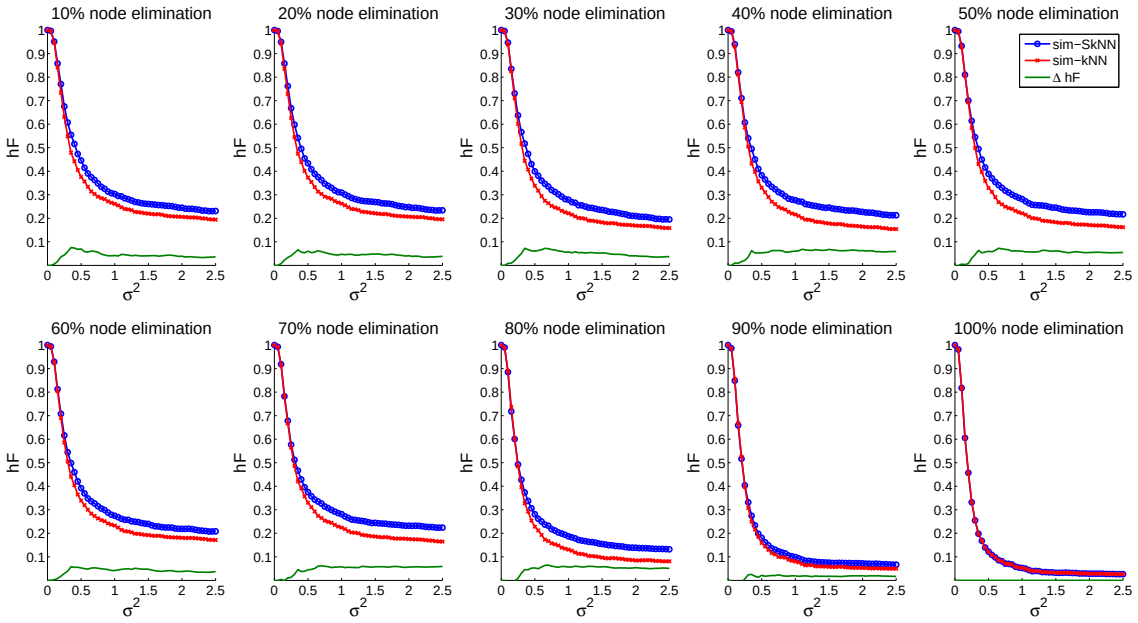


FIGURE 4.11: SkNN vs. kNN results for the second simulation condition, with the elimination perceptual error on the underlying 7-level binary tree taxonomy. Blue and red curves show *hF* for hierarchical and flat classification respectively, against the degree of noise in the features. Green curves show ΔhF, i.e., the performance gain in *hF* with the hierarchical approach.

node elimination and substitution, respectively, to define the perceived taxonomy as an altered version of the underlying 7-level binary tree taxonomy. Table 4.5 gives quantitative results for these experimental cases, where we can observe that the elimination of the interior nodes does not hamper the superiority of the hierarchical classification approach over the flat one, up to 90% of interior node removal. This can be explained by the fact that this type of misinterpretation of the hierarchical phenomenon does not violate the underlying "IS-A" or "PART-OF" partial order relationship present in the underlying taxonomy. Therefore, providing a prior taxonomy that is even severely altered by this error type is still beneficial to the classification problem. However, in the case of interior node substitution, the gain in performance with the hierarchical approach steadily decreases with the proportion of interior nodes that are swapped. Indeed, this type of misinterpretation violates the natural hierarchical order present in the underlying taxonomy of the phenomenon. We therefore conclude from these results that misinterpreting the hierarchical relationships in the underlying taxonomy with the second error type, i.e., violating the true partial order relationship existing between the states of the phenomenon, is more detrimental to the hierarchical classification approach than a misinterpretation with the first error type, i.e., oversimplifying the true hierarchical nature of the phenomenon.

4.5 Conclusion

Validating a general way to improve solutions to vision-based recognition problems represents a difficult and ambitious challenge. A promising avenue is to try to better emulate the biological way in which humans encode visual content. This can notably be approached via the design of

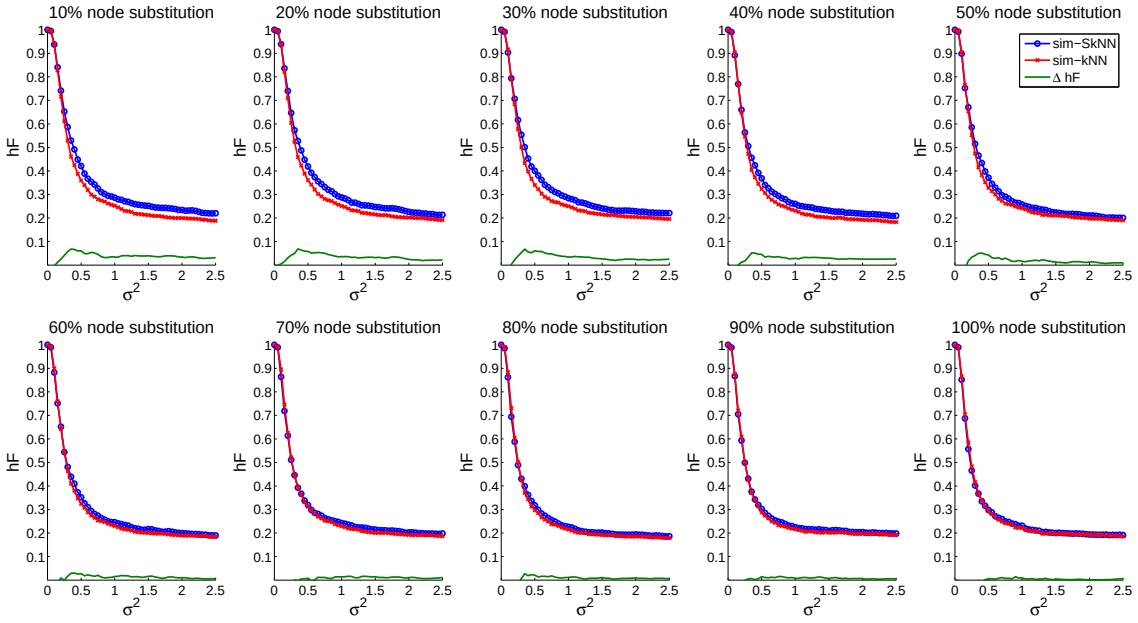


FIGURE 4.12: SkNN vs. kNN results for the second simulation condition, with the substitution perceptual error on the underlying 7-level binary tree taxonomy. Blue and red curves show *hF* for the hierarchical and flat classification respectively, against the degree of noise in the features. Green curves show ΔhF, i.e., the performance gain in *hF* with the hierarchical approach.

TABLE 4.5: Median and maximal ΔhF, i.e. performance gains in *hF* with the hierarchical approach, in our results for the second simulation condition shown in Fig. 4.11 and 4.12, for interior node elimination and substitution respectively.

