A SYSTEM FOR VIDEO-BASED ANALYSIS OF FACE MOTION DURING SPEECH

Christian Kroos

München 2003

A SYSTEM FOR VIDEO-BASED ANALYSIS OF FACE MOTION DURING SPEECH

Christian Kroos

Dissertation
an der Fakultät für Sprach- und
Literaturwissenschaften
der Ludwig–Maximilians–Universität
München

vorgelegt von
Christian Kroos
aus München

München, den 10. Oktober 2003

Erstgutachter: Prof. Hans Günther Tillmann
Zweitgutachter: PD Florian Schiel
Tag der mündlichen Prüfung: 16.02.2004

Preface

The most popular photography ever in the history of the *National Geographic* magazine is not a splendid rugged mountain scenery or a breath-taking underwater wild-life shot. No, it is a face, that of Afghan child refugee Sharbat Gula shot in 1984 by Steve McCurry in a refugee camp in Pakistan.[1]

What is so special about the human face that gives it this priority over all other images, makes it the prime object for front pages, advertisement, politics and not least human relationships? Surely,

> The face provides a plethora of social information about an individual's gender, age, familiarity, emotional expression and potentially their intentions and mental state. (Emery, 2000)

But the knowledge of that alone does not seem to help much to really appreciate the role of the human face in direct personal interactions. It might be that the face is just too predominant in everyday life, omnipresent as it is, and that most of all it is taken for granted - true likewise for its static as well as for its dynamic aspects. Thus the full importance of facial expression becomes only evident - painfully - with its loss. Jonathan Cole describes in his book 'About face' (Cole, 1998) patients suffering from a rare neurological condition called Möbius syndrome in which 'people are born without any ability to move their faces and so cannot make facial expressions', and the devastating impact it has on their everyday life.

Notice that N. J. Emery in the quote above does not mention the enhancement in intelligibility that is gained by watching a talker's face during speech. Maybe as a low level function it does not really qualify as social information; however, auditory-visual integration in speech allows undoubtedly fascinating insights into the way the human brain works. The video-based face motion measurement system developed in this thesis is dedicated to this particular aspect of face motion and hopefully will provided a versatile tool for research in auditory-visual speech processing.

The thesis originated in an invitation by Eric Vatikiotis-Bateson (now Professor at the University of British Columbia, Canada) to come to Japan and work in his excellent auditory-visual speech production research group at ATR Laboratories (Kyoto). So my gratitude goes foremost to him and to Professor Hans Günther Tillmann, who was so open-minded as to accept and actively support this slightly unconventional topic for a doctoral thesis in his phonetic sciences department.

I also wish to thank for active or passive support, for comments or for simply being a subject in one of the experiments, Phil Hoole, Takaaki Kuratate, Kevin Munhall, Ales Ude, Daniel Callan, Akiko Callan, Saeko Masuda, Roberto Ejii Nawa, Michiko Inoue, Hartmut Pfitzinger, Klaus Härtl, Selime Altinbilek, Jeff

[1] See `http://www.melia.com/ngm/0204/feature0/`

Jones, Shinji Maeda, Gèrard Bailly, Steven Greenberg, Kyoshi Honda, Dennis Burnham and Oyunaa Shagdar.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> He was only demonstrating certain basic pronunciation patterns but the transformation in his face and voice made me think he was making a passage between levels of beings.
>
> *Don DeLillo* ***White Noise***

1.1 Why speech?

The overwhelming majority of research resources - and subsequently publications - concerning face motion has so far been dedicated to the analysis and synthesis of *emotional facial expression* and almost marginalised the processing of speaking faces. The balance was shifted slightly in favour of a more profound investigation of visual aspects of speech with the advent of a new interest in 'multi-modal' human-human or human-machine communication both by the academic world and by industry. Again, however, the main emphasis was put on the expressive facial behaviour accompanying speech instead of the genuine speech movements of the facial surface. Only the surge of computer-aided animated human or human-like characters in advertisement, entertainment and as impersonations of computer help agents or human users in virtual reality (avatars) has made realistic 'lip synchronous' face motion during speech a primary concern for some research groups.

In 1991 K. Mase and A. Pentland, two well known senior researchers (the latter being one of the outstanding figures in the emerging science of computer vision) wrote in the summary section of Mase and Pentland (1991), an often cited paper:

> The velocity of lip motion may be measured from optical flow data which allows muscle action to be estimated. Pauses in muscle action result in zero velocity of the flow and are used to locate word boundaries.

Their study does not comprise any experiment with direct measurement of muscle activity (for instance using ***E****lectro****m****yo****g****raphy* (EMG)) to proove their claim. Nevertheless they state later about lip muscle activity during speech:

> There is a stop at word boundaries even in continuous speech. Since the stop results in zero velocity of lip movements, it is easy to find word boundaries by looking for instances of zero velocity.'

Everyone who took pains to look at articulator movements (jaw, tongue or lips) knows that this incorrect: an undergraduate in phonetic sciences would probably fail her or his exam with such remarks having been advised of the contrary in first-semester courses. Careful reading of their paper shows that all of the authors' 'insights' are based on a single experiment with recorded data consisting of

> 'continously spoken utterances comprising three to five digits'.

This admittedly extreme example demonstrates the lack of knowledge of speech in the (machine) vision field at that time.

On the other hand the majority of speech scientists tended to avoid the visual aspects of human speech regarding them at best as useful byproducts. One of the exceptions has been the research concerning the so-called *McGurk-Effect* (McGurk and MacDonald, 1976), even though the discovery itself and many of the following studies were accomplished by psychologists and not speech scientists. The lack of interest in *auditory-visual* speech by, say, linguists and phoneticians might be understandable:

Firstly, there is the obvious question regarding the relevance of auditory visual speech. Since every human being without hearing impairment can perfectly well understand any possible utterance without looking at the speaker's face, one might see no interesting role auditory visual speech could play in research or the development and design of end-user applications. Secondly, adopting an even more pragmatic view one might just point to the heap of yet unsolved problems in the auditory/acoustic domain and merely shrug at the prospect of multiplying these problems by adding a new communication channel.

Additionally it seems not very likely that the visually observable face movements during speech might be anything else than just a simple mechanic consequence of some of the movements of the (mostly hidden) speech articulators that are governed by the human brain with the aim of producing a desired *acoustic* output. Since here is not the place to enter the discussion about the nature of phonetic targets, we only would like to point out that the hypothesis assuming *articulatory* phonetic targets would make an independent significant role of the visual speech more likely - if, quoting Raymond Herbert Stetson,

> Speech is rather a set of movements made audible than a set of sounds produced by movements (Stetson, 1951),

why then should they not be made visible as well whenever possible? Bypassing this dispute and assuming that speech face movements are indeed only a mechanical consequence as stated above, the question remains whether or not the human brain uses them and their relationship to the speech articulator movements. Looking at it in terms of evolutionary development it seems doubtful that a capability would evolve whose gain would almost completely exhaust itself in adding redundancy to an already close to perfect working other ability. Since to our current knowledge speech must have evolved during a period where our human ancestors lived in small groups in 'hunter and gatherer' societies, all speech communication was limited to person-to-person short-distance communication, where visual perception of the speaker was important and frequent, but not at all

a necessary or even dominant condition (the speaker might have turned away or be in darkness).

But what about the McGurk-Effect? The effect, named after one of the two psychologists who discovered it (McGurk and MacDonald, 1976), describes the change in perception of an auditory stimulus depending on synchronous perception of a concordant or discordant visual stimulus: If visually presented /ga/, usually a video clip showing the speakers face in a close-up, is combined with an aligned audio signal of /ba/, subjects perceive /da/. Of course this does not happen with a concordant audio signal. It is particular striking on an intuitive level, that if one looks at McGurk-stimuli repeatedly and closes one's eyes in between for a short time, the auditory perception switches back and forth between /da/ and /ba/ without any feeling of a disruption. So clearly there is *some* auditory-visual integration.

It is often argued against the relevance of the McGurk-Effect that it is highly artificial. And despite the fact that it can be demonstrated live without technical means (Kevin Munhall, personal communication), it is clear that occurrences in non-technological societies are extremely rare and virtually devoid of significance. However, this does not diminish its value as a tool for investigating (a special case of) auditory-visual integration. Though likely, it cannot be determined for sure whether or not the McGurk effect is speech-specific or just a more or less accidental consequence of a general auditory-visual integration mechanism that allows us to visually locate and distinguish sources of sounds even if the sounds emanate from the same direction. That this integration even 'overrides' the auditory location of sound sources can be confirmed easily on a non-scientific level by viewing a movie with a single loudspeaker that is not placed directly behind the screen.

The strongest evidence for speech-specific auditory-visual integration might come from numerous studies documenting that being able to look at the speaker's face significantly enhances overall intelligibility in noisy environments (Sumby and Pollack, 1954). The amount of additionally recovered phonetic information, when the speaker's face is visible, is substantial and the effect is not limited to single phonemes, making it difficult to dismiss it as a byproduct of a general auditory-visual integration mechanism. Furthermore studies investigating brain activity using ***f****unctional* ***M****agnetic* ***R****esonance* ***I****maging* (fMRI) show that speech related areas in the 'auditory cortex' are activated when the stimuli presented to the subject contain video capture of a speaking face only and no sound. For instance Callan, Callan, and Vatikiotis-Bateson (2001) state:

> It is interesting to note that consistent with other studies investigating silent speechreading ... that the visual only condition showed activity in superior and middle temporal areas, including the primary auditory cortex, ... , even though there was no auditory signal.

Now, if there happens to be speech-specific auditory-visual integration it would be worthwhile to look in acoustic/articulatory signals and face/head motion measurements for redundancies. And in fact this has been done in several studies at least up to the point of determining (multiple) correlations between the different groups of signals (e.g. Yehia, Rubin, and Vatikiotis-Bateson, 1998; Jiang, Alwan, Bernstein, Keating, and Auer, 2000) and using them for mutual prediction. The correlations were found to be high and the predictions usable to reconstruct a quite close approximation to the original signal so long as the analysis/synthesis was confined to a small set of sentences or even limited to a

single sentence approach. Of course there are a lot of applications that potentially could be built on a strong acoustic-to-head/face motion relationship: low bandwidth coding of speaking faces for e.g. video conferencing, text-to-audio-visual speech synthesis systems for animated human-like characters, supplementary information for acoustic-based ***A**utomatic **S**peech **R**ecognition* (ASR) systems especially in situations where there is strong background noise (particularly other speakers) but capture of the speakers face is available. As an example for the latter think of automated information terminals in airports, train stations, etc.

The doctoral thesis at hand, however, was primarily motivated by a different aspect: if the human brain uses a speech specific auditory-visual integration mechanism to extract phonetic information from the face, then this process should tell us a lot about speech processing in general, even if it is used only in situations where the acoustical signal is degraded. Interesting questions are: What kind of information from the face is used? In which way is it used? What are its very own dynamics? Are there categories and categorical perception? Since visual perception has different properties concerning e.g. spatial and temporal resolution as compared with auditory perception, investigating speech transmitted via this channel should make further reaching insights not only into speech perception but also into production possible.

1.2 Why video?

Clearly, any research in this area would be severely limited so long as there is no means to *measure* face motion. Perceptual studies are possible without it, but both controlling the stimuli and further interpreting the results remains difficult. However, several systems exist that track face motion by means of active or passive markers placed directly on the face. OPTOTRAK (Northern Digital, Inc) is an example for an active marker system using infrared LEDs that have to be connected to a device that generates a strobe electric pulse and have to be visible for the three tracking cameras. ELITE (BTS, Milan, Italy) and QUALISYS (Qualisys Medical AB) are examples for passive marker systems using reflective, usually spherical, markers and a set of video cameras.

By contrast, reliably measuring the motion of an object without markers is much more difficult. In general, markerless techniques for measuring visible objects have been optical using simple video or structured light (e.g., Carter, Shadle, and Davies, 1996). Despite the low temporal resolution (typically 25 or 30 Hz.), the system must first find the object in a video image, identify the features to be measured, and then convert those measures from image pixels to a more physically meaningful coordinate system. Time-consuming initialisation and intense post-processing is usually necessary to identify and extract detailed measures of face motion relevant to speech analysis.

Thus marker-based systems seem to have the critical advantages of being spatially very accurate, having sufficient temporal resolution, and returning instantly accessible and processable data. But they have some unfavourable limitations, too:

i. Such equipment is highly specialised and cannot be used outside the laboratory, thus restricting the scope of applications.

ii. Even within the laboratory, the method is invasive in that markers must be attached to the subject's skin.

iii. Some of the mentioned methods interfere with other systems providing measurements that might be considered crucial to be acquired simultaneously, e.g., OPTOTRAK sensors and wires cause disturbances in electromagnetic measurement systems (EMA) that could be used to record vocal tract data together with the face motion data. And the other way round, reflection at the plexiglass helmet used in EMA causes problems in most optical systems for face motion measurement.

iv. Marker placement requires *a-priori* decisions about proper measurement locations. Since it is usually not feasible to vary marker location systematically in different runs of the experiments (simply because of the sheer number of runs necessary), marker placement can restrict or bias subsequent analysis.

v. Although perception experiments using the measurement data directly are possible and very useful (so-called *point light display* experiments), use of still images or video sequences acquired simultaneously with the measurement data for perception experiments is not feasible because of the uncontrolled influence of the visible markers on human perception.

All these limitations would be overcome by a method that could derive reliable globally distributed face motion measurements from standard video recordings. Failing automatised methods in the past many researchers turned to manually analysing face motion usually based on Ekman and Friesen's ***F****acial* ***A****ction* ***C****oding* ***S****ystem* (FACS, see Ekman and Friesen, 1978). Despite its usefulness for a number of studies (e.g., Heller and Haynal, 1997) the disadvantages of the procedure are too severe for it to be helpful in analysing speech: The investigators need to be thoroughly trained beforehand and then have to examine every single frame to determine the changes of the so-called ***A****ction* ***U****nits* (AU). The AUs represent the (visible) impact of underlying muscles or muscle groups of parts of the face. Thus in addition to being extremely tedious the resulting measurement is still qualitative. So it may not be surprising that to our knowledge FACS has not been used for analysing auditory-visual speech at all.

On the other hand attempts at automating video-based face motion measurement face an uphill struggle against the inherent general difficulties of *image motion estimation* (see 2.1) that are compounded by the specific nature of the object to be measured: face motion is in principal non-rigid motion with an underlying rigid component (i.e., jaw movements) modified in its overall appearance from a static viewpoint by rigid head motion. The reflectance characteristics (*albedo* and *specular content*) of the different parts of the face are the only entities on which a measurement algorithm can be built, but they cannot be accessed directly with a video camera. The recorded signal, the video image, represents a non-linear mapping of the three-dimensional real world to the two-dimensional image plane (perspective projection, see 2.2). The intensity value of each pixel (the only available variable besides pixel location) is a mixture of the above mentioned reflectance characteristics of the object, the scene illumination, the viewing geometry, and the camera response parameters - integrated over space (a subset of light rays passing through the camera lense) and time (shutter time). The resolution both in space and time is of course limited by the image acquiring mechanism, usually a ***C****harged* ***C****oupled* ***D****evice* (CCD) chip.

Those difficulties explain why video-based face motion tracking methods that return measurements distributed globally over the face are so far rather rare. Instead most researchers employed feature tracking methods focusing usually on the lips, the eyes, the eyebrows and already less frequently including the whole

chin or the alar wing of the nose. It should be clear that video-based methods will not be able to compete with marker-based methods in terms of resolution and reliability, at least not in the near future. Nevertheless when looking at the benefits of video-based face motion analysis the advantages seem to outweigh the limitations:

i. Since the constraint of recording in the lab is no longer confining the recording, it can be done anywhere where a video camera can be used and at any time. In addition, continuous recording time is only limited by the camera's storage capabilities.

ii. With the freedom of choice of space and time for the recording and the no longer existing necessity of marker attachment a whole new variety of situations is made accessible for analysis reaching from quasi-spontaneous dialogs in the lab to authentic TV news coverage and documentaries. With respect to the main principle of a *New Phonetics* - to abandon 'lab speech' in phonetic analyses in favour of spontaneous or at least quasi-spontaneous speech - this would be highly desirable.

iii. A video-based method would make it easier to measure e.g. the vocal tract behaviour at the same time, since the only condition needed is an unimpeded optical connection between the camera and the subject's face. Reflections do not interfere with the tracking process in a significant way.

iv. Any decision about the use of only a subset of analysis points (virtual markers) could be done *a-posteriori*, according to the needs of the particular analysis type or even based on criteria coming from processing the full set of tracking points in a first run.

v. If the measurement points are truly globally distributed over the face, animations based on the results can be accomplished. This could lead to means for perfect stimulus control and verification in auditory-visual speech perception experiments.

Potential applications for video-based face tracking include acoustic speech synthesis from talking faces (Yehia et al., 1998), realistic face synthesis and animation (Kuratate, Yehia, and Vatikiotis-Bateson, 1998), more accurate analysis of human-machine interfaces, and many clinical applications where accurate measures of functional behaviour of the face are required (for details, see Ekman and Rosenberg, 1997). However, it cannot be emphasised enough, that great care is needed given the complexity of both speech and faces and the difficulties involved in image motion estimation.

1.3 Point of view

In this section we briefly lay out the assumptions being made in the development of the face motion tracking algorithm described in this thesis before clarifying in chapter 2 the role different techniques and procedures originating from distinct scientific domains play in working towards a face tracking algorithm.