	SkNN [with an altered perceived taxonomy] vs. kNN			
	Interior node elimination		Interior node substitution	
7-level binary tree alteration ratio	Med(ΔhF)	Max(ΔhF)	Med(ΔhF)	Max(ΔhF)
0% (no alteration, Tab. 4.4)	3.97%	7.52%	3.97%	7.52%
10% (6 interior nodes)	4.08%	7.58%	3.63%	6.80%
20% (13 interior nodes)	4.51%	6.62%	3.26%	6.90%
30% (19 interior nodes)	5.05%	7.28%	2.62%	6.79%
40% (25 interior nodes)	6.17%	6.82%	2.70%	5.19%
50% (31 interior nodes)	5.67%	7.28%	1.45%	5.11%
60% (38 interior nodes)	4.13%	5.80%	1.19%	2.95%
70% (44 interior nodes)	5.67%	6.17%	1.03%	1.63%
80% (50 interior nodes)	5.29%	6.50%	0.78%	2.77%
90% (57 interior nodes)	1.82%	2.51%	0.70%	1.53%
100% (all 63 interior nodes)	0.00%	0.00%	0.29%	1.45%

biologically-inspired feature representations, which hold some structure that echoes the visual encoding mechanism naturally made in the visual cortex [150, 151, 152]. The use of such richly structured feature representations has indeed been shown to improve the classification performance with some degree of generality [150, 153, 154]. Another intuitive path for improvement, which we chose to study in this chapter, is to apply a hierarchical approach to classification, i.e., to specify a taxonomy of concepts embodying the semantic and visual relationships between objects, and to use this taxonomy as a prior for the supervised learning process. Following this path is also encouraged by the fact that the general superiority of hierarchical classification over standard, flat classification has been demonstrated in other fields, such as text categorization and protein function prediction [124, 125, 122]. In these fields, we note that the feature representations used are typically high-level and that the possible states of the observed phenomenon are connected via well-understood hierarchical relationships. Enforcing such a hierarchical prior to the classification of visual content has been shown to be advantageous in some cases, e.g., [130, 131], but with less generality than in other fields [122].

Through the study presented in this chapter, we found that there was no added value in using a straightforward hierarchical approach with general-purpose visual features, for solving our problems of facial expression recognition and 3D shape recognition. However, we also showed via a simulation experiment that hierarchical methods can consistently outperform their flat counterparts, when provided with high-level features that capture the underlying hierarchical relationships present in the data, even when strong noise is added to these high-level features. Our results also showed that the performance gain offered by the hierarchical approach diminishes when the enforced prior taxonomy contains perceptual errors with respect to the underlying taxonomy of the phenomenon from which the data were obtained. These results suggest that vision-based recognition systems could generally benefit from the hierarchical classification approach provided that the following conditions are met: (1) rich, high-level feature representations must be used, designed to capture the underlying hierarchical information present in the measurements of the visual phenomenon, and (2) the underlying hierarchical nature of the visual phenomenon must be well-understood, as some errors in the perceived taxonomy may seriously hinder the benefit of using a hierarchical prior, even when proper hierarchical feature representations are used.

High-level hierarchical feature representations could be obtained via biologically-inspired design [150, 151, 152], or example-driven discovery, which includes information transfer [155, 156] and visual hierarchy learning [132, 133, 136]. Interestingly, the work in [130] on 3D shape recognition showed that the classification performance could be improved by using multiple flat binary classifiers, each trained to classify objects according to their separation by a node in a prior taxonomy. We consider this strategy as a tailor-made way to produce and aggregate high-level features in a hierarchical representation, which corroborates our conclusions in this chapter. In any case, using adequate hierarchical feature representations of the visual content will not bring forth the true potential of hierarchical classification, if the enforced prior taxonomies comprise serious perceptual errors. A deep understanding of the hierarchical semantics behind a visual phenomenon should be acquired before using hierarchical classification methods, which task could also be performed jointly with the design of high-level hierarchical feature representations, e.g., by building on a strategy similar to what was done in [136].

Chapter 5

Conclusion

The human visual system is one of the most effective in the animal kingdom. The prowess of humans in vision, combined with their unrivaled abilities to learn from concepts and recognize patterns, makes computer vision a challenging and exciting research field in artificial intelligence. Among the many vision-based cognitive processes that are tackled with computer vision, the automation of face perception tasks attracts a particular attention. Indeed, face perception, which designates all tasks of interpretation that are carried out by mere macroscopic observation of the face, is thoroughly used by humans in their everyday social interactions. It is actually one of the best mastered specialties of humans, to the point that specific areas of their brain are dedicated to face perception. Face perception tasks are also typically sensitive processes, and an automatic face perception system may quickly lead to experimenter bias because of the discomfort or confusion of the subject caused by the presence of the system. Because of both the clear benefits and intrinsic challenges linked to the automation of face perception tasks, it remains, as of today, a strong research focus in computer vision. Indeed, many face perception tasks have been automated by means of computer vision with remarkable success, yet often with room for significant improvement when compared to the effectiveness of human cognition. Among them, an important category consists of the tasks of facial expression interpretation, which range from recognizing facial muscle contraction patterns to recognizing subtle behaviors related to emotions, physiological states, or communication cues. In this thesis, we have taken a particular interest in the automation of specific tasks of facial expression interpretation.

In Chap. 2, we have presented our practical work on a vision-based system that extracts facial communication cues useful for the automatic recognition of sign language. We originally developed this system as a contribution to the SignSpeak project, where the goal was to create a new vision-based technology for translating sign language to text and improve the communication between deaf and hearing people. Our system is based on the robust AAM-based tracking of landmark points on a signer's face in a video, and continuously derives facial cues from the instantaneous configurations of the tracked landmark points. We showed with several quantitative evaluations that, ultimately, our facial cue extraction system can be advantageously integrated within a sign language recognition system chain. We consider these facial cues, which were defined through various discussions with linguistics and machine translation experts from the SignSpeak consortium, as specific facial expressions with an interpretation related to the recognition of sign language.

In Chap. 3, we have presented our practical work on a vision-based system designed to help clinicians in their bedside assessment of visual pursuit in post-comatose patients. Visual pursuit

is a key clinical marker to assess post-comatose states, and it is a sensitive facial expression-based perception task. Our system is used with a head-mounted device, and it is designed to assist the clinician by providing objective measures of the visual pursuit ability of the patient being tested. Combined with the use of a head-mounted device, our system conforms to the precise clinical guidelines developed by neuroscience researchers for properly performing visual pursuit assessment. Notably, our system is able to work with the recommended visual stimulus, which is a handheld mirror moved by the clinican. We presented the results obtained with our system on post-comatose patients and healthy control subjects. These results showed that our system can be used in a hospital environment to enhance the clinical assessment of visual pursuit.

In Chap. 4, we have presented the empirical study we conducted on the hierarchical and standard, "flat" approaches to classification in the problems of muscle-based facial expression recognition and 3D shape recognition. In hierarchical classification, a semantic class hierarchy is used to enforce a fine-grained notion of semantic similarity between the classes of the problem, so as to discover an overall better classifier than one obtained with flat classification. Our goal was not to provide a new computer vision system for an application of facial expression recognition or 3D shape recognition in that chapter, but to to compare the hierarchical and flat classification approaches in these specific vision-based problems. Through our experiments, we found that, unexpectedly, there was little to no improvement in the recognition performance by using hierarchical classification with visual features provided by off-the-shelf feature extraction methods. Through additional experiments we conducted in a simulation, we found useful general conditions about feature representations and semantic class hierarchies that should be met in order to get the benefits of using hierarchical classification. These conditions should in theory also apply to hierarchical classification in muscle-based facial expression recognition.