1.3.1 Video footage

First of all we assume a static camera directed towards the subject's face with no change in the focal length (i.e. zoom in or out) occurring during the tracking period. The algorithm could probably be adapted to a situation with a moving camera, but this would be beyond the scope of this thesis. Furthermore the face should not appear too small in the video frame in order to have enough pixels left in the segments into which the facial surface will be divided by our algorithm and which will be tracked independently. Somewhere in the entire sequence there should be a frame with an (almost) full frontal view of the face needed for the initialisation.

Of course movements of the subject are not restricted, but - trivially - face motion tacking is not possible when the subject's face is completely turned away from the camera and already not making much sense with the current setup when the subject's head is approaching a profile view. Head motion is tracked using OPTOTRAK and a headmount; in a future version it might be replaced by an integrated video-based method.

It is assumed that there are no occlusions of the facial surface (except occlusions occurring due to head motion); but again, the algorithm might be adapted to cope with temporary occlusions if necessary. The method can handle colour as well as gray-scale image sequences. In fact, currently colour images will be converted to gray-scale before the tracking. This is done mainly in order to speed up the slow tracking procedure, but in principle the three colour channels of RGB-images could be separately processed and the tracking results combined in an optimal way afterwards.

1.3.2 Constraints of the facial surface

Our explicit aim was to avoid any high level assumptions about face motion, especially speech-specific assumptions, for instance, that the lip region must be tracked with a higher resolution than the cheeks because of their supposed importance in visual speech. After all, the whole face motion analysis system described herein was built to investigate what kind of role the different parts of the facial surface play (or can play) in auditory-visual speech processing. Low level constraints, however, are of crucial importance because of the already mentioned inherent difficulties and ambiguities of image motion estimation.

In the case of speaking faces, obtaining accurate motion measures is hindered by a combination of factors: 1) the lack of strong *image gradients* (see also 2.1.1) for the cheeks and chin, 2) the diversity of movement types caused by simultaneous rigid (head and jaw) and non-rigid (face) motions on different scales, and 3) the high velocity of some movements such as mouth opening and closing for transitions between low vowels and bilabial stops. However, workable solutions can be found by using the physical properties of the face as task-specific constraints. The face structure combines rigid jaw and skull components with a non-rigid covering of facial tissue. Although complex, the relation between these components is constrained. In particular, looking at face motion from the camera's perspective, it is possible to analyse it as a combination of rigid translation and rotation of the entire head and non-rigid deformations of facial tissue. The non-rigid deformations result from the contraction of muscles attached to the fascia and from the motion of the jaw. The fascia itself has dynamic tissue properties that passively filter the influence of the musculoskeletal system. Specific activities such

as speech can further influence the shape of the facial surface through changes in tongue position and air pressure within the oral cavity (see Yehia et al., 1998).

Once identification and correction of head motion effects on apparent face motion are accomplished, four assumptions can be made about the physical properties of the face and its motion from the view point of a static camera looking at the frontally displayed face.

i. The face surface is in general continuous, so its parts cannot be dissociated from each other;

ii. There is no image-plane rotation for larger areas of the face;

iii. Larger movements always affect larger regions;

iv. Since occlusion due to head motion can be predicted easily, the visible areas of the face are known, provided they are not very small. That is, they cannot appear or disappear; rather they can only extend, contract or become distorted. The exception to this are the areas within the opened mouth and the eyes that are occluded completely during mouth closure and eye blinking.

These assumed properties, especially *i.* and *iv.*, suggest that the texture map of the face in the video frame can be used to measure the location of facial segments, provided that the image intensity does not change much from one frame to the next (this assumption is critical to other methods such as *optical flow*, see section 2.1.1).

The implication of *iii.* is that a *multi-resolution analysis* (see section 2.3) can be applied, starting at low resolution, without introducing errors into the subsequent finer-grained analyses. It also takes care of the exceptions to *iv.* because of the surface continuity constraint *i.*. To explain this we have to anticipate the motion tracking section 3.3 for a moment: The basic idea is that by increasing the resolution of the tracking procedure in a stepwise manner, it should be possible to track the motion of large areas separately from motions of smaller areas. For example, large motions due to the jaw must be distinguishable from the small motion of the lower lip that may move independently of the jaw in the opposite direction. Thus, by breaking the analysis down into a sequence of resolution-specific analysis steps, then the loss of detailed texture information at a lower resolution, or spatial frequency, defines a natural motion region that can be refined in a subsequent step at higher resolution.

Finally the constraint on image-plane rotation, *ii.*, allows us to avoid relatively cumbersome, orientation sensitive filters such as *Gabor wavelets* or the so-called *steerable filters* (Freeman and Adelson, 1991) in the multi-resolution analysis of the image data. These two-dimensional filters can be designed with arbitrary orientations, while the wavelets used currently in the tracking are fixed to horizontal, vertical and diagonal orientation (see section 2.3.4). However, the constraint assumption might be violated for certain small areas of the facial surface, e.g., the corner of the mouth might undergo quasi-rotational movements during mouth opening and closing. The coarse-to-fine strategy in the tracking in combination with the segment warping process described in section 3.3.5 and the wavelet-based image decomposition can in general catch these exceptions. However, the exclusion of orientations sensitive filters is motivated only by run-time consideration: All correspondence determination procedures (described later in section 3.3.5 and 3.3.6) must be repeated for every selected orientation which currently would slow down the motion tracking too much. Nevertheless - in anticipation

of faster computers in the near future - a steerable filter based on the second derivative of Gaussian (so-called *Mexican hat*) is already implemented in the current version of the algorithm.

Chapter 2

Theoretical and empirical basis

2.1 Image motion estimation

Usually there will be temporal variations of image brightness in an image sequence caused by *image motion*. In natural images image motion is induced by the movement of three-dimensional scene points relative to the camera. The projection of the tree-dimensional velocity field of scene points to the image plane results in a two-dimensional *motion field*. This might be due to the movement of objects in the scene or movements of the camera (including changes of the focal length) or both combined. Since image sequences, for instance video sequences, are time-discrete, continuous 'real world' trajectories are sampled into time-varying discrete changes of image coordinates of the respective image points. Depending on the shutter time of the camera during the recording a shorter or longer time period is integrated into a single image.

In contrast to the A/D conversion of continuous sound pressure variations into a sampled digital sound signal via microphone and digitising equipment and the D/A conversion of the sampled signal back into continuous sound waves via D/A converter and loudspeaker, the video image remains discrete when played back. All motion a human observer perceives, when an image sequence is presented with a sufficiently high frame rate, is *apparent motion*. The perceived continuous trajectories or discontinuous 'jumps' of image elements are a (re-)construction of the human eye and brain (see Marr, 1982; Wandell, 1995; Palmer, 1999). The insight has created one of the most fruitful paradigms for visual perception research, since very basic stimuli could be conceived and realised (e.g., a black square on a white background which changes position from one frame to the next) and systematically varied.

This introduces a fundamental problem of *image motion estimation*: the *correspondence problem*. How can be determined to which new location the scene objects (represented in the image as brightness variation) moved in a frame-to-frame transition? In other words, which image object in the incoming frame n corresponds to which image object in the previous frame $n-1$. Suggestions made for human perception as well as solution attempts in machine vision include: the closest one, the most similar one concerning shape or texture map values, the one in the direction of the motion of the object determined in earlier frames, any combination of those already mentioned, and so on. Note that the human visual

system has excellent capabilities to solve the correspondence problem - for example a grayscale video sequence of swirling leaves in the wind does not throw our motion perception into disarray. Or as shown in perception experiments, transparent motion of two planes of random dots could still be resolved correctly.

Before presenting two main strategies to determine image motion in the next section a short excursion is necessary to address technical difficulties when working with standard video sequences in speech face motion tracking. Evidently the correspondence problem is easier to tackle if the displacement of image objects from one frame to the next is very small. Despite the fact that the average velocity of observable speech movements is relatively low compared to the standard frame rate, there are - as already mentioned - some very fast movements most of the times involving combined jaw and lip movements. Standard video frames are composed of two half frames, also called *fields*, that are acquired successively. They consist of every other line of the full video frame, alternating between odd and even lines. After the acquisition each two of the fields are interlaced into the full frame. This leads to severe artifacts for frames containing fast motions; imagine for example the area of the mouth in a fast opening gesture: every other line may contain the dark area inside the mouth, while the remaining ones still have intensity values corresponding to the lip and chin texture. Note that low pass filtering applied during preprocessing in almost all tracking methods does not solve the problem: the averaging effect of it still corrupts the intensity values of the area in question even though the sandwich-like artifact structure on line level has disappeared.

Full frames can be decomposed into their fields again later leading to an increase of the frame rate to 50 Hz (PAL) or 60 Hz (NTSC) in the analysis. This would facilitate the tracking of fast speech movements considerably, if it was not traded for half of the vertical resolution available with a one line offset every other field. The decision whether to use frames or fields can only be made according to the actual tracking task and the properties of the video footage. For instance in our speech face motion tracking we would only decompose frames into fields if the height of the subject's face is at least half of the frame height. But even with decomposition into fields the standard shutter time of 1/60 second results every now and again in blurring of parts of the fields thereby eradicating almost completely the original texture map of that area.

2.1.1 Image flow and image registration

Very generally speaking there exist two alternative ways of image motion estimation for tracking real world objects in a frame-to-frame transition:

i. First determine where each pixel moved, which includes knowing which pixels disappeared or appeared. Then infer from the dense pixel-based motion field the image motion of the real world objects, that is, the two-dimensional projection of the three-dimensional trajectories of the scene objects. If required, infer the three-dimensional trajectories from the image motion using prior knowledge about the objects.

ii. First generate an *appearance model* of the scene objects, which usually means a description of the geometrical shape and the texturing of the objects together with a model of their appearance change due to all kinds of admissible/expected motion. The latter could for instance include a model of the scene lighting. Then generate the appearance of the object over the

entire motion parameter space and use an appropriate distance measure to compare the generated image (or part of the image) with the original. The parameter set that produces the minimum distance describes the motion of the particular object. Since the parameter search space might be high-dimensional and infinite, almost always some kind of *optimisation method* is used to limit the number of views required to be generated to a small subset of the set of potential views.

The first class of methods (*i.*) uses *image flow* (also called *optical flow*) techniques, the second class (*ii.*) is known as *image registration*. For an example of the use of constrained image flow in full body tracking see Ju, Black, and Yacoob (1996), and for an example of image registration using an appearance model in full body tracking see Ude (2001). There are, of course, all kinds of hybrid approaches.

Image flow techniques (see Barron, Fleet, and Beauchemin, 1994, for an overview) assume that the image brightness $I(x(t), y(t), t)$ is constant along the visual trajectories, where $x(t)$ and $y(t)$ are spatial coordinates of the image and t denotes time. Formulated as a differential equation this is:

$$\frac{dI}{dt} = 0 \tag{2.1}$$

Applying the chain rule for differentiation equation 2.1 becomes

$$\frac{dI(x(t), y(t), t)}{dt} = \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0 \tag{2.2}$$

Equation (2.2) can be rewritten as

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = (\nabla I)^2 \boldsymbol{v} + \frac{\partial I}{\partial t} = 0 \tag{2.3}$$

where $\boldsymbol{v} = (u, v)$ is the velocity consisting the two velocity components of the motion field (change in horizontal and vertical direction) and ∇I is the spatial gradient. Since there is only one scalar constraint to determine the two velocity components, this is an *ill-posed* problem. It can only be solved by adding additional constraints, usually by assuming a certain degree of smoothness of the motion field often both in spatial and in temporal respect. Thus the actual computation is usually based on larger areas, not on single pixels, and sometimes needs to consider several successive frames (up to 64). Observe that the approach requires differentiable trajectories, thus even in the discrete form will look only at very local surroundings for the continuation of a movement. To allow bigger movements propagation and refinement of image flow values from coarse to fine scale image representations are necessary.

Image registration techniques provide in our opinion a greater robustness and a more 'realistic' overall scenario for high level vision. This is traded for a higher computational effort and a greater task dependence. In practise, the methods often suffer badly from the simplifications that have to be made in the model building process. The algorithm proposed in this thesis applies image registration.

We want to conclude this section with a quote from Stiller and Konrad (1999), which was published in 1999, but is still valid (even if the real-time requirement is relaxed):

Although the understanding of issues involved in the computation of motion has significantly increased over the last decade, we are still far from generic, robust real-time motion-estimation algorithms.

2.1.2 Video-based head and face motion tracking

Since the human head is a rigid structure, its motion can trivially be only rigid body motion. This makes head tracking easier, but by no means easy. Problems posed by varying illumination are further aggravated by the fact that in video sequences face motion, especially jaw movements, influences the appearance of the head. Only because the magnitude and the size of the affected area of the coherent parts of face motion are usually small compared to the remaining unaffected rigid parts, rigid body constraints can be used effectively in the tracking. Basu, Essa, and Pentland (1996) used a three-dimensional ellipsoidal model of the head, a parametrised but then resampled mesh, and interpreted the optical flow in terms of the possible rigid motions of the model. La Cascia, Isidoro, and Sclaroff (1998) employed a cylindrical surface model of the head and projected the video image into the surface texture map of the model. Model parameters were updated 'via robust image registration in the texture map space'. The most sophisticated head model, a commercially available polygon model consisting of about 7000 triangles, was used by Schödl, Haro, and Essa (1998). The rendered image of the textured model was registered with the video images and the motion parameters were found by 'mapping the derivative of the error with respect to the parameters to intensity gradients in the image'.

As mentioned in the introduction face motion tracking has attracted remarkable attention in the last decade mainly with respect to automatic recognition of emotional face expressions. The exception proving the rule is the abundance of lip tracking systems that have appeared almost everywhere in recent years and that are almost always designed for speech applications. Apart from them two main approaches researchers have taken so far are methods that:

i. identify and track pre-determined facial features;

ii. track a set of measurement points globally distributed across the face.

Feature tracking methods are by far the most popular and typically use either *optical flow techniques* (see section 2.1.1) and/or a statistical constraint on the desired features, such as an *adaptive shape model* (e.g., Revèret, Garcia, Benoit, and Vatikiotis-Bateson, 1997). For example Mase (1991) used optical flow limited to pre-determined areas containing the specific facial features that they want to measure (e.g., mouth). Black and Yacoob (1997) have added higher-order constraints that restrict the shape changes of the selected areas to an affine model. Methods based on *globally distributed measurements* may also use dense optical flow (e.g., Wu, Kanade, Cohn, and Li, 1998) or as in Essa and Pentland (1997) an *optimal optical flow* method coupled to a physical model describing the skin and muscle structure of the face. Alternatively, the video texture map may be fit to a global model as has been done for head tracking. In extending their method Cohn and colleagues (Wu et al., 1998) have developed a hybrid approach that combines global measurement techniques and feature tracking (Lien, Kanade, Cohn, and Li, 1999).

2.2 Perspective transformation

Every natural image taken by a photo or video camera is the mapping of a three-dimensional 'real world' scene to the two-dimensional image plane. Light rays emitted from a light source fall either directly into the camera or are reflected from object surfaces first and then fall into the camera. The latter is the more important process for imaging and allows us to draw conclusions about spatial arrangement, shape, surface texture and other properties of objects in the scene. The mapping is called *perspective projection*, it can be modelled with the so-called *perspective transformation*. The actual parameters of the transformation depend on the camera properties, which can be quite complex to determine in modern video cameras with clusters of lenses and zoom ability. Fortunately, however, the simplest and most fundamental camera model, *the ideal pinhole* camera, is for most applications a sufficiently accurate simulation of the real imaging process.

2.2.1 The ideal pinhole camera

The principle of the pinhole camera was already realised in the *camera obscura* invented in the 16th century. The names already describe the underlying idea: a very small hole in one side of a closed black box facing the object or scene to be imaged is basically all that is needed. If the opposite site is replaced by a translucent plate a dim upside-down image can be observed. Forsyth (2003) notes:

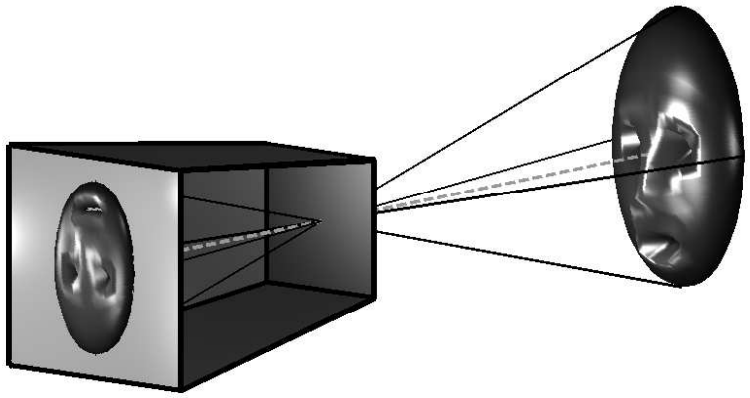
> If the pinhole were really reduced to a point (which is of course physically impossible), exactly one light ray would pass through each point in the plane of the plate (or *image plane*), the pinhole and some scene point. In reality, the pinhole has a finite (albeit small) size, and each point in the image plane collects light from a cone of rays subtending a finite solid angle ... (Forsyth, 2003, page 4, italics by the original author)

Note that with the ideal infinite small pinhole the image would be always in focus now matter how far away from the pinhole the image plane would be placed. Figure 2.1 shows a schematic drawing of this idealised model of the imaging geometry.