At this point, we would like to mention another contribution we made within the application domain of drowsiness monitoring. Drowsiness is generally defined as the intermediate physiological state between wakefulness and sleep. It can lead to a temporary impairment of performance that may be dangerous in various private and professional activities, including those of the transportation and construction industries, to name a few. The hazards caused by drowsiness have motivated the research in drowsiness monitoring, and the search for effective systems able to automatically, continuously, and objectively estimate the level of drowsiness of a person busy at a task. We participated in the development of a fully automatic drowsiness monitoring system based on the analysis of ocular parameters. This system produces a numerical level of drowsiness with great reliability [157], and has therefore significant potential for preventing drowsiness-related accidents. To best promote the advantages of this system, a spin-off company has emerged at the university of Liège, where this system is currently exploited through a commercial activity. Our contribution to this drowsiness monitoring system consists of the software module that performs the automatic extraction of the ocular parameters from images of the eye. Figure 5.1 illustrates the effectiveness of our ocular parameter extraction module. We consider that these ocular parameters are particular expressions with a specific interpretation, according to our definition in Chap. 1. Specifically, we consider that our module automates a task of facial expression interpretation that is related to the recognition of a physiological activity.

To conclude, we would like to further discuss the concept of facial expression interpretation and its automation in a broad sense. In the present document, we started off by giving our own definition of a facial expression and its interpretation in the introductory chapter (Chap. 1). Specifically, we extended the usual facial muscle-based definition to include head and eye movements, and we emphasized that an interpretation of a facial expression is in many cases related to a hidden process that notably shows through the face, but is otherwise most often dependent on context information that is not necessarily present in the face image. This latter aspect about context dependency is of particular importance in practice, because it often significantly influences the design of a computer vision solution for facial expression interpretation. As a matter of fact, numerous computer vision solutions quite heterogeneous in their design have been proposed for various tasks of facial expression interpretation where a fixed context is assumed, because fixing the context helps to simplify the analysis and obtain effectiveness for specific applications. As effective as fixed-context solutions are (including the solutions we proposed in Chap. 2 and Chap. 3, as well as our contribution to a drowsiness monitoring system illustrated in Fig. 5.1), their heterogeneous design makes them not obviously reusable in other applications. One could be satisfied with this state of affairs, and consider the automation of facial expression interpretation tasks as an umbrella term for various computer vision problems that necessarily require significantly different solutions. It is besides not obvious how a general-purpose mechanism should be designed for incorporating nonfacial, even nonvisual context information within a unified automation approach for facial expression interpretation. Yet such a context integration mechanism could be the way toward raising computer vision solutions to human-like abilities in tasks of facial expression interpretation, as humans are not only remarkable for their precision in performing such tasks, but also for their flexibility to adapt to any current situation, i.e., any particular context. With the recent introduction of methods and hardware able to effectively leverage large amounts of semantically annotated data of general nature, namely the machine learning approach known as deep learning, new methodological opportunities have emerged notably to automate visual perception with unprecedented effectiveness. Indeed, the use of the deep learning approach results in a rapid and steady evolution of the performance standards in the automation of all tasks of visual perception. However, the spectacular effectiveness of a deep learning approach solely based on visual data is still only indicative of the full extent of a visual analysis made in a fixed, particular context. In other words, one cannot hope to resolve visual ambiguities due to a variable context by the sole yet thorough analysis of the available visual content. Perhaps that the underlying flexibility of the deep learning approach could also be the way to design general-purpose integration mechanisms of nonvisual context information, to dynamically adapt the vision-based analysis to variable context, notably for face perception tasks, and especially for tasks of facial expression interpretation.

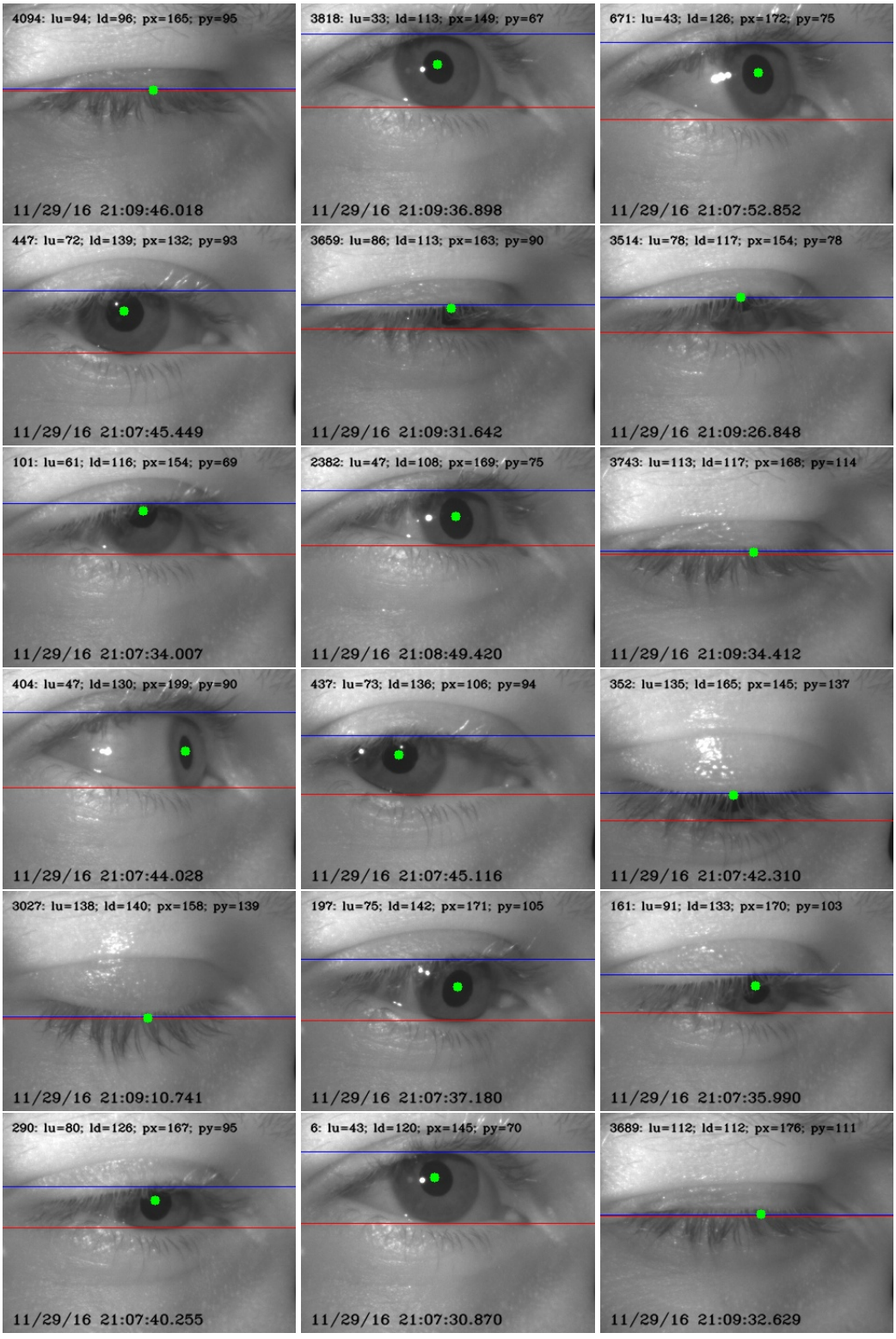


FIGURE 5.1: Ocular parameters automatically extracted with our software module (integrated in a drowsiness monitoring system). The blue and red horizontal lines are indicative of the calculated vertical positions of the upper and lower eyelids in the image, respectively. The green dot indicates the calculated position of the center of the pupil in the image. These ocular parameters are extracted in real-time with great robustness and accuracy among individuals, notably for new users of the system, as it does not require any preparatory calibration procedure.