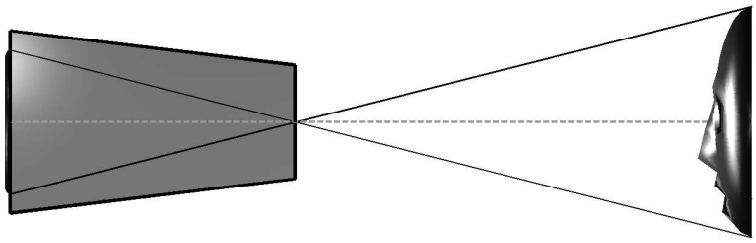
The origin of the camera coordinate system is usually assumed to lie in the *optical centre* of the lense system, i.e., in the pinhole camera approximation the hole is the optical centre. The *optical axis* is the line perpendicular to the image plane going through the optical centre. The *focal length* is simply the distance from the optical centre to the image plane and the *principal point* is the intersection of the optical axis with the image plane, usually the centre of the image plane.

If $\mathcal{P}_W$ is a scene point with coordinates $[x_W, y_W, z_W]$ and $\mathcal{P}_I$ its corresponding image point with coordinates $[x_I, y_I, z_I]$, then $z_I = f$, since $\mathcal{P}_I$ lies in the image plane, and the perspective transformation is given by

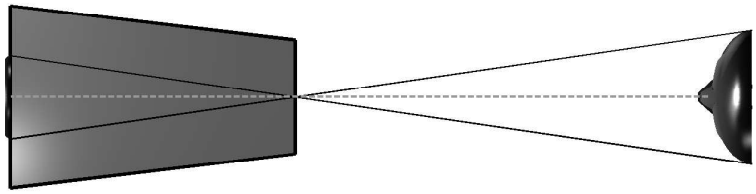
$$\begin{aligned} x_I &= f\frac{x_W}{z_W} \\ y_I &= f\frac{y_W}{z_W} \end{aligned} \tag{2.4}$$



(a) 40 degree



(b) Side view



(c) Top view

Figure 2.1: Ideal pinhole camera model

Equation (2.4) only holds if all measurements are made in the camera reference frame and share the same unit and origin, which has to be the principal point. In practice, the coordinate system for the scene points, usually the 'world coordinate' system (see section 3.1.4), and the camera coordinate system are related by a set of physical parameters including pixel size, position and orientation of the camera, etc. Therefore we will derive a more complex perspective projection matrix that accounts for these parameters in the next section.

2.2.2 Extrinsic and intrinsic camera parameters

In computer vision *intrinsic* and *extrinsic* camera parameters are distinguished. Intrinsic parameters relate the real camera coordinate system to an idealised image coordinate system. They include the principal point, scale factors for pixel size in both image dimensions, an aspect distortion factor that models the aspect ratio of the camera, focal length and a lense distortion factor that models radial lense distortion effects. These are only the more important and frequently modelled ones, but real lenses exhibit for instance nonlinear *spherical aberrations*, *coma*, *astigmatism* and *field curvature* as well.

Extrinsic parameters relate the idealised image coordinate system to a fixed world coordinate system by specifying its position via translation parameters and its orientation via rotation parameters.

We will now build a camera model which incorporates all of the extrinsic and two of the intrinsic parameters (focal length and the scale factor for pixel size) following Shapiro and Stockman (2001). It approximates the perspective projection process that happens in a real video camera such as we use in the motion tracking. Let $\mathcal{P}_W$ be again a point in the world coordinate system with coordinates $[x, y, z]^T$ and $\mathbf{P}_W$ a set of those points. Let $\mathcal{P}_I$ be the corresponding image point in the camera coordinate system, thus it is actually a pixel in the image matrix $\mathbf{P}_I$ with row and column index $[r, c]^T$ and the origin by convention in the upper left corner. In order to be able to use uniformly matrix notation we change to homogenous coordinates (see Faugeras, 1993; Shapiro and Stockman, 2001; Forsyth, 2003), i.e., $\mathcal{P}_W$ becomes $[x, y, z, 1]^T$.

2.2.2.1 Translation

$\mathbf{P}_W$ can be translated from the world coordinate system to a coordinate system having the same origin as the camera coordinate system using a translation matrix $\mathbf{T}_{W\to T}$ by

$$\begin{aligned} \mathbf{P}_T &= \mathbf{T}_{W\to T}\,\mathbf{P}_W \\ \begin{bmatrix} \mathcal{P}_{Tx} \\ \mathcal{P}_{Ty} \\ \mathcal{P}_{Tz} \\ 1 \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathcal{P}_{Wx} \\ \mathcal{P}_{Wy} \\ \mathcal{P}_{Wz} \\ 1 \end{bmatrix} \end{aligned} \tag{2.5}$$

where t_x, t_y, and t_z, are the entries for each of the three dimensions in the translation vector $\mathbf{t}$ that maps the origins of the two coordinate systems on each other.

2.2.2.2 Rotation

Rotations in three-dimensional space are possible around any arbitrary rotation axis, but it is convenient to express them as components around the three coordinate axis. They are given by the following equations.

Rotation of α about the x-axis:

$$\mathbf{P}_{R_\alpha} = \mathbf{R}_{x,\alpha}\,\mathbf{P}_W$$

$$\begin{bmatrix} \mathcal{P}_{R_\alpha x} \\ \mathcal{P}_{R_\alpha y} \\ \mathcal{P}_{R_\alpha z} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathcal{P}_{Wx} \\ \mathcal{P}_{Wy} \\ \mathcal{P}_{Wz} \\ 1 \end{bmatrix} \tag{2.6}$$

Rotation of β about the y-axis:

$$\mathbf{P}_{R_\beta} = \mathbf{R}_{y,\beta}\,\mathbf{P}_W$$

$$\begin{bmatrix} \mathcal{P}_{R_\beta x} \\ \mathcal{P}_{R_\beta y} \\ \mathcal{P}_{R_\beta z} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathcal{P}_{Wx} \\ \mathcal{P}_{Wy} \\ \mathcal{P}_{Wz} \\ 1 \end{bmatrix} \tag{2.7}$$

Rotation of γ about the z-axis:

$$\mathbf{P}_{R_\gamma} = \mathbf{R}_{z,\gamma}\,\mathbf{P}_W$$

$$\begin{bmatrix} \mathcal{P}_{R_\gamma x} \\ \mathcal{P}_{R_\gamma y} \\ \mathcal{P}_{R_\gamma z} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 & 0 \\ \sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathcal{P}_{Wx} \\ \mathcal{P}_{Wy} \\ \mathcal{P}_{Wz} \\ 1 \end{bmatrix} \tag{2.8}$$

In this way we obtain three rotation matrices depending on three parameters (the rotation angles). All possible three-dimensional rotations can be performed by applying sequentially these three matrices. In addition they can be combined in a single one by

$$\mathbf{R} = \mathbf{R}_{x,\alpha}\,\mathbf{R}_{y,\beta}\,\mathbf{R}_{z,\gamma} \tag{2.9}$$

since matrix multiplication is associative. Accordingly the orientation of the world coordinate system and the camera coordinate system can be align with:

$$\mathbf{P}_R = \mathbf{R}_{W\to R}\,\mathbf{P}_W \tag{2.10}$$

$$\begin{bmatrix} \mathcal{P}_{Rx} \\ \mathcal{P}_{Ry} \\ \mathcal{P}_{Rz} \\ 1 \end{bmatrix} =$$

$$\begin{bmatrix} \cos\beta\cos\gamma & -\cos\beta\sin\gamma & \sin\beta & 0 \\ \sin\alpha\sin\beta\cos\gamma+\cos\alpha\sin\gamma & -\sin\alpha\sin\beta\sin\gamma+\cos\alpha\cos\gamma & -\sin\alpha\cos\beta & 0 \\ -\cos\alpha\sin\beta\cos\gamma+\sin\alpha\sin\gamma & \cos\alpha\sin\beta\sin\gamma+\sin\alpha\cos\gamma & \cos\alpha\cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathcal{P}_{Wx} \\ \mathcal{P}_{Wy} \\ \mathcal{P}_{Wz} \\ 1 \end{bmatrix}$$

Translation and rotation can be integrated in to a single matrix describing rigid motion:

$$\mathbf{V} = \mathbf{T}\,\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.11}$$

where r_{ij} are the rotation coefficients from equation (2.10). This step is actually the motivation to switch to homogenous coordinates.

2.2.2.3 Projection

After we moved the world coordinate system to be aligned with the image coordinate system the projection itself can be applied. Since our underlying model for the projection process is still the pinhole camera this amounts only to rewrite equation (2.4) in matrix form:

$$\begin{aligned} \mathbf{P}_F &= \mathbf{F}_{V\to F}\,(\mathbf{V}_{W\to V}\,\mathbf{P}_W) \\ \begin{bmatrix} s\,\mathcal{P}_{Fx} \\ s\,\mathcal{P}_{Fy} \\ s\,\mathcal{P}_{Fz} \\ s \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{f} & 1 \end{bmatrix} \left(\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathcal{P}_{Wx} \\ \mathcal{P}_{Wy} \\ \mathcal{P}_{Wz} \\ 1 \end{bmatrix} \right) \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{f} & 1 \end{bmatrix} \begin{bmatrix} \mathcal{P}_{Vx} \\ \mathcal{P}_{Vy} \\ \mathcal{P}_{Vz} \\ 1 \end{bmatrix} \end{aligned} \tag{2.12}$$

The third row of the matrix can be dropped, since it only returns a constant value for $\mathcal{P}_{Fz}$,[1] and because matrix multiplication is associative we can rewrite equation (2.12) as

$$\begin{aligned} \mathbf{P}_F &= (\mathbf{F}_{V\to F}\,\mathbf{V}_{W\to V})\,\mathbf{P}_W \\ \begin{bmatrix} s\,\mathcal{P}_{Fx} \\ s\,\mathcal{P}_{Fy} \\ s \end{bmatrix} &= \begin{bmatrix} g_{11} & g_{12} & g_{13} & g_{14} \\ g_{21} & g_{22} & g_{23} & g_{24} \\ g_{31} & g_{32} & g_{33} & 1 \end{bmatrix} \begin{bmatrix} \mathcal{P}_{Wx} \\ \mathcal{P}_{Wy} \\ \mathcal{P}_{Wz} \\ 1 \end{bmatrix} \end{aligned} \tag{2.13}$$

[1] Evaluating the fourth row of $\mathbf{P}_F = \mathbf{F}_{V\to F}\,\mathbf{P}_V$ gives of course $s = \frac{1}{f}\mathcal{P}_{Vz}$ (see second half of the explicit version of the projection equation above). Accordingly evaluating the third row yields $\mathcal{P}_{Fz} = \mathcal{P}_{Vz}\,\frac{f}{\mathcal{P}_{Vz}} = f$.

2.2.2.4 Accounting for pixel size

Up till now we still have the same unit as the world coordinate system and real-valued coordinates instead of row and column index integers. Therefore another scaling is needed that at the same time should take care of the fact that image matrix rows correspond to the y-axis and columns to the x-axis as well as invert the y-axis:

$$\mathbf{P}_I = \mathbf{S}_{F\to I}\,(\mathbf{F}_{V\to F}\,\mathbf{V}_{W\to V})\,\mathbf{P}_W$$

$$\begin{bmatrix} s\,\mathcal{P}_{Ir} \\ s\,\mathcal{P}_{Ic} \\ s \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{d_y} & 0 \\ \frac{1}{d_x} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} g_{11} & g_{12} & g_{13} & g_{14} \\ g_{21} & g_{22} & g_{23} & g_{24} \\ g_{31} & g_{32} & g_{33} & 1 \end{bmatrix} \begin{bmatrix} \mathcal{P}_{Wx} \\ \mathcal{P}_{Wy} \\ \mathcal{P}_{Wz} \\ 1 \end{bmatrix} \tag{2.14}$$

where d_x and d_y are the respective scaling factors.

2.2.2.5 The final camera model

As the result of the preceeding steps we obtain the final full perspective projection matrix:

$$\mathbf{P}_I = \mathbf{C}_{W\to I}\,\mathbf{P}_W$$

$$\begin{bmatrix} s\,\mathcal{P}_{Ir} \\ s\,\mathcal{P}_{Ic} \\ s \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & 1 \end{bmatrix} \begin{bmatrix} \mathcal{P}_{Wx} \\ \mathcal{P}_{Wy} \\ \mathcal{P}_{Wz} \\ 1 \end{bmatrix} \tag{2.15}$$

2.3 Wavelets and multiresolution analysis

2.3.1 Spatial frequencies

As many authors (e.g., Burke Hubbard, 1998) have pointed out the term 'spatial frequency' is actually inappropriate, since frequency is defined for time only. Therefore it is often substituted by the term 'wave number'. In this thesis we will use the term nevertheless, because the concept of oscillation can be very easily transferred from time to space, and it seems to be somewhat more intuitive than 'wave number'. Some caution in using the term is still recommended though, since there are differences between temporal and spatial frequencies, most notably that at least in the Newton Universe time has only one direction.

Having said this we would like to ask the reader not familiar with image processing to imagine how a vertically uniform horizontal sinusoid looks in a grey-scale image. The answer can be inspected in Figure 2.2 on the facing page. Our human visual perception can hardly interprete the image as anything else than a line of vertical bars in front of a black background.[2] A 'cross section' through

[2] Please send an email to kroos@phonetik.uni-muenchen.de, if you have other suggestions

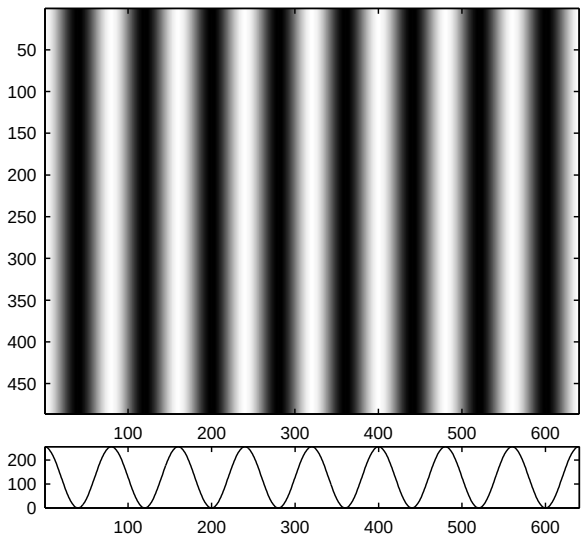


Figure 2.2: Image with an horizontally oriented sinusoid of frequency 8 c/image width (0.0125 c/p). Below the image the intensity values of a single line are plotted.

an arbitrary row, i.e., plotting the intensity values, reveals the sinusoid. The intensity values of the pixels oscillate with a spatial frequency of 8 cycles per image width along the horizontal axis. The term 'cycles' is very broadly used and commonly abbreviated to just the letter c. To specify spatial frequencies relative to the size of the image frame cannot be considered very favourable, even if the approach is intuitively appealing, since image size is a rather arbitrary quantity. Accordingly one switches in digital image processing to the smallest unit, the pixel and specifies spatial frequencies in cycles/pixel. It needs a little bit of rethinking to familiarise oneself with the fact that with this definition all relevant frequencies in the image will be smaller than 1. More precisely less than or equal to 0.5, because the *sampling theorem* holds naturally for spatial frequencies as well as for temporal frequencies. Thus the highest frequency that is contained in the image is the *Nyquist frequency*, the bandwidth of a sampled signal, equal to half the sampling frequency of that signal, i.e. 0.5 c/p in the case of a digital image.[3] The image

[3] Very irritatingly, there seem to exist two different definitions of the Nyquist frequency. For example Trucco and Verri write in Trucco and Verri (1998) on page 314:

> The frequency $\nu_c = \omega_c/\pi$, inverse of the sampling interval $T_c = \pi/\omega_c$, is named *Nyquist frequency* and is typical of the signal. It is *the minimal sampling frequency necessary to reconstruct the signal.* [italics by the original authors]

Also Eric Weisstein (Weisstein, 1999) defines the Nyquist frequency as follows:

> In order to recover all Fourier components of a periodic waveform, it is necessary to sample more than twice as fast as the *highest* waveform frequency ν, i.e.,
>
> $$f_{Nyquist} = 2\nu$$
>
> The cutoff frequency $f_{Nyquist}$ above which a signal must be sampled in order to be able to fully reconstruct it is called the Nyquist frequency. [italics by the original author]

Those definitions might not even be just idiosyncratic deviations from the more generally accepted definition presented in the main text, but stem from different introduction but interchanging use

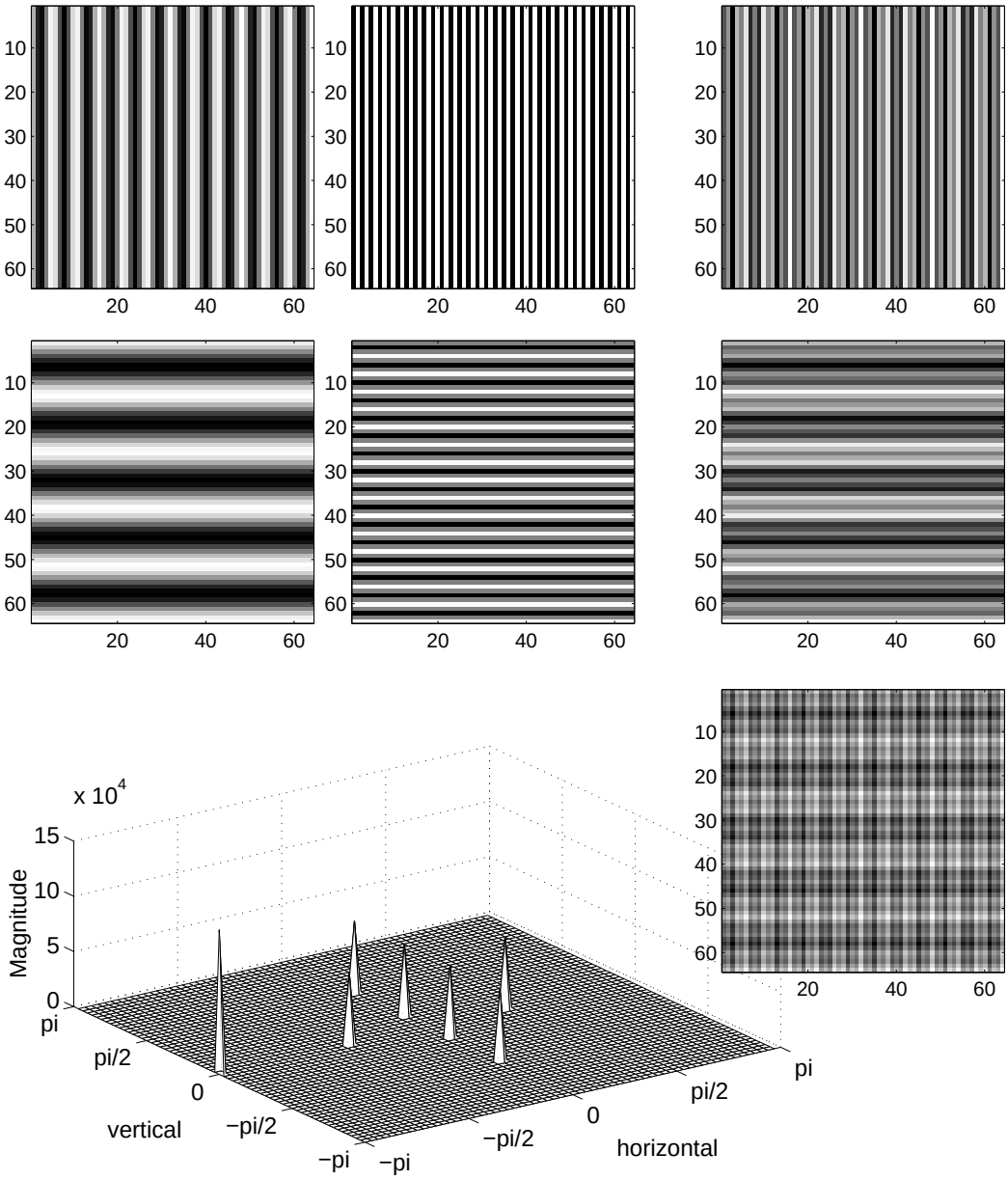


Figure 2.3: Top row: Two images (64 x 64 pixels) with horizontal sinusoids of frequency 0.1875 c/p (left) and 0.5 c/p (centre) and image of the same size that combines the two sinusoids. Centre row: Two images (64 x 64 pixels) with vertical sinusoids of frequency 0.0926 c/p (left) and 0.25 c/p (centre) and image of the same size that combines the two sinusoids. Bottom row: Image that contains all four sinusoids (right) and its power spectrum.

underlying Figure 2.2 has the size of a full NTSC video frame with square pixels, i.e. 640 pixels (width) times 486 pixels (height). That means that the frequency of

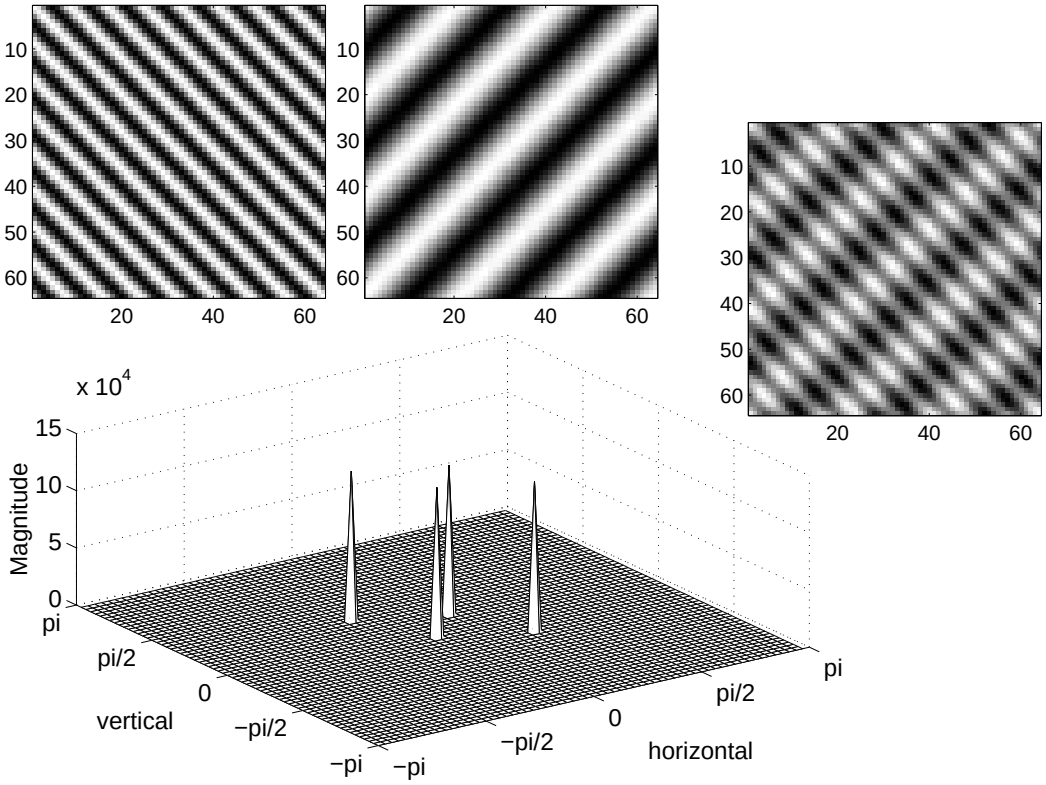


Figure 2.4: Two diagonal spatial sinusoids of frequencies 0.1768c/p (left) and 0.0663 c/p (centre), the combined image (right), and the power spectrum of the combined image (bottom).

the sinusoid is $8/640 = 0.0125$ c/p.

It goes without saying that the concept of the *Fourier transformation* can be applied to spatial frequencies in the same way as to temporal. Its one-dimensional formulation (here in the most general form) for discrete-time aperiodic signals

$$X(\omega) = \sum_{t=-\infty}^{\infty} x(t)\, e^{-i\omega t} \tag{2.16}$$

where $x(t)$ is the signal value at time sample t, ω the *angular frequency* $2\pi k$, and $X(\omega)$ represents the frequency content of signal $x(t)$, can be directly transferred to space-discrete aperiodic signals

$$F(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n)\, e^{-i\frac{2\pi k}{N} n} \tag{2.17}$$

of the terms *Nyquist frequency* and *Nyquist rate*. For example Proakis and Manolakis (1996) define *Nyquist rate* as in the two examples just mentioned.

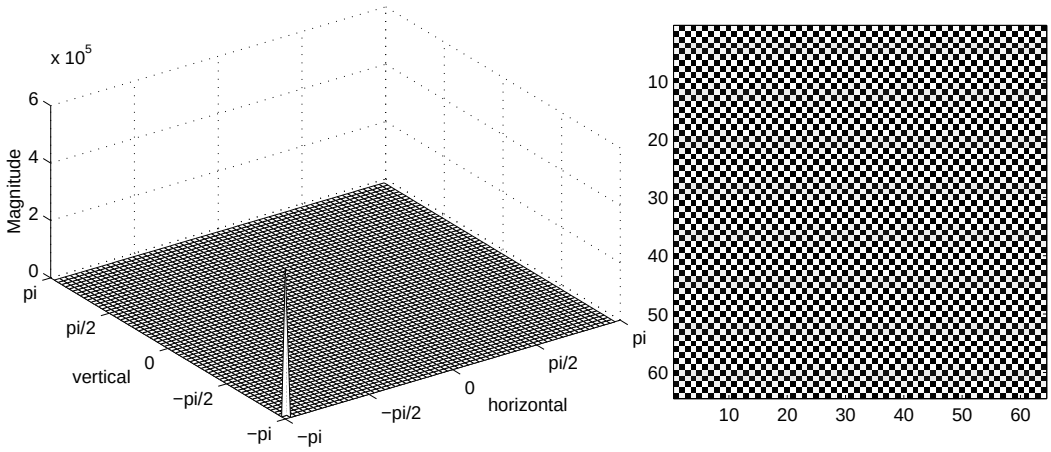


Figure 2.5: Diagonal oriented spatial sinusoids at Nyquist frequency.

We only exchanged n for t to indicate the shift from the temporal to the spatial domain and additionally we confined the signal to finite length of N samples. The latter was merely done to make the comparison easier to the two-dimensional formulation in a form that could be directly applied to an image:

$$F(u,v) = \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} I(x,y)\, e^{-i\,2\pi\,(\frac{xu}{N}+\frac{yv}{M})} \tag{2.18}$$

Here $F(u,v)$ represents again the frequency content of the signal but this time depending on oriented spatial frequencies (horizontal and vertical), $I(x,y)$ is the two-dimensional signal, i.e., the spatially ordered intensity values of the image, and N and M the extent of the image along its two axes.

Figure 2.3 shows in its top row images with a original width and height of 64 pixels containing two horizontal sinusoids of different spatial frequencies (0.1875 c/p and 0.5 c/p). Whereas the left and the centre image comprise just one sinusoid only, the right image combines the two. The second row depicts the vertical equivalent with sinusoids of frequencies 0.0926 c/p and 0.25 c/p. In the third row the image on the right hand side combines all of the sinusoids. Left to it a three-dimensional plot of the corresponding power spectrum is shown. Note that the Fourier transformation of a 64 x 64 image results in a matrix with 64 rows and columns and thus could be displayed as an image as well even though it is not an image and should not be confounded with one.

In the figure frequency increases from the centre to the outer edges. The energy contributed to the overall signal by each sinusoid can be seen as a sharp peak in the otherwise flat power spectrum. Because of the symmetry around the Nyquist frequency of the real part of the Fourier transform there are two peaks for each sinusoid, except for the one horizontal sinusoid at the Nyquist frequency where the two peaks fall together. Since the intensity values along one of the two axes are constant for all sinusoids, all peaks appear on the centrelines of the spectrum. Only frequency components with an askew orientation relative to the image axis

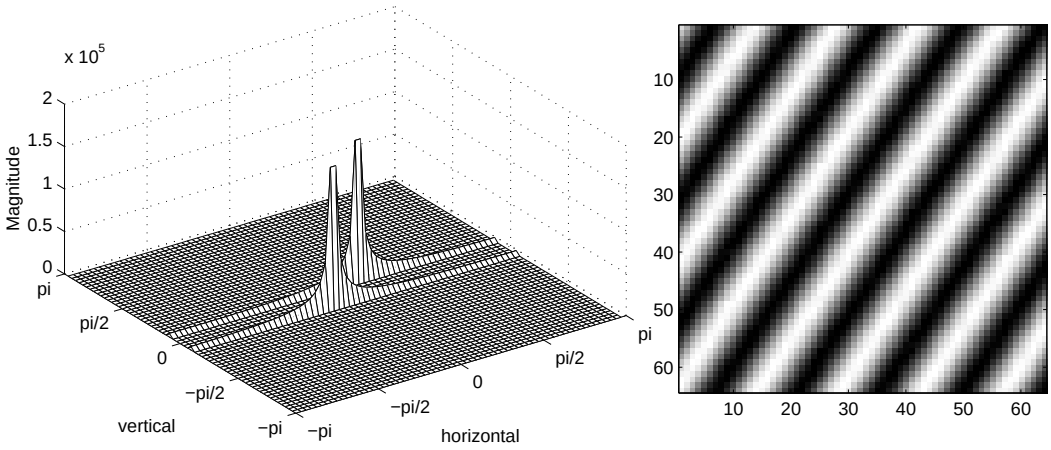


Figure 2.6: Spatial sinusoid with orientation angle of 56.3 degree and frequency 0.0845 c/p and its power spectrum.

would appear in the remaining area.

This becomes evident in Figure 2.4 that shows two sinusoids (0.1768c/p and 0.0663 c/p) oriented at 135 and 45 degrees respectively and their power spectrum. Now peaks can only be found in the diagonals. Let us look more closely at the second sinusoid. It was generated by the formula $128\cos(2\pi(3x/64 + 3y/64))$. Accordingly it is oscillating between -128 and 128. For display purposes we add 128 to keep the range of the values between 0 and 256, but for the Fourier transform the *zero mean* signal avoids the peak at the *zero frequency*. Focusing only on the x- or y-axis clearly three cycles can be counted everywhere in the image. But because of the 45 degree angle the frequency per image width or height would be $\sqrt{3^2+3^2} = 4.2426$ (see Shapiro and Stockman, 2001, for the rather simple derivation). Taking into account that the diagonal length of a pixel is $\sqrt{2} \approx 1.4$ we should find $4.2426 \cdot 1.4 \approx 6$ cycles along the diagonal, which can be easily checked.

At first glance it may be puzzling that the range of orientations of sinusoids that can occur in the signal and are represented in the power spectrum seems to be limited. For instance one step above the 0 frequency coefficient (or graphically one step away from the centre in the spectrum figures), there seem to be only four unique angles (0,45,90 and 135 degree) available.

But firstly some confinements really do exist due to the sampling theorem, i.e. analogous to the one-dimensional case where two adjacent sampling values - a black and a white pixel in our case - are needed to represent the Nyquist frequency, the highest diagonal frequency is a checkerboard-like pattern on pixel level. As can be seen in Figure 2.5 this corresponds to two sinusoids of 0.5 c/p that can only have orientation angles of 45 and 135 degree. It is then easy (and does not need to be done here) to examine what frequencies of what orientations could for instance be present in a 3 x 2 pixel neighbourhood.

And secondly, not all possible sines and cosines are needed as basic building blocks or conversely would appear as single sharp peaks in the power spectrum. In the one-dimensional case only sinusoids of integer-valued frequencies appear as single sharp peak and only those form the set of basic functions the signal is decomposed into. Accordingly in the two-dimensional case the tangent of the

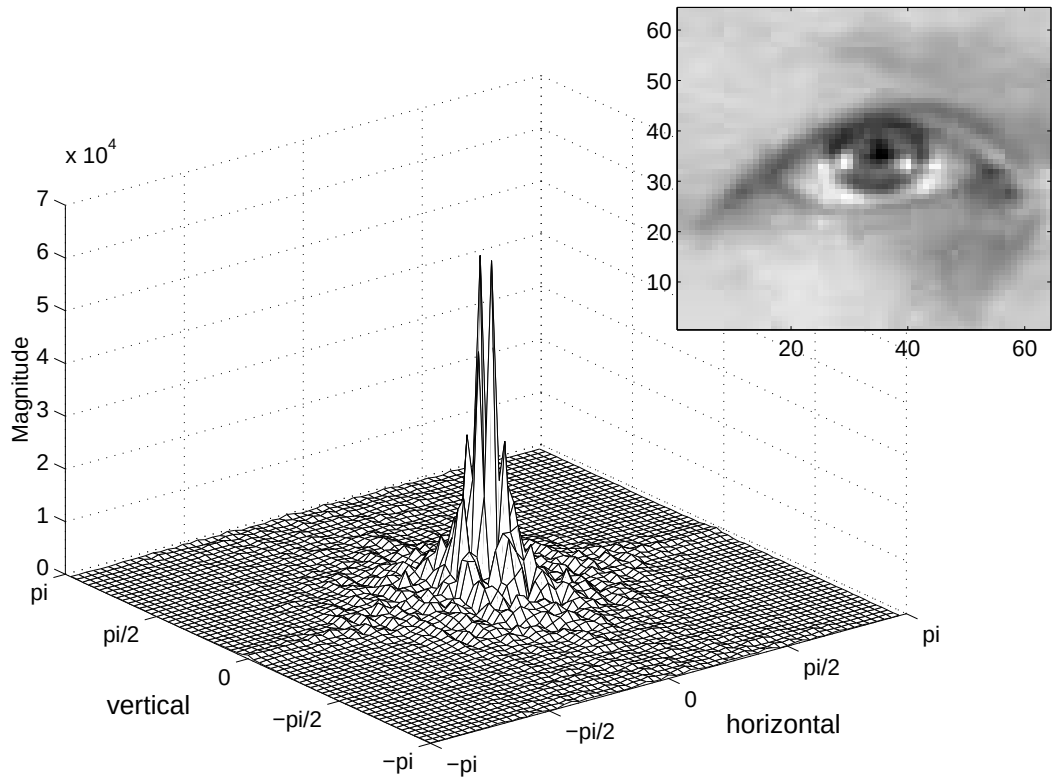


Figure 2.7: Picture of a human eye (64 x 64 pixel) and its power spectrum.

orientation angle must be a rational number. Figure 2.6 shows an example where the sinusoid was generated by the formula $128\cos(2\pi(4.5x/64 + 3y/64))$. The resulting orientation angle is 56.3 degree as given by $(\arctan(4.5/3)/\pi)180$ and the frequency is 0.0845 c/p which leads to the 7 or so visible cycles. If the image would be enlarged $4.5/3 = 1.5$ times in the x-direction the almost 10 cycles that are predicted by the appropriately modified calculation given above for the 45 degree example could be observed. As can be seen from the power spectrum in the same figure a whole range of frequency components contributes significant energy to the signal.

In natural images usually most of the energy is contained in the lower frequencies and spread over all possible angles. Figure 2.7 shows as an example a picture of a human eye in the same size as the artificial images above.