Bibliography

[1] Gail W. Jenkins and Gerard J. Tortora. *Anatomy and physiology: From science to life*. 3rd ed. Hoboken, NJ: John Wiley & Sons, 2013.

[2] James V. Haxby, Elizabeth A. Hoffman, and M. Ida Gobbini. "The distributed human neural system for face perception". *Trends in Cognitive Sciences*, 4(6), pp. 223–233, 2000.

[3] Maja Pantic and Leon J. M. Rothkrantz. "Automatic analysis of facial expressions: The state of the art". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12), pp. 1424–1445, 2000.

[4] Bruno Rossion, Bernard Hanseeuw, and Laurence Dricot. "Defining face perception areas in the human brain: A large-scale factorial fMRI face localizer analysis". *Brain and Cognition*, 79(2), pp. 138–157, 2012.

[5] Nancy Kanwisher, Josh McDermott, and Marvin M. Chun. "The fusiform face area: A module in human extrastriate cortex specialized for face perception". *Journal of Neuroscience*, 17(11), pp. 4302–4311, 1997.

[6] Charles A. Nelson. "The development and neural bases of face recognition". *Infant and Child Development*, 10(1-2), pp. 3–18, 2001.

[7] Stefanie Hoehl and Tricia Striano. "Neural processing of eye gaze and threat-related emotional facial expressions in infancy". *Child Development*, 79(6), pp. 1752–1760, 2008.

[8] Johannes Sobotta, J. Playfair McMurrich, and William H. Thomas. *Atlas and text-book of human anatomy*. Vol. 3. Philadelphia, PA: W.B. Saunders Company, 1909.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet classification with deep convolutional neural networks". In: *Proceedings of the 26th Annual Conference on Neural Information Processing Systems*, pp. 1097–1105. Stateline, NV, 2012.

[10] Andrej Karpathy and Li Fei-Fei. "Deep visual-semantic alignments for generating image descriptions". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4), pp. 664–676, 2017.

[11] Anil K. Jain, Ruud M. Bolle, and Sharath Pankanti. *Biometrics: Personal identification in networked society*. New York, NY: Springer Publishing, 2006.

[12] Arun Hampapur, Lisa Brown, Jonathan Connell, Ahmet Ekin, et al. "Smart video surveillance: Exploring the concept of multiscale spatiotemporal tracking". *IEEE Signal Processing Magazine*, 22(2), pp. 38–51, 2005.

[13] Dayong Wang, Steven C.H. Hoi, Ying He, and Jianke Zhu. "Mining weakly labeled web facial images for search-based face annotation". *IEEE Transactions on Knowledge and Data Engineering*, 26(1), pp. 166–179, 2014.

[14] Siddharth S. Rautaray and Anupam Agrawal. "Vision based hand gesture recognition for human computer interaction: A survey". *Artificial Intelligence Review*, 43(1), pp. 1–54, 2015.

[15] Peter Lewinski, Marieke L. Fransen, and Ed S.H. Tan. "Predicting advertising effectiveness by facial expressions in response to amusing persuasive stimuli". *Journal of Neuroscience, Psychology, and Economics*, 7(1), pp. 1–14, 2014.

[16] Stefan Scherer, Giota Stratou, Marwa Mahmoud, Jill Boberg, et al. "Automatic behavior descriptors for psychological disorder analysis". In: *Proceedings of the 10th IEEE International Conference on Automatic Face & Gesture Recognition*, pp. 1–8. Shanghai, China, 2013.

[17] Paul Viola and Michael Jones. "Rapid object detection using a boosted cascade of simple features". In: *Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 511–518. Kauai, HI, 2001.

[18] Cha Zhang and Zhengyou Zhang. *A survey of recent advances in face detection*. Tech. rep. MSR-TR-2010-66. Redmond, WA: Microsoft Research, 2010.

[19] Timothy F. Cootes and Christopher J. Taylor. "Active shape models – 'Smart snakes'". In: *Proceedings of the 3rd British Machine Vision Conference*, pp. 266–275. Leeds, UK, 1992.

[20] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. "Active appearance models". In: *Proceedings of the 5th European Conference on Computer Vision*, pp. 484–498. Freiburg, Germany, 1998.

[21] Ralph Gross, Iain Matthews, and Simon Baker. "Generic vs. person specific active appearance models". *Image and Vision Computing*, 23(12), pp. 1080–1093, 2005.

[22] Jason M. Saragih, Simon Lucey, and Jeffrey F. Cohn. "Deformable model fitting by regularized landmark mean-shift". *International Journal of Computer Vision*, 91(2), pp. 200–215, 2011.

[23] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. "Face alignment by explicit shape regression". *International Journal of Computer Vision*, 107(2), pp. 177–190, 2014.

[24] Yi Sun, Xiaogang Wang, and Xiaoou Tang. "Deep convolutional network cascade for facial point detection". In: *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3476–3483. Portland, OR, 2013.

[25] Adin Ramirez Rivera, Jorge Rojas Castillo, and Oksam Chae. "Local directional number pattern for face analysis: Face and expression recognition". *IEEE Transactions on Image Processing*, 22(5), pp. 1740–1752, 2013.

[26] Wenyi Zhao, Rama Chellappa, P. Jonathon Phillips, and Azriel Rosenfeld. "Face recognition: A literature survey". *Association for Computing Machinery: Computing Surveys*, 35(4), pp. 399–458, 2003.

[27] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. "DeepFace: Closing the gap to human-level performance in face verification". In: *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701–1708. Columbus, OH, 2014.

[28] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. *Labeled faces in the wild: A database for studying face recognition in unconstrained environments*. Tech. rep. 07-49. Amherst, MA: University of Massachusetts Amherst, 2007.

[29] Christos Sagonas, Epameinondas Antonakos, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. "300 faces in-the-wild challenge: Database and results". *Image and Vision Computing*, 47, pp. 3–18, 2016.

[30] Brendan F. Klare, Ben Klein, Emma Taborsky, Austin Blanton, et al. "Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A". In: *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1931–1939. Boston, MA, 2015.

[31] Abhinav Dhall, Roland Goecke, Jyoti Joshi, Karan Sikka, and Tom Gedeon. "Emotion recognition in the wild challenge 2014: Baseline, data and protocol". In: *Proceedings of the 16th International Conference on Multimodal Interaction*, pp. 461–466. Istanbul, Turkey, 2014.

[32] Mengyi Liu, Ruiping Wang, Shaoxin Li, Shiguang Shan, et al. "Combining multiple kernel methods on Riemannian manifold for emotion recognition in the wild". In: *Proceedings of the 16th International Conference on Multimodal Interaction*, pp. 494–501. Istanbul, Turkey, 2014.

[33] Yin Fan, Xiangju Lu, Dian Li, and Yuanliu Liu. "Video-based emotion recognition using CNN-RNN and C3D hybrid networks". In: *Proceedings of the 18th International Conference on Multimodal Interaction*, pp. 445–450. Tokyo, Japan, 2016.

[34] Hu Han, Charles Otto, and Anil K. Jain. "Age estimation from face images: Human vs. machine performance". In: *Proceedings of the 6th International Conference on Biometrics*, pp. 1–8. Madrid, Spain, 2013.

[35] Martin J. Tovée, Joanne L. Emery, and Esther M. Cohen-Tovée. "The estimation of body mass index and physical attractiveness is dependent on the observer's own body mass index". *Proceedings of the Royal Society of London B: Biological Sciences*, 267(1456), pp. 1987–1997, 2000.

[36] Paul Ekman and Erika L. Rosenberg. *What the face reveals: Basic and applied studies of spontaneous expression using the facial action coding system (FACS)*. 2nd ed. Oxford, UK: Oxford University Press, 2005.

[37] Philippe Dreuw, Jens Forster, Yannick Gweth, Daniel Stein, et al. "Signspeak – Understanding, recognition, and translation of sign languages". In: *Proceedings of the 4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, pp. 65–72. Valletta, Malta, 2010.

[38] Justus H. Piater, Thomas Hoyoux, and Wei Du. "Video analysis for continuous sign language recognition". In: *Proceedings of the 4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, pp. 192–195. Valletta, Malta, 2010.

[39] Jens Forster, Christoph Schmidt, Thomas Hoyoux, Oscar Koller, et al. "RWTH-PHOENIX-Weather: A large vocabulary sign language recognition and translation corpus". In: *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pp. 3785–3789. Istanbul, Turkey, 2012.

[40] Christoph Schmidt, Oscar Koller, Hermann Ney, Thomas Hoyoux, and Justus H. Piater. "Using viseme recognition to improve a sign language translation system". In: *Proceedings of the 10th International Workshop on Spoken Language Translation*, pp. 197–203. Heidelberg, Germany, 2013.

[41] Christoph Schmidt, Oscar Koller, Hermann Ney, Thomas Hoyoux, and Justus H. Piater. "Enhancing gloss-based corpora with facial features using active appearance models". In: *Proceedings of the 3rd International Symposium on Sign Language Translation and Avatar Technology*, pp. 1–8. Chicago, IL, 2013.

[42] Ceil Lucas. *The sociolinguistics of sign languages*. Cambridge, UK: Cambridge University Press, 2001.

[43] Ted Supalla and Rebecca Webb. "The grammar of International Sign: A new look at pidgin languages". *Language, Gesture, and Space*, pp. 333–352, 1995.

[44] William C. Stokoe Jr. "Sign language structure: An outline of the visual communication systems of the American deaf". *Journal of Deaf Studies and Deaf Education*, 10(1), pp. 3–37, 2005.

[45] Jerome D. Schein and Mona Mark. *Speaking the language of sign: The art and science of signing*. New York, NY: Doubleday, 1984.

[46] Pauline Conroy. *Signing in & signing out: The education and employment experiences of deaf adults in Ireland: A study of inequality and deaf people in Ireland*. Dublin, Ireland: Irish Deaf Society, 2006.

[47] Karen McQuigg. "Are the deaf a disabled group, or a linguistic minority? Issues for librarians in Victoria's public libraries". *Australian Library Journal*, 52(4), pp. 367–377, 2003.

[48] Philippe Dreuw, Hermann Ney, Gregorio Pérez Martínez, Onno A. Crasborn, et al. "The SignSpeak project – Bridging the gap between signers and speakers". In: *Proceedings of the 7th International Conference on Language Resources and Evaluation*, pp. 476–481. Valletta, Malta, 2010.

[49] Myriam Vermeerbergen, Lorraine Leeson, and Onno A. Crasborn. *Simultaneity in signed languages: Form and function*. Amsterdam, The Netherlands: John Benjamins Publishing Company, 2007.

[50] Scott K. Liddell. *Grammar, gesture, and meaning in American sign language*. Cambridge, UK: Cambridge University Press, 2003.

[51] Barry B. Powell. *Writing: Theory and history of the technology of civilization*. Hoboken, NJ: Wiley-Blackwell, 2009.

[52] Elena Pizzuto, Paolo Rossini, and Tommaso Russo. "Representing signed languages in written form: Questions that need to be posed". In: *Proceedings of the 2nd Workshop on the Representation and Processing of Sign Languages: Lexicographic Matters and Didactic Scenarios*, pp. 1–6. Genoa, Italy, 2006.

[53] Philipp Koehn. *Statistical machine translation*. Cambridge, UK: Cambridge University Press, 2009.

[54] David Rybach, Christian Gollan, Georg Heigold, Björn Hoffmeister, et al. "The RWTH Aachen University open source speech recognition system". In: *Proceedings of the 10th Annual Conference of the International Speech Communication Association*, pp. 2111–2114. Brighton, UK, 2009.

[55] Patrick Buehler, Andrew Zisserman, and Mark Everingham. "Learning sign language by watching TV (using weakly aligned subtitles)". In: *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2961–2968. Miami, FL, 2009.

[56] Helen Cooper and Richard Bowden. "Learning signs from subtitles: A weakly supervised approach to sign language recognition". In: *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2568–2574. Miami, FL, 2009.

[57] Hee-Deok Yang, Stan Sclaroff, and Seong-Whan Lee. "Sign language spotting with a threshold model based on conditional random fields". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(7), pp. 1264–1277, 2009.

[58] Jens Forster, Christian Oberdörfer, Oscar Koller, and Hermann Ney. "Modality combination techniques for continuous sign language recognition". In: *Proceedings of the 6th Iberian Conference on Pattern Recognition and Image Analysis*, pp. 89–99. Madeira, Portugal, 2013.

[59] Jason M. Saragih, Simon Lucey, and Jeffrey F. Cohn. "Real-time avatar animation from a single image". In: *Proceedings of the 9th IEEE International Conference on Automatic Face & Gesture Recognition*, pp. 117–124. Santa Barbara, CA, 2011.

[60] Shaoqing Ren, Xudong Cao, Yichen Wei, and Jian Sun. "Face alignment at 3000 FPS via regressing local binary features". In: *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1685–1692. Columbus, OH, 2014.

[61] Xin Jin and Xiaoyang Tan. "Face alignment in-the-wild: A survey". arXiv: 1608.04188, 2016.

[62] Iain Matthews, Takahiro Ishikawa, and Simon Baker. "The template update problem". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6), pp. 810–815, 2004.

[63] Timothy F. Cootes and Cristopher J. Taylor. *Statistical models of appearance for computer vision*. Tech. rep. Manchester, UK: University of Manchester, 2004.

[64] Ian L. Dryden and Kanti V. Mardia. *Statistical shape analysis*. Hoboken, NJ: John Wiley & Sons, 1998.

[65] Iain Matthews and Simon Baker. "Active appearance models revisited". *International Journal of Computer Vision*, 60(2), pp. 135–164, 2004.

[66] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent component analysis*. Hoboken, NJ: John Wiley & Sons, 2001.