2.3.2 Discrete wavelet transformation

After a rather hesitant start in the 1980's wavelets became a shooting star in the world of applied mathematics used in signal processing, statistics and numerical analysis - triggered not at least by Daubechies (1992). Once the mathematical foundation was laid out, wavelet analysis asserted itself in fields as diverse as quantum physics, electrical engineering, seismic geology, image com-

pression, human vision, turbulence, radar, earthquake prediction, meteorology, etc.. What are wavelets and what made them so successful? Of course we can only give here a very superficial account of the vast field wavelet analysis comprises and we will do so from the viewpoint of signal processing. For more or less detailed overviews see Kaiser (1994); Strang and Nguyen (1997); Prasad and Iyengar (1997); Burrus, Gopinath, and Guo (1998); Nievergelt (1999) or Raghuveer and Bopardikar (1998), and for a very readable account of wavelet history, principles and relevance Burke Hubbard (1998).

Jawerth and Sweldens (Jawerth and Sweldens, 1993) summarise the general principle

> Wavelet theory involves representing general functions in terms of simpler, fixed building blocks at different scales and positions.

While Fourier transformation expands signals (or functions) in terms of sines and cosines (or equivalently in terms of complex exponentials) that are infinite, wavelet transformations use 'small waves', wavelets, that have their energy concentrated around a point in time or space, i.e. the energy of the wavelet function is finite.[4] Therefore they are well localised in time or space in contrast to Fourier analysis which is not localised at all. Because of this property Fourier analysis is very well suited for periodic, time/space-invariant or stationary signals, but naturally not so well for aperiodic, time/space-varying, transient signals. Here the two parameter dependency of the wavelet transform makes it superior, since scale *and* position are varied allowing simultaneous time and frequency analysis. The wavelet transformation is often compared to a musical score, which tells the musician what note to play and when to play it, while in the Fourier transformation the temporal localisation is inaccessibly hidden in the phases.

Very much at the heart of the matter is the *time-frequency resolution* problem called at times *Heisenberg uncertainty principle* or *Heisenberg-Gabor inequality*. It states that a signal cannot be perfectly localised both in time and frequency at the same time.

Let $x(t)$ with $t \in \mathbb{R}$ be an arbitrary signal whose energy E_x is finite (bounded):

$$E_x = \int_{-\infty}^{+\infty} |x(t)|^2 \, dt < \infty \tag{2.19}$$

and $X(\tau)$ its Fourier transform

$$X(\tau) = \int_{-\infty}^{+\infty} x(t) \, e^{-i2\pi\tau t} \, dt \tag{2.20}$$

Then we can consider $|x(t)|^2$ and $|X(\tau)|^2$ as probability distributions (see

[4] Wavelets that have finite duration as well are said to have *compact support*. Note that not all wavelets are compactly supported (see Raghuveer and Bopardikar, 1998).

Auger, Flandrin, Gonçalvès, and Lemoine, 1997) and look at their means

$$t_m = \frac{1}{E_x} \int_{-\infty}^{+\infty} t\,|x(t)|^2\,dt$$
$$\tau_m = \frac{1}{E_x} \int_{-\infty}^{+\infty} \tau\,|X(\tau)|^2\,d\tau \tag{2.21}$$

and variances

$$T^2 = \frac{4\pi}{E_x} \int_{-\infty}^{+\infty} (t - t_m)^2\,|x(t)|^2\,dt$$
$$B^2 = \frac{4\pi}{E_x} \int_{-\infty}^{+\infty} (\tau - \tau_m)^2\,|X(\tau)|^2\,d\tau \tag{2.22}$$

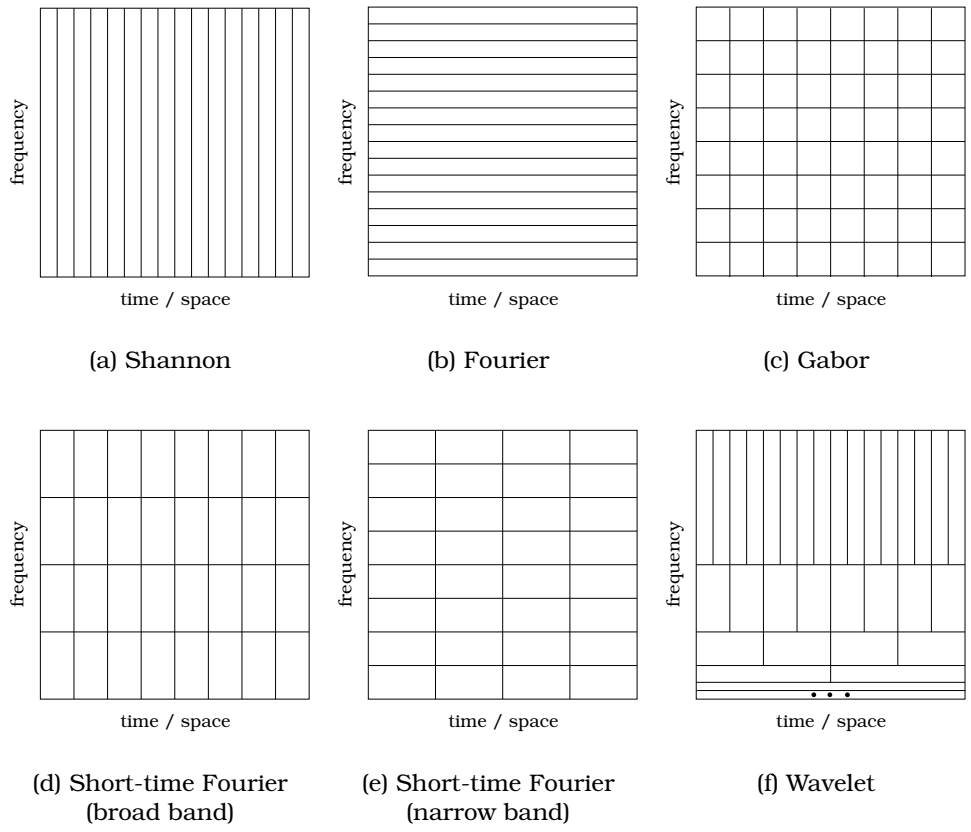
The Heisenberg-Gabor inequality now gives a lower bound for the product of these variances or standard deviations respectively, namely

$$T\,B \leq 1 \tag{2.23}$$

This means that if the spreading in time (or space) T decreases, the spreading in frequency, the bandwidth B, increases, and the other way round. Since the value for the joint time-frequency uncertainty can be exactly calculated, one can look for a transformation that minimises it and indeed there is a transformation right at the absolute minimum. This is the transformation using the so-called *Gabor* function, a sinusoid modulated by a Gaussian. The Gabor transformation resembles a *windowed Fourier transformation* (also called *Short-time Fourier transformation*), if a Gaussian window is used, but with one essential difference: the Gabor function has infinite support.

Calculating the averages and the standard deviations as done with equation (2.21) and (2.22) allows the creation of pictorial time-frequency representations, so-called *Heisenberg boxes* (also called *Heisenberg cells* or *time-frequency cells*), where the area of a rectangular box corresponds to the joint time-frequency uncertainty, the lengths of the box sides to the variance or standard deviation of the time and frequency probability distribution, and its position to the average of each distribution.

Figure 2.8 shows schematic time/space-frequency plots for several different transformations. In reality the distributions are of course not formed like those neat rectangulars, rather they are elliptic or irregular shaped blobs and they might overlap. Figure 2.8(a) depicts a sampled signal, which of course is well located in time (space), but provides no frequency information (except the bandwidth of the entire signal via the sampling theorem). In contrast the Fourier



(a) Shannon (b) Fourier (c) Gabor

(d) Short-time Fourier (broad band) (e) Short-time Fourier (narrow band) (f) Wavelet

Figure 2.8: Time/space-frequency representations (Heisenberg boxes)

transformation shown to the right of it in Figure 2.8(b) gives good frequency resolution, but contains no information about their locations (except the length of the whole signal). The Gabor transformation shown in Figure 2.8(c) not only minimises the time/space-bandwidth product, thus having the smallest possible area a Heisenberg box can take on, but also is equally well located in time (space) as in frequency yielding the square shape of the box.

The Short-time Fourier transform can be considered to be half way between the sampled time (space) signal and the Fourier transform. With its flexible window length it can be tuned towards a better temporal (spatial) resolution and less good frequency location (Figure 2.8(d), broad-band) or in the opposite direction (Figure 2.8(e), narrow-band). The area of the rectangle, however, is not at the minimum and fixed for all frequencies and the entire temporal (spatial) domain. Wavelet transformation on the contrary adapts its way of decomposition to the frequencies under scrutiny, as shown in Figure 2.8(f): At the lowest frequencies very good frequency resolution is traded for very poor temporal (spatial) resolution. Moving towards higher frequencies the trend is stepwise reversed. Thus the wavelet transformation is best suited for signals with high frequency components of short duration and low frequency components of long duration. On both sides of the spectrum this would equally mean that the signal has to be stationary only for one to a few cycles in order to be localised in time (space) and frequency

sufficiently well. Most signals encountered in practise fulfil this condition.

How can the wavelet transform achieve these favourable properties. Let us have a look at the formal definition. In the wavelet transformation a function or signal $f(t)$ is linearly decomposed into a set of linearly independent functions, the so-called *basis functions*, $\psi_{s,l}(t)$, the wavelets,

$$f(t) = \sum_s \sum_l c_{s,l}\, \psi_{s,l}(t) \tag{2.24}$$

where $c_{s,l}$ are the wavelet coefficients. In particular a wavelet system can be generated from a single mother wavelet $\psi(t)$, i.e., all basis functions $\psi_{s,l}(t)$ can be constructed by simply scaling and translating $\psi(t)$:

$$\psi_{s,l}(t) = \psi(2^{-s}t - l) \tag{2.25}$$

with $s, l \in \mathbb{Z}$, the set of all integers. The expression $\psi(2^{-s}t - l)$ causes the wavelet in question to be shifted in time or space by l, thus providing the temporal or spatial localisation, and rescaled by 2^{-s}, thus analysing the signal on different scales. Note that the generating wavelet is dilated by powers of two, the most common choice for the basis resulting in the so-called *dyadic sampling* or *dyadic wavelet*. Thus the ***D**iscrete **W**avelet **T**ransform* (DWT) becomes

$$f(t) = \sum_{s=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} c_{s,l}\, 2^{-\frac{s}{2}}\, \psi(2^{-s}t - l) \tag{2.26}$$

The factor $2^{-\frac{s}{2}}$ ensures a constant norm over all scales. Note that at any dilation by 2^s the translation parameter takes effectively the form $2^s l$ where l is again an integer.[5] In this way the wavelet transform's adapting localisation in time (space) and frequency is accomplished.

The DWT is in principle not redundant (see Raghuveer and Bopardikar, 1998), though in practise for most signals many of the coefficients might be zero or close to zero. Since the DWT is still a transformation of a continuous-time signal, it should be denoted - analogously to the Fourier series - as 'continuous-time wavelet series', as many authors point out. But the different naming is already established.

So far we have not said much about the wavelet function ψ itself. In general it can be a real- or complex-valued function that fulfils three requirements:

[5] This might look like a parenthesis error in the DWT equation on first glance, however, it becomes clear, when looking at the discretisation of the CWT (***C**ontinous **W**avelet **T**ransformation*) as given in Daubechies (1992):

$$(T^{wav} f)(a, b) = |a|^{-1/2} \int f(t)\, \psi\left(\frac{t-b}{a}\right)\, dt$$

For the dilation parameter a positive or negative powers of one fixed dilation parameter $a_0 \geq 1$ are chosen: $a = a_0^m$. Since m changes the wavelet width, the discretisation of the translation parameter b should depend on m. As in deed the width of $\psi(a_0^{-m} t)$ is proportional to a_0^m, b is discretised as $b = n b_0 a_0^m$ with a fixed $b_0 \geq 0$ and $n \in \mathbb{Z}$. In this way the DWT becomes

$$\begin{aligned} T^{wav}_{m,n}(f) &= a_0^{-m/2} \int f(t)\, \psi(a_0^{-m}(t - n b_0 a_0^m))\, dt \\ &= a_0^{-m/2} \int f(t)\, \psi(a_0^{-m} t - n b_0)\, dt \end{aligned}$$

i. The integral of the function is zero

$$\int_{-\infty}^{\infty} \psi(t)\, dt = 0 \tag{2.27}$$

ii. The function has finite energy (i.e., is square integrable)

$$\int_{-\infty}^{\infty} |\psi(t)|^2\, dt < \infty \tag{2.28}$$

iii. Its Fourier transform decays sufficiently fast

$$\int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{\omega}\, d\omega < \infty \tag{2.29}$$

where $\Psi(\omega)$ is the Fourier transform of $\psi(t)$.

The first condition essentially says that the function has to oscillate (though there are exceptions, see Raghuveer and Bopardikar, 1998), the second forces finite support. The third condition, known as *admissibility condition*, is only necessary to ensure that the inverse of the wavelet transformation can be computed as well (see Daubechies, 1992, page 7,63). For all practical purposes it is equivalent to the first condition (see Daubechies, 1992, page 24). Since the construction of wavelets is far beyond the scope of this work, we point the reader to the already cited literature, especially Daubechies (1992), and resort to just show two wavelets in Figure 2.9.

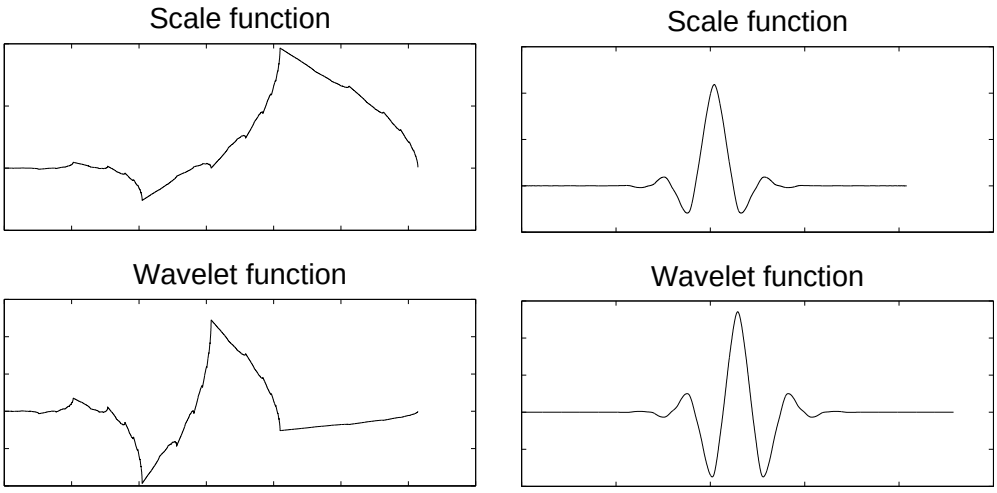
Most wavelets are *orthogonal*, i.e . their *inner product* is zero:

$$\langle \psi_{s,l}(t), \psi_{S,L}(t) \rangle = \int_{-\infty}^{\infty} \psi_{s,l}(t)\, \psi_{S,L}(t)\, dt = 0 \tag{2.30}$$

Thus they constitute an orthogonal basis. On the one hand this makes it easier to compute the coefficients $c_{S,L}$ by using the inner product of the signal (or function) and the corresponding wavelet $\psi_{S,L}$. Multiply expansion equation (2.26) by $\psi_{S,L}(t)$ and integrate (Strang and Nguyen, 1997):

$$\int_{-\infty}^{\infty} f(t)\, \psi_{S,L}(t)\, dt = c_{s,l} \int_{-\infty}^{\infty} (\psi_{S,L}(t))^2\, dt \tag{2.31}$$

Because of orthogonality all integrals of $\psi_{s,l}$ times $\psi_{S,L}$ disappear except the one term that has $s = S$ and $l = L$, which leads to $(\psi_{S,L}(t))^2$. $c_{S,L}$ is simply the ratio of the two integrals. On the other hand this allows using wavelets in a *multiresolution analysis* described in the next section.



(a) Daubechies wavelet (b) Cubic spline wavelet

Figure 2.9: Two wavelets with corresponding scaling function

2.3.3 Multiresolution analysis

Multiresolution techniques exploit the idea of analysing a given signal at different resolutions or scales. This can be compared with measuring an ocean coast line with rigid rulers of different sizes (e.g., powers of two). The representation of the coast line will look different depending on what ruler size had been used and the resulting length would be different as well. However, in multiresolution analysis as defined by S. Mallat (e.g., Mallat, 1989) for image processing the so-called *scaling function* takes the part of the ruler and extracts a series of images at resolutions differing by a factor of two, which include only the low frequency part of the image at each level. In one direction, moving towards higher resolutions each successive level contains all previous ones and eventually approximates the original image. In the other direction increasingly less information is represented. The difference between two successive levels can be encoded using a wavelet function.

We will now describe multiresolution analysis more formally following Raghuveer and Bopardikar (1998) and Daubechies (1992), and return to a one-dimensional signal for the time being, but again the reader must be warned that the account will be very superficial and he or she should consult the already mentioned wavelet literature and Vaidyanathan (1993).