[67] Stan Sclaroff and John Isidoro. "Active blobs". In: *Proceedings of the 6th International Conference on Computer Vision*, pp. 1146–1153. Bombay, India, 1998.

[68] Simon Baker and Iain Matthews. "Lucas-Kanade 20 years on: A unifying framework". *International Journal of Computer Vision*, 56(3), pp. 221–255, 2004.

[69] Jing Xiao, Simon Baker, Iain Matthews, and Takeo Kanade. "Real-time combined 2D+3D active appearance models". In: *Proceedings of the 2004 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 535–542. Washington, DC, 2004.

[70] Lorenzo Torresani, Aaron Hertzmann, and Chris Bregler. "Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5), pp. 878–892, 2008.

[71] Daniel F. Dementhon and Larry S. Davis. "Model-based object pose in 25 lines of code". *International Journal of Computer Vision*, 15(1-2), pp. 123–141, 1995.

[72] Onno A. Crasborn and Inge Zwitserlood. "The Corpus NGT: An online corpus for professionals and laymen". In: *Proceedings of the 3rd Workshop on the Representation and Processing of Sign Languages: Construction and Exploitation of Sign Language Corpora*, pp. 44–49. Marrakech, Morocco, 2008.

[73] Alexander Klaser, Marcin Marszałek, and Cordelia Schmid. "A spatio-temporal descriptor based on 3D-gradients". In: *Proceedings of the 19th British Machine Vision Conference*, pp. 275–284. Leeds, UK, 2008.

[74] Ulrich Von Agris, Moritz Knorr, and Karl-Friedrich Kraiss. "The significance of facial features for automatic sign language recognition". In: *Proceedings of the 8th IEEE International Conference on Automatic Face & Gesture Recognition*, pp. 1–6. Amsterdam, The Netherlands, 2008.

[75] Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. "A study of translation edit rate with targeted human annotation". In: *Proceedings of the 7th Biennial Conference of the Association for Machine Translation in the Americas*, pp. 223–231. Cambridge, MA, 2006.

[76] Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. "Report on the 11th IWSLT evaluation campaign, IWSLT 2014". In: *Proceedings of the 11th International Workshop on Spoken Language Translation*, pp. 2–11. South Lake Tahoe, CA, 2014.

[77] Trevor Johnston. “The lexical database of Auslan (Australian sign language)”. *Sign Language & Linguistics*, 4(1-2), pp. 145–169, 2001.

[78] Oscar Koller, Hermann Ney, and Richard Bowden. “Weakly supervised automatic transcription of mouthings for gloss-based sign language corpora”. In: *Proceedings of the 6th Workshop on the Representation and Processing of Sign Languages: Beyond the Manual Channel*, pp. 89–94. Reykjavik, Iceland, 2014.

[79] Oscar Koller, Hermann Ney, and Richard Bowden. “Read my lips: Continuous signer independent weakly supervised viseme recognition”. In: *Proceedings of the 13th European Conference on Computer Vision*, pp. 281–296. Zurich, Switzerland, 2014.

[80] Oscar Koller, Jens Forster, and Hermann Ney. “Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers”. *Computer Vision and Image Understanding*, 141, pp. 108–125, 2015.

[81] Oscar Koller, Hermann Ney, and Richard Bowden. “Deep learning of mouth shapes for sign language”. In: *Proceedings of the 3rd Workshop on Assistive Computer Vision and Robotics*, pp. 477–483. Santiago, Chile, 2015.

[82] Oscar Koller, Sepehr Zargaran, and Hermann Ney. “Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent CNN-HMMs”. In: *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9. Honolulu, HI, 2017.

[83] Thomas Hoyoux, Sarah Wannez, Thomas Langohr, Jérôme Wertz, et al. “A new computer vision-based system to help clinicians objectively assess visual pursuit with the moving mirror stimulus for the diagnosis of minimally conscious state”. In: *Proceedings of the 2016 IEEE Winter Conference on Applications of Computer Vision*, pp. 1–8. Lake Placid, NY, 2016.

[84] Sarah Wannez, Thomas Hoyoux, Thomas Langohr, Olivier Bodart, et al. “Objective assessment of visual pursuit in patients with disorders of consciousness: An exploratory study”. *Journal of Neurology*, 264(5), pp. 928–937, 2017.

[85] Steven Laureys, Gastone G. Celesia, François Cohadon, Jan Lavrijsen, et al. “Unresponsive wakefulness syndrome: A new name for the vegetative state or apallic syndrome”. *BioMed Central: Medicine*, 8, p. 68, 2010.

[86] Joseph T. Giacino, Stephen Ashwal, Nancy Childs, Ronald Cranford, et al. “The minimally conscious state: Definition and diagnostic criteria”. *Neurology*, 58(3), pp. 349–353, 2002.

[87] Marie-Aurélie Bruno, Steve Majerus, Mélanie Boly, Audrey Vanhaudenhuyse, et al. “Functional neuroanatomy underlying the clinical subcategorization of minimally conscious state patients”. *Journal of Neurology*, 259(6), pp. 1087–1098, 2012.

[88] Fred Plum and Jerome B. Posner. *The diagnosis of stupor and coma*. Philadelphia, PA: F.A. Davis Company, 1966.

[89] Mélanie Boly, Marie-Elisabeth Faymonville, Philippe Peigneux, Bernard Lambermont, et al. "Cerebral processing of auditory and noxious stimuli in severely brain injured patients: Differences between VS and MCS". *Neuropsychological Rehabilitation*, 15(3-4), pp. 283–289, 2005.

[90] Jacques Luauté, Delphine Maucort-Boulch, Laurence Tell, François Quelard, et al. "Long-term outcomes of chronic minimally conscious and vegetative states". *Neurology*, 75(3), pp. 246–252, 2010.

[91] Joseph T. Giacino, Kathleen Kalmar, and John Whyte. "The JFK Coma Recovery Scale-Revised: Measurement characteristics and diagnostic utility". *Archives of Physical Medicine and Rehabilitation*, 85(12), pp. 2020–2029, 2004.

[92] Ronald T. Seel, Mark Sherer, John Whyte, Douglas I. Katz, et al. "Assessment scales for disorders of consciousness: Evidence-based recommendations for clinical practice and research". *Archives of Physical Medicine and Rehabilitation*, 91(12), pp. 1795–1813, 2010.

[93] Caroline Schnakers, Audrey Vanhaudenhuyse, Joseph T. Giacino, Manfredi Ventura, et al. "Diagnostic accuracy of the vegetative and minimally conscious state: Clinical consensus versus standardized neurobehavioral assessment". *BioMed Central: Neurology*, 9, p. 35, 2009.

[94] Sergio Bagnato, Cristina Boccagni, Antonino Sant'Angelo, Alexander A. Fingelkurts, et al. "Longitudinal assessment of clinical signs of recovery in patients with unresponsive wakefulness syndrome after traumatic or nontraumatic brain injury". *Journal of Neurotrauma*, 34(2), pp. 535–539, 2017.

[95] Anna Estraneo, Pasquale Moretta, Viviana Cardinale, Antonio De Tanti, et al. "A multicentre study of intentional behavioural responses measured using the Coma Recovery Scale-Revised in patients with minimally conscious state". *Clinical Rehabilitation*, 29(8), pp. 803–808, 2015.

[96] Marie Thonnard, Sarah Wannez, Shannan Keen, Serge Brédart, et al. "Detection of visual pursuit in patients in minimally conscious state: A matter of stimuli and visual plane?" *Brain Injury*, 28(9), pp. 1164–1170, 2014.

[97] Audrey Vanhaudenhuyse, Caroline Schnakers, Serge Brédart, and Steven Laureys. "Assessment of visual pursuit in post-comatose states: Use a mirror". *Journal of Neurology, Neurosurgery & Psychiatry*, 79(2), p. 223, 2008.

[98] Damian Cruse, Marco Fattizzo, Adrian M. Owen, and Davinia Fernández-Espejo. "Why use a mirror to assess visual pursuit in prolonged disorders of consciousness? Evidence from healthy control participants". *BioMed Central: Neurology*, 17, p. 14, 2017.

[99] Martin M. Monti, Audrey Vanhaudenhuyse, Martin R. Coleman, Mélanie Boly, et al. "Willful modulation of brain activity in disorders of consciousness". *New England Journal of Medicine*, 362(7), pp. 579–589, 2010.

[100] Damian Cruse, Srivas Chennu, Camille Chatelle, Davinia Fernández-Espejo, et al. "Relationship between etiology and covert cognition in the minimally conscious state". *Neurology*, 78(11), pp. 816–822, 2012.

[101] Tristan A. Bekinschtein, Martin R. Coleman, Jorge Niklison, John D. Pickard, and Facundo F. Manes. "Can electromyography objectively detect voluntary movement in disorders of consciousness?" *Journal of Neurology, Neurosurgery & Psychiatry*, 79(7), pp. 826–828, 2008.

[102] Johan Stender, Olivia Gosseries, Marie-Aurélie Bruno, Vanessa Charland-Verville, et al. "Diagnostic precision of PET imaging and functional MRI in disorders of consciousness: A clinical validation study". *The Lancet*, 384(9942), pp. 514–522, 2014.

[103] Johan Stender, Ron Kupers, Anders Rodell, Aurore Thibaut, et al. "Quantitative rates of brain glucose metabolism distinguish minimally conscious from vegetative state patients". *Journal of Cerebral Blood Flow & Metabolism*, 35(1), pp. 58–65, 2015.

[104] Olivia Gosseries, Aurore Thibaut, Mélanie Boly, Mario Rosanova, et al. "Assessing consciousness in coma and related states using transcranial magnetic stimulation combined with electroencephalography". *Annales Françaises d'Anesthésie et de Réanimation*, 33(2), pp. 65–71, 2014.

[105] Luigi Trojano, Pasquale Moretta, Vincenzo Loreto, Autilia Cozzolino, et al. "Quantitative assessment of visual behavior in disorders of consciousness". *Journal of Neurology*, 259(9), pp. 1888–1895, 2012.

[106] Luigi Trojano, Pasquale Moretta, Vincenzo Loreto, Lucio Santoro, and Anna Estraneo. "Affective saliency modifies visual tracking behavior in disorders of consciousness: A quantitative analysis". *Journal of Neurology*, 260(1), pp. 306–308, 2013.

[107] Dan Witzner Hansen and Qiang Ji. "In the eye of the beholder: A survey of models for eyes and gaze". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3), pp. 478–500, 2010.

[108] Vincent Lepetit and Pascal Fua. "Monocular model-based 3D tracking of rigid objects: A survey". *Foundations and Trends in Computer Graphics and Vision*, 1(1), pp. 1–89, 2005.

[109] Rui Rodrigues, João P. Barreto, and Urbano Nunes. "Camera pose estimation using images of planar mirror reflections". In: *Proceedings of the 11th European Conference on Computer Vision*, pp. 382–395. Heraklion, Greece, 2010.

[110] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. 2nd ed. Cambridge, UK: Cambridge University Press, 2004.

[111] Greg Welch and Gary Bishop. *An introduction to the Kalman filter*. Tech. rep. TR 95-041. Chapel Hill, NC: University of North Carolina at Chapel Hill, 1995.

[112] Evimaria Terzi. "Problems and algorithms for sequence segmentations". PhD thesis. Finland: University of Helsinki, 2006.

[113] Anthony J. Onwuegbuzie, Larry Daniel, and Nancy L. Leech. "Pearson product-moment correlation coefficient". In: *Encyclopedia of measurement and statistics*, pp. 750–755. Thousand Oaks, CA: SAGE Publications, 2007.

[114] Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. "Torch7: A MATLAB-like environment for machine learning". In: *Proceedings of the 1st Workshop on Big Learning: Algorithms, Systems, and Tools for Learning at Scale*, pp. 1–6. Granada, Spain, 2011.

[115] Vinod Nair and Geoffrey E. Hinton. "Rectified linear units improve restricted Boltzmann machines". In: *Proceedings of the 27th International Conference on Machine Learning*, pp. 807–814. Haifa, Israel, 2010.

[116] J. Richard Landis and Gary G. Koch. "The measurement of observer agreement for categorical data". *Biometrics*, 33(1), pp. 159–174, 1977.

[117] Athena Demertzi, Didier Ledoux, Marie-Aurélie Bruno, Audrey Vanhaudenhuyse, et al. "Attitudes towards end-of-life issues in disorders of consciousness: A European survey". *Journal of Neurology*, 258(6), pp. 1058–1065, 2011.

[118] Athena Demertzi, Caroline Schnakers, Didier Ledoux, Camille Chatelle, et al. "Different beliefs about pain perception in the vegetative and minimally conscious states: A European survey of medical and paramedical professionals". *Progress in Brain Research*, 177, pp. 329–338, 2009.

[119] Marie-Aurélie Bruno, Didier Ledoux, Audrey Vanhaudenhuyse, Olivia Gosseries, et al. "Pronostic des patients récupérant du coma". In: *Coma et états de conscience altérée*. Chap. 2, pp. 17–29. Paris, France: Springer Verlag, 2011.

[120] Thomas Hoyoux, Antonio J. Rodríguez-Sánchez, and Justus H. Piater. "Can computer vision problems benefit from structured hierarchical classification?" *Machine Vision and Applications*, 27(8), pp. 1299–1312, 2016.

[121] Thomas Hoyoux, Antonio J. Rodríguez-Sánchez, Justus H. Piater, and Sandor Szedmak. "Can computer vision problems benefit from structured hierarchical classification?" In: *Proceedings of the 16th International Conference on Computer Analysis of Images and Patterns*, pp. 403–414. Valletta, Malta, 2015.

[122] Carlos N. Silla Jr. and Alex A. Freitas. "A survey of hierarchical classification across different application domains". *Data Mining and Knowledge Discovery*, 22(1-2), pp. 31–72, 2011.

[123] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. "Support vector machine learning for interdependent and structured output spaces". In: *Proceedings of the 21st International Conference on Machine Learning*, pp. 104–111. Banff, Canada, 2004.

[124] Miguel E. Ruiz and Padmini Srinivasan. "Hierarchical text categorization using neural networks". *Information Retrieval*, 5(1), pp. 87–118, 2002.

[125] Roman Eisner, Brett Poulin, Duane Szafron, Paul Lu, and Russ Greiner. "Improving protein function prediction using the hierarchical structure of the gene ontology". In: *Proceedings of the 2005 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pp. 1–10. San Diego, CA, 2005.