A multiresolution analysis consists of a sequence of nested linear vector spaces $\ldots \subset V_1 \subset V_0 \subset V_{-1} \ldots$. The closed subspaces must satisfy the following five conditions:

i. Every subspace is entirely included in the next one, forming a sequence of successive approximation spaces:

$$V_k \subset V_{k-1} \qquad \text{for all } k \in \mathbb{Z} \tag{2.32}$$

ii. The union of all subspaces is *dense* in the space of all square integrable

functions

$$\overline{\bigcup_{k\in\mathbb{Z}} V_k} = L^2(\mathbb{R}) \tag{2.33}$$

where the over-line denotes the *set closure* (for an explanation of the term 'dense' and 'set closure' see texts on mathematical analysis or Weisstein, 1999).

iii. All subspaces only have the set containing the all-zero function or zero vector in common.

$$\bigcap_{k\in\mathbb{Z}} V_k = \{0\} \tag{2.34}$$

The following last two conditions make the nested vector space sequence an actual multi-resolution analysis:

v. Elements in a space V_k are simply scaled versions of the elements in subspace V_{k-1}

$$f(t) \in V_k \Leftrightarrow f(2t) \in V_{k-1} \tag{2.35}$$

A function in a certain subspace dilated by factor 2 yields a function in the next coarser subspace. The value 2 of the factor is not a necessity, actually it is required only to be a power of two, but we confine ourselves in this presentation to dyadic relationships.

vi. There exist a so-called *scaling function* $\phi(t)$ such that

$$\{\phi(t-n);\ n \in \mathbb{Z}\} \quad \text{is a basis in } V_0 \tag{2.36}$$

According to Raghuveer and Bopardikar (1998)

> ... the final property requires that there be a scaling function $\phi(t)$ such that the set $\{\phi(t-n) : n \text{ integer}\}$ is linearly independent, and any function $f_0(t) \in V_0$ is expressible as
>
> $$f_0(t) = \sum_{n=-\infty}^{\infty} a(0,n)\phi(t-n)$$
>
> for a sequence of scalars $a(0,n)$ where $n = 0, \pm 1, \pm 2$, and so on.

We can now formulate the recursive so-called ***M**ulti**r**esolution **A**nalysis* (MRA) equation (also called *refinement* or *dilation* equation).

$$\phi(t) = \sum_{n=-\infty}^{\infty} c(n)\,\sqrt{2}\,\phi(2t-n) \tag{2.37}$$

that allows us to write ϕ as a weighted sum of its translates at a resolution twice as fine. The coefficients $c(n)$ are called the *scaling function coefficients* and the factor $\sqrt{2}$ ensures a constant norm over all scales.

Two properties of an MRA are now of special interest for us. Firstly, as shown in Daubechies (1992) for every sequence of closed subspaces fulfilling conditions (2.32)-(2.36) with the additional requirement in (2.36) that $\{\phi(t-n);\ n \in \mathbb{Z}\}$ must

be an *orthonormal* basis,[6] there exists an orthonormal wavelet basis $\{\psi_{s,l}\,;\, s,l \in \mathbb{Z}\}$ of $L^2(\mathbb{R})$, $\psi_{s,l}(t) = 2^{-s/2}\,\psi(2^{-s}\,t - l)$ such that, for all $f \in L^2(\mathbb{R})$

$$P_{s-1}f = P_s f + \sum_{l\in\mathbb{Z}} \langle f, \psi_{s,l}\rangle\, \psi_{s,l} \tag{2.38}$$

where $P_s f$ is the orthogonal projection of f on V_s. This means that the difference between each MRA level is covered by a certain wavelet function on the corresponding level and all its translates:

$$V_{s-1} = V_s \oplus W_s \tag{2.39}$$

where W_s is the subspace generated by the set $\{\psi(2^{-s}\,t - l);\; s,l \in \mathbb{Z}\}$, and $\oplus$ is the orthogonal sum (since $V_{s-1} \perp W_s$).

Secondly, a MRA can be expressed and realized as a cascade filter bank of pairwise low and high pass filters. The filters must be half-band filters and moreover satisfy certain criteria which led to the name ***Q**uadrature **M**irror **F**ilters* (QMV, see the already cited wavelet literature, in particular Strang and Nguyen, 1997). Their coefficients can be calculated from the scaling function and vice versa.

Therefore the wavelet transformation and in particular its discrete time version DTWT (***D**iscrete-**t**ime **W**avelet **T**ransform*) can be realized as filter bank, too. The resulting output signals at each level of the DWT/DTWT will be bandlimited with theoretically a bandwidth of exactly one octave. In practise, however, no filter is perfect and the results will vary slightly with the steepness of the filter slope.

Note that we will use the term multiresolution analysis in a less formal way in chapter 3 to characterise analogies between the motion tracking coarse-to-fine strategy per se and its included image processing part.

2.3.4 Image decomposition with two-dimensional wavelets

So far we have considered only one-dimensional signals. Our goal, however was to obtain a wavelet decomposition of digital images - spatially discrete two-dimensional finite signals.

In the two-dimensional case we need to construct three two-dimensional wavelets by multiplying together a one-dimensional scaling function ϕ and the corresponding wavelet ψ

$$w_{\phi\psi} = \phi(x)\,\psi(y) \tag{2.40a}$$

$$w_{\psi\phi} = \psi(x)\,\phi(y) \tag{2.40b}$$

$$w_{\psi\psi} = \psi(x)\,\psi(y) \tag{2.40c}$$

These wavelets are directionally sensitive: $w_{\phi\psi}$ encodes changes in the vertical direction, $w_{\psi\phi}$ in the horizontal direction, and $w_{\psi\psi}$ in the diagonal direction. By a linear combination of the dyadic translates and dilates of all three

[6] An orthonormal basis is a basis that is orthogonal, i.e. for all vectors v belonging to the basis $\langle v_n, v_m\rangle = 0$ when $n \neq m$, and further the vectors have unit length, i.e., $\langle v_n, v_n\rangle = 1$.

wavelets every two-dimensional square integrable function can be approximated (Raghuveer and Bopardikar, 1998):

$$\begin{aligned} f(x,y) = &\sum_{s=-\infty}^{\infty}\sum_{l=-\infty}^{\infty}\sum_{k=-\infty}^{\infty} b_w(l,k)\, w_{\psi\phi}(2^{-s}x-l,\, 2^{-s}y-k) + \\ &\sum_{s=-\infty}^{\infty}\sum_{l=-\infty}^{\infty}\sum_{k=-\infty}^{\infty} c_w(l,k)\, w_{\phi\psi}(2^{-s}x-l,\, 2^{-s}y-k) + \\ &\sum_{s=-\infty}^{\infty}\sum_{l=-\infty}^{\infty}\sum_{k=-\infty}^{\infty} d_w(l,k)\, w_{\psi\psi}(2^{-s}x-l,\, 2^{-s}y-k) \end{aligned} \tag{2.41}$$

where $k, l, s \in \mathbb{Z}$ and $b_w(l,k)$, $c_w(l,k)$, $d_w(l,k)$ are the wavelet coefficients of each oriented wavelet. To expand the multiresolution approach to the two-dimensional case the scaling function has to be augmented as well by defining

$$w_{\phi\phi} = \phi(x)\,\phi(y) \tag{2.42}$$

Observe that the subspaces spanned by the four functions $w_{\phi\psi}$, $w_{\psi\phi}$, $w_{\psi\psi}$, and $w_{\phi\phi}$ on each scaling level $W_{a,s}$, $W_{b,s}$, $W_{c,s}$, and V_s are orthogonal to each other and we have:

$$V_{s-1} = V_s \oplus W_{a,s} \oplus W_{b,s} \oplus W_{c,s} \tag{2.43}$$

Thus equivalently to the one-dimensional case the differences between two successive levels of a multiresolution analysis are covered by wavelet functions, albeit in the two-dimensional case the linear combination of three wavelet functions for one level is necessary. Moreover the wavelet decomposition can be again expressed and implemented as cascade filter bank. Figure 2.10 shows the required filter bank structure.

The input is the approximation A_{s-1} of the lower level $s-1$. The way the two-dimensional wavelets and the scaling function are derived in equation (2.41) and (2.42) ensures separability of the filter kernel, which allows filtering along the rows and along the columns separately. Accordingly the next step entails forking the path depending on which kind of filter is applied along the image rows only. One branch is created by filtering with a high pass filter, the other one by filtering with a low pass filter. Since the bandwidth in both resulting images is reduced by factor two, the rows can be decimated by factor two. Afterwards the path is split again to filter each decimated image along the columns with either a high or low pass filter and subsequently downsample along the columns. The outcome consists of four images at level s with an area size of one fourth of the input image. $D_s^{diagonal}$, filtered along both dimensions with the high pass filter, contains the diagonal details, $D_s^{vertical}$ the vertical, and $D_s^{horizontal}$ the horizontal details. A_s, low pass filtered along both dimensions, is the approximation at level s. It will be used as input for the next decomposition step leading to level $s+1$. Notice that because of the subsampling the filters do not need to be changed at all.

The different D's are called details or subbands, A is called approximation. On the lowest level the input image is of course the original image. At first sight the

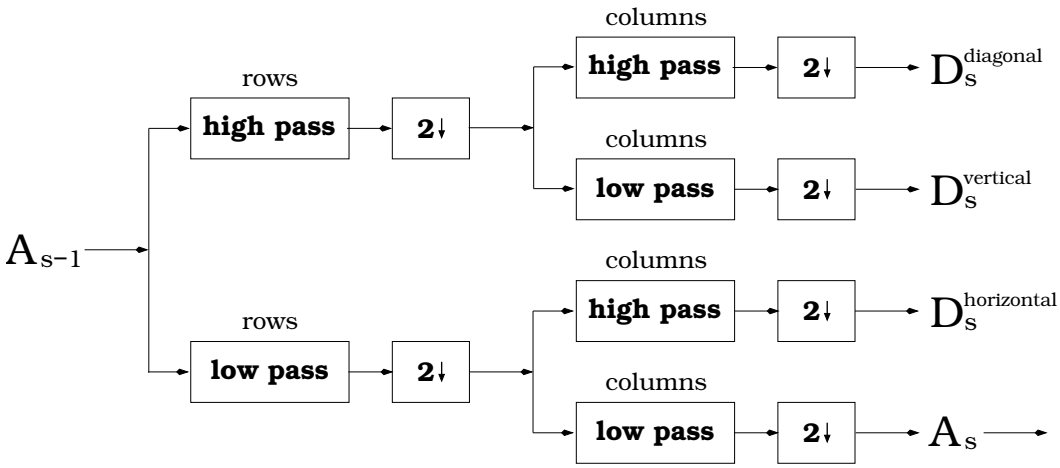


Figure 2.10: Filter bank implementing the two-dimensional DTWT. The input approximation is filtered first along the rows with a high pass or a low pass filter and then decimated. Each of the resulting signals is filtered along the columns with a high pass or a low pass filter and decimated. This yields four output signals for the next level of the DTWT: three spatially oriented subbands and an approximation.

typical wavelet terminology seems counter-intuitive in describing a level as higher when actually moving towards lower frequencies. Within the 'wavelet world', however, the equivalent parameter in the wavelet filter function is not frequency but scale, which is the inverse of frequency. Nevertheless some authors (e.g., Strang and Nguyen, 1997 and Burrus et al., 1998) started to invert the naming based on a slightly different formulation of the wavelet equation, i.e. the wavelet is compressed and not dilated, and reversed subspace indices in the MRA, i.e., $\ldots \subset V_{-1} \subset V_0 \subset V_1 \ldots$ (cf. equation (2.32)). The seemingly more intuitive approach, however, is then sometimes led *ad absurdum* when the authors following the original wavelet terminology speak of dilation while meaning compression.

For several reasons explained below and in chapter 3 we do not want to subsample the image in the motion tracking procedure. As a consequence we in fact have to adapt the filter on each level. This can be achieved by convolving the low pass filter (corresponding to the scaling function) with itself iteratively $n-1$ times on level n given the original image was the input at level 1, and then convolve the high pass as well as the low pass filter with the iterated scaling function/low pass filter (starting of course not before level 2).

Wavelet coefficients can be spatially arranged like the original image and because of their good spatial location they look very much like an image when interpreted as intensity values (unlike Fourier transform coefficients). But in a strict sense they do not constitute an image, and a synthesis filter bank should be applied first to recover some kind of image. However, for many wavelets the difference between reconstruction and wavelet coefficients is almost not perceptible. In the following we will present wavelet subbands as images and when without ambiguity from the context speak of pixel instead of wavelet coefficients.

Figure 2.11 shows the four output 'images', for the fifth level of the wavelet filtering for a video frame showing a face and an artificially constructed pattern

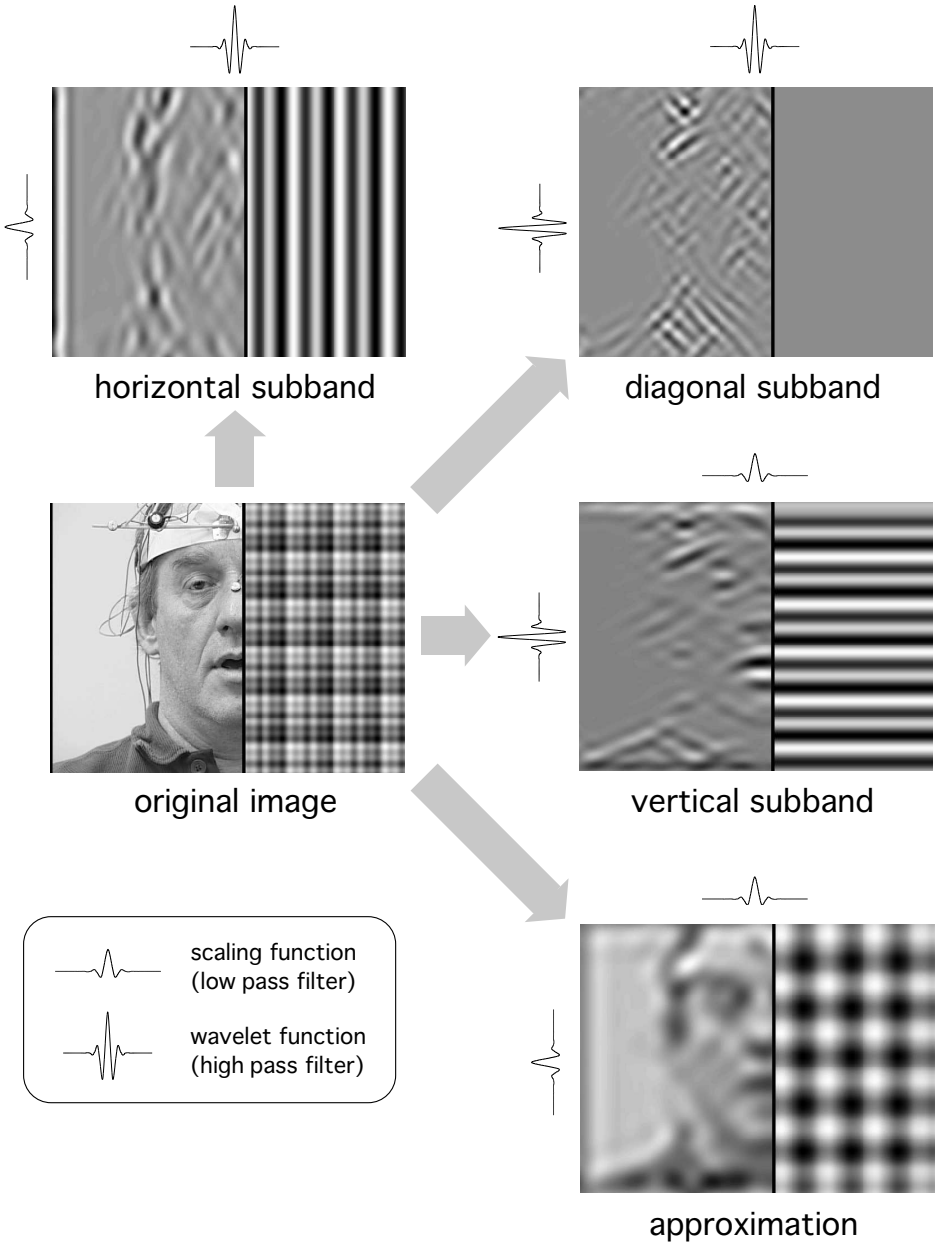


Figure 2.11: Wavelet transformation of a natural image and an artificial pattern. The pattern was created by combining three sinusoids in both the horizontal and vertical directions. The original image is shown in the middle on the left side. The remaining images show the decomposition into the three subbands and the approximation on the fifth level of the DWT. They are created by filtering along rows and columns with either the high pass or the low pass filter (indicated schematically by the graph of the wavelet function, which constitutes the high pass filter, and the graph of the scaling function, which constitutes the low pass filter).

(level zero corresponds to the unfiltered image, level one to the output of the first pass through the filter bank, and so on). The spatial frequency bandwidth for each subband ranges from 0.016 to 0.031 cycles/pixel, which corresponds to wavelengths from 64 to 32 pixels. The pattern was created by combining three sinusoids in both the horizontal and vertical directions. The frequencies of the sinusoids (0.047, 0.023 and 0.012 cycles/pixels) were chosen to be equal to the centre frequency of the spatial frequency band for the fourth, fifth and sixth level of the wavelet transform.

As can be seen in the figure each subband contains only the one sinusoid, which lies within its frequency range and is of the same direction. The lower frequency sinusoid remains in the approximation, while the higher one is completely filtered out (but would appear in the subbands of the fourth level). Note that the signal is not subsampled: Even though the signal does not contain the high spatial frequency part of the original image and therefore, could be re-sampled at lower sampling rate without loss of information, doing so would introduce a larger error in the motion tracking on a coarser scale, simply because one pixel in the subsampled image would correspond to several pixels in the original image. As we want to use the tracking on a coarser scale to predict the approximate motion on a finer scale, we have to avoid this error (see chapter 3).