[126] Jin Ha Lee and J. Stephen Downie. "Survey of music information needs, uses, and seeking behaviours: Preliminary findings". In: *Proceedings of the 5th International Conference on Music Information Retrieval*, pp. 1–6. Barcelona, Spain, 2004.

[127] Tai Sing Lee and David Mumford. "Hierarchical Bayesian inference in the visual cortex". *Journal of the Optical Society of America A: Optics, Image Science, and Vision*, 20(7), pp. 1434–1448, 2003.

[128] Yukako Yamane, Eric T. Carlson, Katherine C. Bowman, Zhihong Wang, and Charles E. Connor. "A neural code for three-dimensional object shape in macaque inferotemporal cortex". *Nature Neuroscience*, 11(11), pp. 1352–1360, 2008.

[129] Carlo Baldassi, Alireza Alemi-Neissi, Marino Pagan, James J. DiCarlo, et al. "Shape similarity, better than semantic membership, accounts for the structure of visual object representations in a population of monkey inferotemporal neurons". *Public Library of Science: Computational Biology*, 9(8), e1003167, 2013.

[130] Zafer Barutcuoglu and Christopher DeCoro. "Hierarchical shape classification using Bayesian aggregation". In: *Proceedings of the 2006 IEEE International Conference on Shape Modeling and Applications*, pp. 289–293. Matsushima, Japan, 2006.

[131] Ivica Dimitrovski, Dragi Kocev, Suzana Loskovska, and Sašo Džeroski. "Hierarchical annotation of medical images". *Pattern Recognition*, 44(10-11), pp. 2436–2449, 2011.

[132] Marcin Marszałek and Cordelia Schmid. "Constructing category hierarchies for visual recognition". In: *Proceedings of the 10th European Conference on Computer Vision*, pp. 479–491. Marseille, France, 2008.

[133] Gregory Griffin and Pietro Perona. "Learning and using taxonomies for fast visual categorization". In: *Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8. Anchorage, AK, 2008.

[134] Roni Mittelman, Min Sun, Benjamin Kuipers, and Silvio Savarese. "A Bayesian generative model for learning semantic hierarchies". *Frontiers in Psychology*, 5, p. 417, 2014.

[135] Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. "Semantic taxonomy induction from heterogenous evidence". In: *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pp. 801–808. Sidney, Australia, 2006.

[136] Li-Jia Li, Chong Wang, Yongwhan Lim, David M. Blei, and Li Fei-Fei. "Building and using a semantivisual image hierarchy". In: *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3336–3343. San Francisco, CA, 2010.

[137] Peter Lewinski, Tim M. den Uyl, and Crystal Butler. "Automated facial coding: Validation of basic emotions and FACS AUs in FaceReader". *Journal of Neuroscience, Psychology, and Economics*, 7(4), pp. 227–236, 2014.

[138] Philip Shilane, Patrick Min, Michael Kazhdan, and Thomas Funkhouser. "The Princeton Shape Benchmark". In: *Proceedings of the 2004 IEEE International Conference on Shape Modeling and Applications*, pp. 167–178. Genoa, Italy, 2004.

[139] Svetlana Kiritchenko, Stan Matwin, and A. Fazel Famili. "Functional annotation of genes using hierarchical text categorization". In: *Proceedings of the 2005 Workshop of the BioLINK Special Interest Group: Linking Literature, Information and Knowledge for Biology*, pp. 1–4. Detroit, MI, 2005.

[140] Andrea Vedaldi. *A MATLAB wrapper of SVM^{struct}*. http://www.robots.ox.ac.uk/~vedaldi/svmstruct.html. 2011.

[141] Patrick Lucey, Jeffrey F. Cohn, Takeo Kanade, Jason Saragih, et al. "The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression". In: *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition – Workshops*, pp. 94–101. San Francisco, CA, 2010.

[142] Koby Crammer and Yoram Singer. "On the algorithmic implementation of multiclass kernel-based vector machines". *Journal of Machine Learning Research*, 2, pp. 265–292, 2001.

[143] Subramaniam Jayanti, Yagnanarayanan Kalyanaraman, Natraj Iyer, and Karthik Ramani. "Developing an engineering shape benchmark for CAD models". *Computer-Aided Design*, 38(9), pp. 939–953, 2006.

[144] Walter Wohlkinger and Markus Vincze. "Ensemble of shape functions for 3D object classification". In: *Proceedings of the 2011 IEEE International Conference on Robotics and Biomimetics*, pp. 2987–2992. Karon Beach, Phuket, 2011.

[145] Radu Bogdan Rusu, Gary Bradski, Romain Thibaux, and John Hsu. "Fast 3D recognition and pose using the viewpoint feature histogram". In: *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2155–2162. Taipei, Taiwan, 2010.

[146] Xu-Lei Wang, Yi Liu, and Hongbin Zha. "Intrinsic spin images: A subspace decomposition approach to understanding 3D deformable shapes". In: *Proceedings of the 5th International Symposium on 3D Data Processing, Visualization and Transmission*, pp. 17–20. Paris, France, 2010.

[147] Federico Tombari, Samuele Salti, and Luigi Di Stefano. "Unique signatures of histograms for local surface description". In: *Proceedings of the 11th European Conference on Computer Vision*, pp. 356–369. Heraklion, Greece, 2010.

[148] Serge Belongie, Jitendra Malik, and Jan Puzicha. "Shape matching and object recognition using shape contexts". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4), pp. 509–522, 2002.

[149] Radu Bogdan Rusu and Steve Cousins. "3D is here: Point cloud library (PCL)". In: *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, pp. 1–4. Shanghai, China, 2011.

[150] Thomas Serre, Lior Wolf, Stanley Bileschi, Maximilian Riesenhuber, and Tomaso Poggio. "Robust object recognition with cortex-like mechanisms". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3), pp. 411–426, 2007.

[151] Ulrich Weidenbacher and Heiko Neumann. “Extraction of surface-related features in a recurrent model of V1-V2 interactions”. *Public Library of Science: ONE*, 4(6), e5909, 2009.

[152] Antonio J. Rodríguez-Sánchez and John K. Tsotsos. “The roles of endstopped and curvature tuned computations in a hierarchical representation of 2D shape”. *Public Library of Science: ONE*, 7(8), e42058, 2012.

[153] Antonio J. Rodríguez-Sánchez and John K. Tsotsos. “The importance of intermediate representations for the modeling of 2D shape detection: Endstopping and curvature tuned computations”. In: *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4321–4326. Colorado Springs, CO, 2011.

[154] Antonio J. Rodríguez-Sánchez, Sandor Szedmak, and Justus H. Piater. “SCurV: A 3D descriptor for object classification”. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1320–1327. Hamburg, Germany, 2015.

[155] Li Fei-Fei, Robert Fergus, and Pietro Perona. “One-shot learning of object categories”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4), pp. 594–611, 2006.

[156] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. “Attribute-based classification for zero-shot visual object categorization”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3), pp. 453–465, 2014.

[157] Clémentine François, Thomas Hoyoux, Thomas Langohr, Jérôme Wertz, and Jacques G. Verly. “Tests of a new drowsiness characterization and monitoring system based on ocular parameters”. *International Journal of Environmental Research and Public Health*, 13(2), p. 174, 2016.