For Figure 2.11 and for the motion tracking procedure filters were used that correspond to a *biorthogonal scheme with cubic spline wavelets* (see Sánchez, Prelic, and Galán, 1996, for details about the algorithm). In spline wavelets the scaling and the wavelet function are constructed using basic spline functions (Daubechies, 1992) - in our case cubic splines. Spline wavelets can be created with compact support, i.e., the scaling and wavelet functions are finite: they have the value zero outside a certain interval. This avoids truncation errors during computation of the wavelet transform. Their 'disadvantage' is that they cannot be constructed from analytic formulae, but their graph can be computed with arbitrarily high precision using iterations. The scaling and wavelet functions shown in Figure 2.11 are determined in this way. The real filter kernels were approximation of these functions with a few coefficients and of course not as smooth.

2.4 Principal component analysis

The statistical method of ***P**rincipal **C**omponent **A**nalysis* (PCA) plays an important role in the validation of the motion tracking system. For that reason we will very briefly present the concept here. For details the reader is referred to the excellent and extensive covering of all aspects of PCA in Jackson (1991).

> The method of principal components is primarily a data-analytic technique that obtains linear transformations of a group of correlated variables such that certain optimal conditions are achieved. The most important of these conditions is that the transferred variables are uncorrelated. (Jackson, 1991, page 1)

PCA forms a part of the large field of multivariate data analysis. Often the assumption is made that all or a subset of the components represent essential or even causal factors of the correlated variables under scrutiny, thus 'explain' their covariance. For instance they could allow separation of noise from the actual data or - in case of PCA applied to face motion measurements - identify physiological mechanism underlying face motion.

Let us assume we have a set of p variables with equally n observations represented in the $n \times p$ matrix $\mathbf{X}$:

$$\begin{bmatrix} x_{11} & \dots & \dots & x_{1p} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ x_{n1} & \dots & \dots & x_{np} \end{bmatrix} \tag{2.44}$$

In the case of face motion the variables are the measurement points. The different coordinates of a single measurement point are treated as different variables and become individual columns, e.g, with two-dimensional motion data $p = 2h$ where h is the number of measurement points.

Then the $p \times p$ *covariance matrix* $\mathbf{S}$ of the data becomes:

$$\begin{bmatrix} s_1^2 & \dots & \dots & s_{1p} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ s_{p1} & \dots & \dots & s_p^2 \end{bmatrix} \tag{2.45}$$

where

$$s_{ij} = \frac{n \sum_{k=1}^{n} x_{ki} x_{kj} - \sum_{k=1}^{n} x_{ki} \sum_{k=1}^{n} x_{kj}}{n(n-1)} \tag{2.46}$$

Using the matrix product (2.45) and (2.46) can be written

$$\mathbf{S} = \frac{\mathbf{X}_{cen}^T \mathbf{X}_{cen}}{n} \tag{2.47}$$

where $\mathbf{X}_{cen}$ is the centred data set, i.e., the variable mean is subtracted for each variable.

Clearly, the sum of the diagonal elements $\mathrm{Tr}(S) = s_{11} + s_{22} + \ldots + s_{pp}$ is the sum of the variances of all variables and in this way a measure for the overall variability of the data. If an off-diagonal element s_{ij} of $\mathbf{S}$ is not zero, then the variables represented by $\mathbf{X}(i)$ and $\mathbf{X}(j)$, the i^{th} and the j^{th} column of $\mathbf{X}$, are linearly related. The strength of the relationship is given by the correlation coefficient $r_{ij} = s_{ij}/(s_i s_j)$. However, the correlation coefficient is not needed for PCA:

> The method of principal components is based on key results from matrix algebra: A $\mathbf{p} \times \mathbf{p}$ symmetric, nonsingular matrix, such as the covariance matrix $\mathbf{S}$, may be reduced to a diagonal matrix $\mathbf{L}$ by premultiplying and postmultiplying it by a particular orthonormal matrix $\mathbf{U}$ ... (Jackson, 1991, page 7)

Formally expressed this is

$$\mathbf{U}^T \mathbf{S} \mathbf{U} = \mathbf{L} \tag{2.48}$$

$\mathbf{L}$ is $\mathbf{p} \times \mathbf{p}$ with at most the diagonal elements $l_1, l_2, \ldots, l_p$ being non-zero. They are called *eigenvalues* or *characteristics roots*. The columns of the $p \times p$ matrix $\mathbf{U}$ $(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_p)$ are called *eigenvectors* or *characteristics vectors*. l_m is the corresponding eigenvalue to the eigenvector $\mathbf{u}_m$.

Viewed from a geometrical point of view the above is a *principal axis* rotation of the original coordinate axis $x_1, x_2, \ldots, x_p$ around their means. The eigenvectors contain the direction cosines relating the original axis to the new. The values of the variables in the new coordinate system can be obtained by

$$\mathbf{z}_k = (\mathbf{x}_k - \bar{x})\,\mathbf{U} \tag{2.49}$$

$\mathbf{x}_k$ is an $1 \times p$ vector of the k^{th} observations on the original variables, $\bar{\mathbf{x}}$ an $1 \times p$ vector with the variable means, and $\mathbf{z}_k$ an $1 \times p$ vectors with the transformed observations. The transformed variables are called *principal components*, its individual (transformed) observations *z-scores*. Thus the k^{th} observation of the i^{th} principal component is

$$z_{ki} = (\mathbf{x}_k - \bar{x})\,\mathbf{u}_i \tag{2.50}$$

As can be seen from (2.50) z_{ki} is indeed a linear combination of the original observations on all variables weighted with the coefficients $\mathbf{u}_i$

$$z_{ki} = (x_{k1} - \bar{x})u_{1i} + (x_{k2} - \bar{x})u_{2i} + \ldots + (x_{kp} - \bar{x})u_{pi} \tag{2.51}$$

Note that a linear combination of variables like in 2.51 is only then a rotational transformation (Bortz, 1993), if

$$\sum_{m=1}^{p} u_{mi}^2 = 1 \tag{2.52}$$

This is guaranteed by the property of $\mathbf{U}$ to be orthonormal, i.e., the eigenvectors are orthogonal and have unit length. This also ensures that the principal components are uncorrelated and have variances equal to the corresponding eigenvalues.

So far we have skipped the question of how to obtain $\mathbf{L}$ and $\mathbf{U}$. There are at least two techniques (see Jackson, 1991) of which the more frequently used is ***S**ingular **V**alue **D**ecomposition* (SVD).

In its general form SVD decomposes a matrix $\mathbf{X}$ such that

$$\mathbf{X} = A\,D\,B^T \tag{2.53}$$

where D is a matrix with nonnegative diagonal elements in decreasing order (the *singular values* of $\mathbf{X}$) and A and B have orthonormal columns and are called the *left* and *right singular vectors*. In connection with PCA equation (2.53) can be rewritten as

$$\mathbf{X} = U^*\,L^{1/2}\,U^T \tag{2.54}$$

Notice that we do not have to obtain the covariance matrix but instead can work on the data itself, and that we therefore get the square root of the eigenvalues. There exist several algorithms solving the SVD problem and a lot of numerical packages provide functions implementing them. In e.g., MATLAB the solution can be obtained with the SVD function by

```
[U,L,V] = SVD(X)
```

An important advantage of the PCA model is that it is invertible. Since $\mathbf{U}$ is orthonormal, the inverse of $\mathbf{U}$ is its transpose:

$$\mathbf{U}^{-1} = \mathbf{U}^{T} \tag{2.55}$$

Thus equation (2.49) can be inverted as follows

$$\mathbf{x}_k = \bar{\mathbf{x}} + \mathbf{U}^{T}\mathbf{z}_k \tag{2.56}$$

In this way the original variables could be reconstructed exactly. In practise, however, the case where the reconstruction is based only on a subset of components (by deleting the z-scores and respective eigenvectors of components that should be disregarded) is the more interesting. For example, noise in the data could be removed in this manner. Or even more interestingly in the case of face motion measurements as input data, the impact of single components on the face motion could be studied systematically.

Chapter 3

Video-based face motion tracking: the system

Our proposed algorithm can be broken down into two major procedures that are independent of each other both conceptually and implementation-wise: *initialisation* and *motion tracking*.

The purpose of the initialisation procedure is to handle all 'administrative' tasks with regard to the input video sequence and, if available, external head motion tracking data. This comprises the temporal arrangement and labelling of the filename list pointing to the image files according to their frame type, reading in of the external head motion tracking data and synchronisation of them with the video sequence. Furthermore the initialisation procedure acquires from the raw image data with the aid of the user the parameters that allow the creation and projection of the three-dimensional ellipsoid mesh onto the face in the video sequence. Obviously the mesh has to fit, at least approximately, the spatial extent of the face in all three dimensions. Part of this step is the assumption and application of a camera model to be used throughout the whole motion tracking procedure.

The motion tracking procedure itself progresses frame by frame through the video sequence. The measurement is accomplished by projecting the half-ellipsoid mesh onto the subjects face (with a fixed attachment to the facial surface maintained by the use of head motion tracking data) and then deform its interior according to two-dimensional face motion data derived from an image motion estimation process. Realizing a coarse-to-fine strategy the mesh is not superimposed onto the raw image but onto wavelet subbands of the image data proceeding from lower spatial frequencies to higher ones and with it a coarser mesh is iteratively refined until the final resolution for the tracking is reached.

3.1 Coordinate systems

Before describing both procedures in detail the main coordinate systems used must be described.

3.1.1 Image

Digital images are usually stored as a matrix of intensity values that correspond to the amount of received light energy integrated over the area of a single CCD (**C**harge **C**oupled **D**evice) element in case of a digital camera or a small part of the image quantised in case of a analogous image source. A possible coordinate system would be to treat the row and column indices as the midpoint values of the pixels with the x-axis corresponding to the columns and the y-axis to the rows starting from the upper left corner of the image. This is actually the most widely used coordinate system in image processing, however, the different orientation (different sign) of the y-axis compared to the conventional Cartesian coordinate system can cause some inconvenience. Note that the axes have continuous values serving the purpose of allowing coordinate transformations from other coordinate systems (e.g., 'real world') without quantisation as well as image processing algorithm involving subpixel arithmetic.

From the set of alternative systems a somewhat unfamiliar option was chosen for this work: a standard two-dimensional Cartesian coordinate system but with its origin located in the image centre (see Fig. 3.1). The main advantage arises from the fact that most camera models assume the optical axes as a line perpendicular to the image plane intersecting it right at the centre of the image. (cf. section 2.2.1).

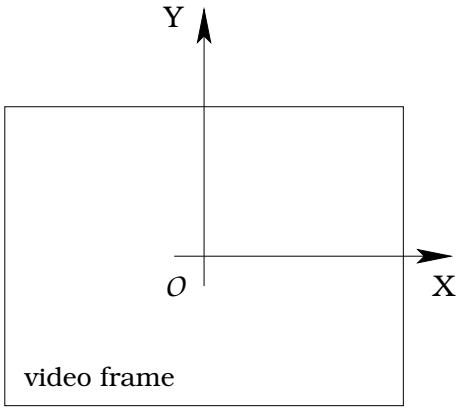


Figure 3.1: Image coordinate system with origin $\mathcal{O}$ in the centre of the image

3.1.2 OPTOTRAK

The OPTOTRAK coordinate system is a three-dimensional Cartesian coordinate system defined relative to OPTOTRAK's multiple camera device. The origin is located at the centre camera. The x-axis is oriented vertically increasing upwards, the y-axis horizontally increasing to the left hand side of the device, and the z-axis is in line with the optical axis of the centre camera increasing towards the camera (see Fig. 3.2).

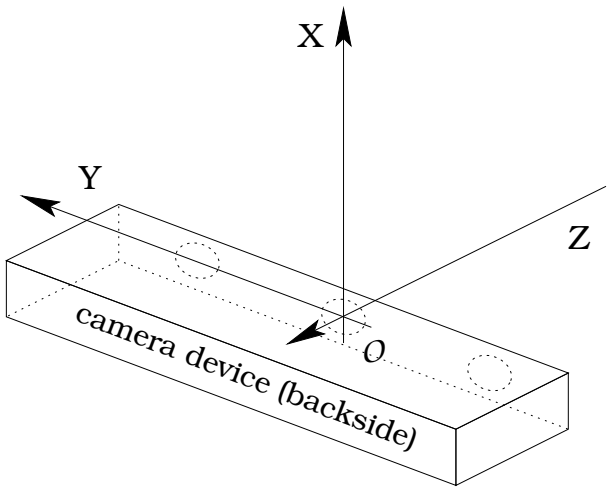


Figure 3.2: OPTOTRAK coordinate system

3.1.3 Subject centred

Our own system-inherent coordinate system is centred on the speaker or more precisely on the ellipsoid mesh representing the speaker's head in the tracking. It is based on the DICOM (**D**igital **I**maging and **Co**mmunications in **M**edicine) standard coordinate system as specified in Nat (2003), PS 3.3, C.7.6.2.1.1 (page 237):

> The direction of the axes is defined fully by the patient's orientation. The x-axis is increasing to the left hand side of the patient. The y-axis is increasing to the posterior side of the patient. The z-axis is increasing toward the head of the patient. The patient based coordinate system is a right handed system, i.e. the vector cross product of a unit vector along the positive x-axis and a unit vector along the positive y-axis is equal to a unit vector along the positive z-axis.

The origin lies in the centre of the ellipsoid (see Figure 3.3). Since in this case we can rename the axis with terms used in anatomy, we happily do so in order to reduce a little bit the danger of confusion generated by multiple coordinate system's x, y-, and z-axis. Table 3.1 shows the designations for axes and planes of the subject centred coordinate system.

axes	transversal	≡	x-axis
	anterior-posterior	≡	y-axis
	longitudinal	≡	z-axis
planes	axial	≡	xy-plane
	coronal	≡	xz-plane
	sagittal	≡	yz-plane

Table 3.1: Names of axes and planes in the subject centred coordinate system

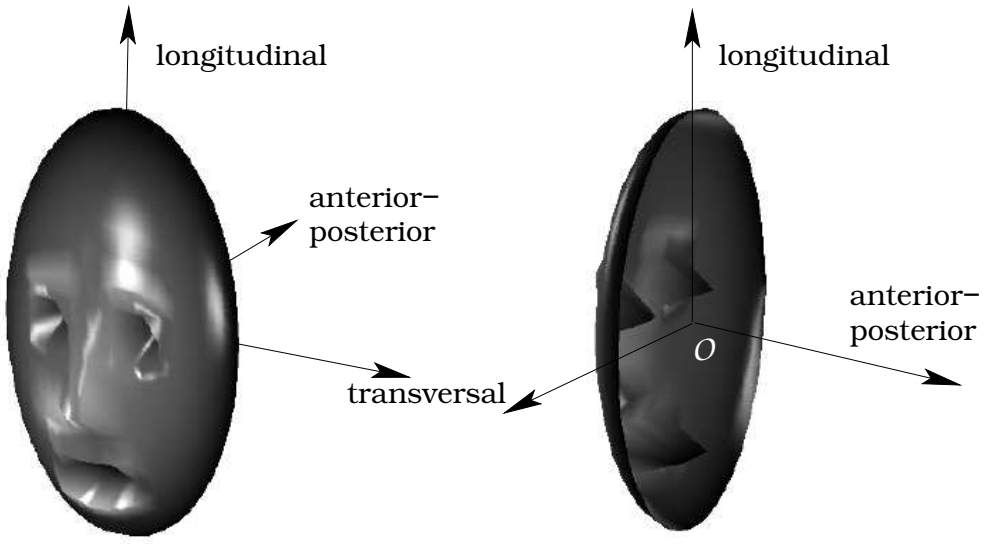


Figure 3.3: Subject centred coordinate system

3.1.4 'World'

When dealing with physical objects or virtual objects representing physical ones in several different coordinate systems, it is convenient to have a single 'world' coordinate system that acts as a global frame of reference anchored in the 'real world'. In the case of head motion data provided by OPTOTRAK (or similar robust and accurate tracking devices) we could just use the OPTOTRAK coordinate system. However, this option is not available in the case of video-based head tracking and besides this it simplifies the perspective projection equations if the world coordinate system is centred at the (assumed) focal point of the camera. Accordingly our world coordinate system is tightly attached to the single static video camera and has its origin at the focal point of the ideal pinhole camera modelling the video camera.

The x-axis is oriented horizontally increasing towards the right hand side of the camera, the y-axis is also horizontal in line with the optical axis and increases into the scene, and the z-axis is the vertical axis increasing upwards. In fact, the world coordinate system is the same as the subject centred one up to a translation offset if the subject/ellipsoid mesh is straightly facing the camera (see Figure 3.4).

3.1.5 Implementation issues

Since dealing with several coordinate systems can very easily turn into an implementation nightmare and is an inexhaustible source of potential hard-to-detect errors, we decided to represent the coordinate system as classes in MATLAB. More precisely there is a general parent class `coordinate_system` and several child classes that implement the above described specific coordinate systems (e.g., `video_image` or `world_reference`). The coordinate transformation of, for instance, the ellipsoid mesh (represented by a class itself) can then 'simply' be done by using a method (function) of the `coordinate_system` class that we named `transform`: It takes as first argument the new coordinate system (an

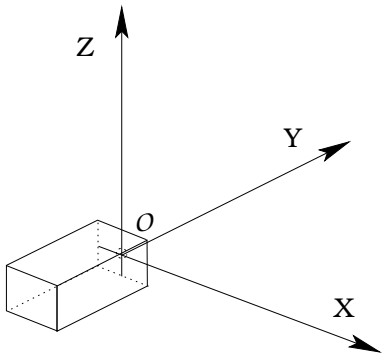


Figure 3.4: World reference coordinate system

object instance of the appropriate class) and as second the `ellipsoid_mesh` object. The `ellipsoid_mesh` object has as one of its attributes its current coordinate system. The whole transformation process becomes basically one line:

```
mesh_obj=transform(coord_sys,mesh_obj)
```

Creating the specific instance of the coordinate system class, however, still requires heightened attention, since some of the parameters defining the relationship between two arbitrary coordinate systems cannot be fixed in a permanent way beforehand - for example the transformation between the image coordinate system and the world coordinate system depends on the results of the camera calibration (see below). Additionally the video image was represented within the system as a class which allows capsulating its behaviour and its accessing by subroutines depending on whether frames or fields were its original source. See Zimmer and Bonz (1995) for a general account of object-oriented programming in digital image processing.

3.2 Initialisation

Figure 3.5 shows a schematic overview of the several stages of the initialisation procedure. We will closely follow the graph with our description by dedicating a section to each of the boxes in the procedure scheme.

3.2.1 External head tracking data

In section 2.1.2 we described several existing methods for video-based head tracking. It should be clear that in spite of fulfilling their purpose in general within the given task they are far from being perfect, that is, their accuracy was seldom evaluated beyond visual inspection. However, even small errors in the head tracking can completely spoil the measurement of the 'real' or implicit face motion. The reader may compare the displacement magnitude of, say, the raising of the corners of the mouth to a simple nodding movement, not to speak of the effect of movements of the whole upper body.

Very early in the development of this algorithm it was thus decided to bypass the head tracking problem for the time being by using external head motion tracking. The particular system chosen was OPTOTRAK in combination with a special

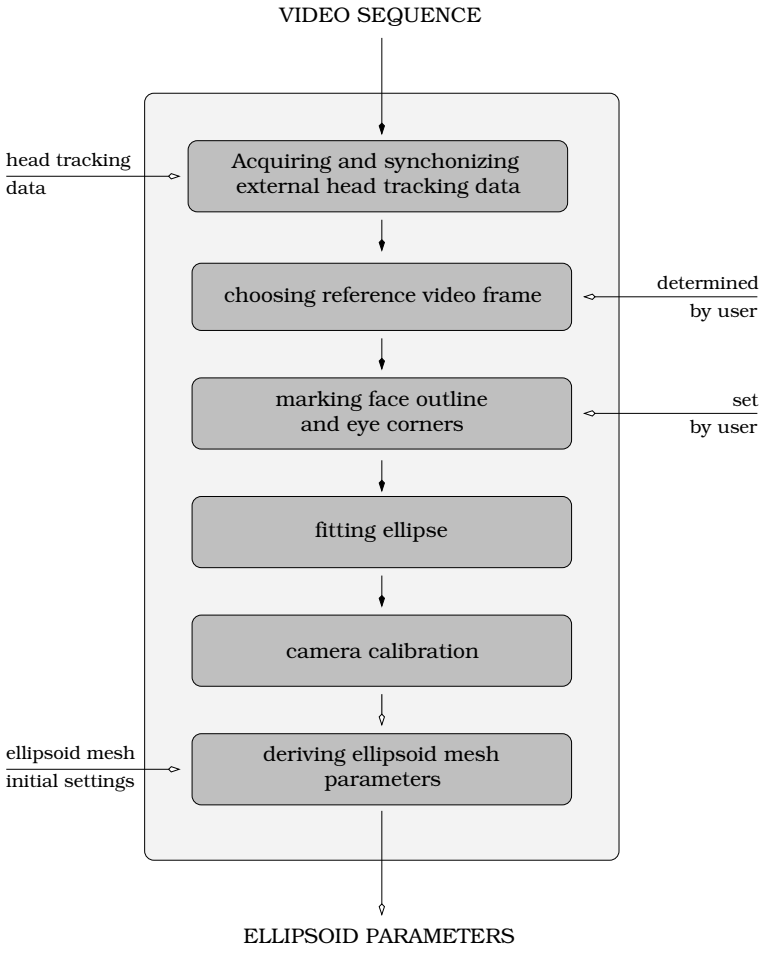
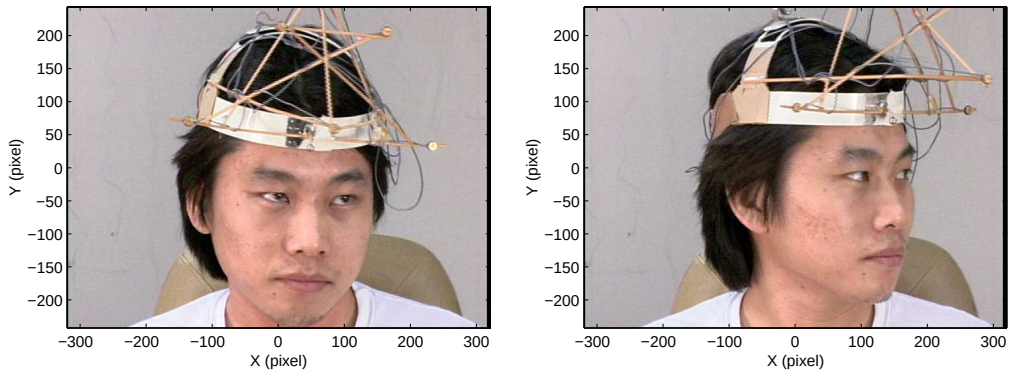


Figure 3.5: Scheme of the initialisation procedure

headmount (see Figure 3.6). The headmount consisted of a skeleton structure of thin bamboo sticks that carried six OPTOTRAK sensors ensuring that the sensors were spread out in all three spatial dimensions.

OPTOTRAK's own software allows the construction of a rigid body object from the sensors and calculates its rotation and translation values over the time course of the recording. Additionally it can try to determine the real centre of rotation compared to just taking the centroid of the set of sensors. The choice of the centre of rotation does not influence the rotation values of the rigid body but it may have a substantial effect on the translation values. The procedure, however, is always prone to some error for several reasons. Firstly, in case of head movements there is not a single point that acts as a centre of rotation for the head, since no single joint but the combined action of eight joints of complex geometry of the upper and lower cervical spine accomplishes the movement (see Zatsiorsky, 1998, page 326-336). The contribution of any single joint to the overall head-neck movement varies with the kind and amount of the executed movement. Secondly, even without knowing details about the algorithm implemented by Northern Digital it is clear that an extended range of movement is needed in order to ensure accuracy.



(a) Front view (b) Side view

Figure 3.6: Subject with headmount.

In the usual lab situation with the subject reading the stimuli from some kind of prompting device head movements will be small. The error from this source, however, can be minimised by including extra trials with random but deliberately large head movements. Trivially the calculated centre of rotation is despite its error better than any arbitrary choice.

The OPTOTRAK signal converted to rigid body motion parameters has to be synchronised with the video sequence. One way to achieve this is to use the capability of the OPTOTRAK system to record synchronously with the motion measurement an arbitrary analogue signal. If the acoustic signal of the speaker is recorded, it can be perfectly aligned with the audio channel of the video camera using cross-correlation.

The first step of the initialisation procedure reads in the head motion tracking data and synchronises them with the video sequence.

3.2.2 Determining a reference frame

For every input video sequence a *reference frame* must be determined, i.e. a frame where the motion tracking should be started with the undeformed ellipsoid mesh and from which the motion measurement proceeds frame by frame either forwards or backwards. Currently the choice is left to the user, but one could easily imagine basing it on some automatically verifiable criteria or just the first frame where a face detection algorithm finds the face of the speaker in the video frame. There is only one essential criterion for the reference frame and that is that it should contain the face to be examined in a more or less frontal view, since the ellipsoid mesh can be only rotated in the image plane (around the optical axis) during the fitting process (see section 3.2.4). If the tracking results are intended to be visualised as animation it is favourable if the speaker's mouth is slightly opened in the reference frame in order to have a representation of the area within the opened mouth in the extracted reference *texture map* (see section 4.1).

The second step of the procedure displays the first frame for which head tracking data are available and than asks the user whether or not to use this frame as reference frame. If the user declines, she can search the entire video sequence by

requesting the display of any frame that has head tracking data assigned.

3.2.3 Marking the face

The size of the ellipsoid mesh along its transversal and longitudinal axis can be directly adjusted to the subject's face by determining the face outline in the reference frame (remember that we required a frontal view of the face). The cross-section of an ellipsoid in the coronal plane is an ellipse (as any cross-section parallel to one of the major planes). If the coronal cross-section contains the centre point of the ellipsoid, the ellipse will have maximum size. Therefore an ellipse fit to the face in the image is an ideal way to determine the length of the transversal and longitudinal axis of the ellipsoid. This could be done either manually with an ellipse drawn by the user, which is rather tedious, or automatically by means of a face detection routine. In fact, many face detection programmes return as their output an ellipse fit to the face (see for example Nefian, Khosravi, and Hayes, 1997). However, they might not always work with the precision we would like to have guaranteed here (after all they are usually conceived to fulfil a different purpose).

We tried to strike a happy balance for our system insofar as we prompt the user to mark a few points on the face outline and then fit an ellipse automatically to these points. Additionally the user has to mark the outer or inner eye corners of the subject. The reason for this is explained in the next section. Figure 3.7 shows an example.

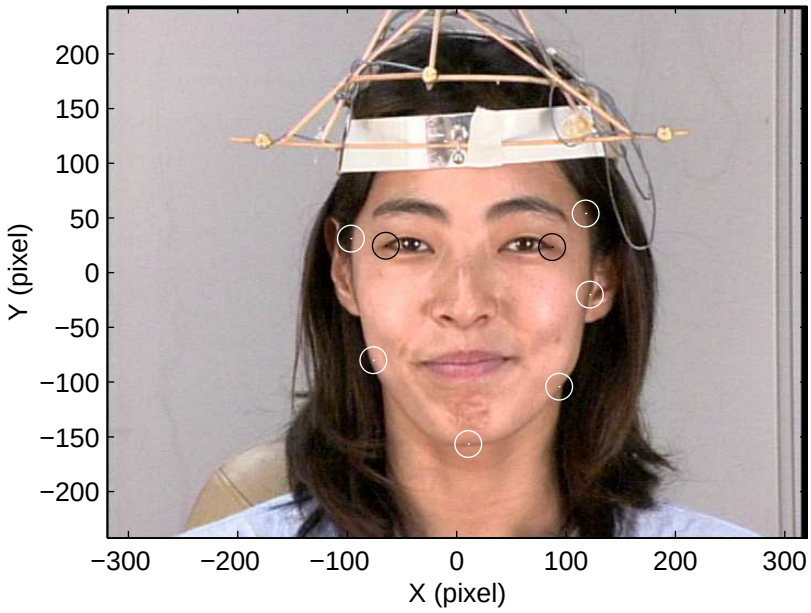


Figure 3.7: Subject with face outline and eye corner marking points

3.2.4 Ellipse fitting

It might come as a surprise that there existed no direct ellipse-specific method to fit an ellipse to a set of points before the publication of Fitzgibbon, Pilu, and Fisher (1996). As reviewed therein earlier methods were either linear or quadratic general cone fitting methods that were only biased towards an ellipse by applying different constraints or iterative, i.e. indirect methods.

The normal form of the ellipse is

$$\frac{x^2}{k^2} + \frac{y^2}{l^2} = 1 \tag{3.1}$$

where k and l equal the length of the semimajor axis and the semiminor axis, respectively. However, the normal form assumes that the centre of the ellipse lies at the origin of the coordinate system, and that the ellipse is oriented in such that its axes are aligned with the axes of the coordinate system. Thus it cannot be used for ellipse fitting in general.

As is evident from above we need to estimate five parameters: major and minor axis, centre coordinates in two dimensions, and orientation. Therefore we switch to the representation of the ellipse as a general conic by a second order polynomial

$$ax^2 + bxy + cy^2 + dx + ey + f = 0 \tag{3.2}$$

The general conic represents not necessarily an ellipse, it includes hyperbolas and parabolas as well. More precisely it will be an ellipse only if

$$b^2 - 4ac < 0 \tag{3.3}$$

Let $\mathbf{p}_i = [x_i, y_i]^T$ represent a single point in the set of N image points $\mathbf{p}_1 \dots \mathbf{p}_N$ marked by the user, let $\mathbf{x} = [x^2, xy, y^2, x, y, 1]^T$ and gather the polynomial coefficients in $\mathbf{a} = [a, b, c, d, e, f]^T$. Then (3.2) could be expressed as

$$f(\mathbf{p}, \mathbf{a}) = \mathbf{x}^T\mathbf{a} = 0 \tag{3.4}$$

Now a general cone can be fitted to the image points in the least-squares sense by minimising

$$\min_{\mathbf{a}} \sum_{i=1}^{N} (\mathfrak{D}(\mathbf{p}_i, \mathbf{p}_a))^2 \tag{3.5}$$

where $\mathfrak{D}(\mathbf{p}_i, \mathbf{p}_a)$ is a suitable distance, e.g. the *algebraic distance* (see Trucco and Verri, 1998, page 101). Since the *algebraic distance* of a point $\mathbf{p}$ from a curve $f(\mathbf{p}, \mathbf{a}) = 0$ is $|f(\mathbf{p}, \mathbf{a})|$, (3.5) becomes

$$\min_{\mathbf{a}} \sum_{i=1}^{N} |\mathbf{x}_i^T\mathbf{a}|^2 \tag{3.6}$$

So far we would still fit a general conic to the data points that would not be forced to be an ellipse. In addition the trivial solution $\mathbf{a} = [0,0,0,0,0,0]^T$ is not excluded. Since the inequality constraint (3.3) is difficult to solve (see Fitzgibbon, Pilu, and Fisher, 1999) and parameter vector $\mathbf{a}$ is defined only up to a scale factor we can reformulate it as an equality constraint:

$$b^2 - 4ac = -1 \tag{3.7}$$

Since the details of the solution to this problem are beyond the scope of this thesis we will only give a coarse description of the subsequent procedure in the next paragraph following the presentation in Trucco and Verri (1998, page 104). Using matrix notation (3.7) becomes

$$\begin{bmatrix} a & b & c & d & e & f \end{bmatrix} \begin{bmatrix} 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = \mathbf{a}^T \mathbf{C}\, \mathbf{a} = -1 \tag{3.8}$$

$\mathbf{C}$ is called the *constraint matrix*.

With

$$\mathbf{D} = \begin{bmatrix} x_1^2 & x_1 y_1 & y_1^2 & x_1 & y_1 & 1 \\ x_2^2 & x_2 y_2 & y_2^2 & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_N^2 & x_N y_N & y_N^2 & x_N & y_N & 1 \end{bmatrix} \tag{3.9}$$

called the *design matrix* minimisation problem (3.7) can be rewritten as

$$\min_{\mathbf{a}} \|\mathbf{a}^T \mathbf{D}^T \mathbf{D}\, \mathbf{a}\| \quad \text{or shorter as} \quad \min_{\mathbf{a}} \|\mathbf{a}^T \mathbf{S} \mathbf{a}\| \tag{3.10}$$

where $\mathbf{S}$ is called the *scatter matrix*. Differentiating the minimisation function with the use of the Lagrange multiplier λ and re-ordering the terms results in (see Fitzgibbon et al., 1999, for the single steps of the derivation)

$$\mathbf{S}\mathbf{a} = \lambda \mathbf{C}\mathbf{a} \tag{3.11}$$

The solution for problem (3.10) under the constraint (3.7) is found by determining the only negative *eigenvalue* of the *general eigenvalue problem* posed by (3.11).

Many numerical packages provide functions to solve the eigenvalue problem, for example in MATLAB the solution would be found by

```
[A,V]=eig(S,C)
```

where S and C are the scatter and the constraint matrix as defined above. Diagonal matrix V contains the eigenvalues and the columns of A the corresponding eigenvectors. MATLAB is in general able to cope with the rank suffiency of

C and **S**, however there are exceptions that lead either to no solution at all or several solutions. Fitzgibbon et al. (1999) presents a numerically more stable version of their first algorithm from Fitzgibbon et al. (1996) by basically partitioning (block decomposition) the scatter and the constraint matrix as suggested by Halir and Flusser (2000).

The improved algorithm can also handle a special case: if all data points lie perfectly on an ellipse, the older version of the algorithm breaks down completely (see Halir and Flusser, 2000). This is a rare situation in 'real world' applications, since usually one would like to fit an ellipse to set of data points that are disturbed by noise and hence do not lie exactly on an ellipse even though they might originate from one; see Halir and Flusser (2000) for an interesting example of estimating the diameter of an archaeological pot from a single fragment.

Rosin (1999) showed that constrained least-squares fitting of ellipses performs well if the noise is essentially Gaussian, but breaks down very early if there are non-Gaussian outliers. In our case of fitting an ellipse to a face in a video image it is difficult to make any estimate about the nature of the noise, since we simply do not have an underlying ellipse: the face outline does not conform to an ellipse. Assuming that there is at least an ellipse which approximates the face outline in the best way, one must doubt that the deviation of the user marked points from it would follow a Gaussian distribution considering for instance the jaw region. However, none of the discussed methods in Rosin (1999), e.g., Theil-Sen estimator, **L**east **Med**ian of **S**quares (LMedS) estimator, could be used here, because they all need a larger set of data points while we would like to keep the set of points as small as possible to avoid the initialisation process becoming too cumbersome for the user. Therefore we implemented the improved version of the Fitzgibbon-Fischer-Pilu algorithm.

Surprisingly the results were not very satisfying. Though of course an optimal ellipse was fit to the data points, it was not covering the face in such a way that one would have liked to base the ellipsoid mesh on it: in most cases the orientation seemed to be wrong. Two reasons are responsible for that:

i. As described in all the papers cited above least-squares fits are biased towards thinner ellipses.

ii. Very often points could not be set to cover the whole face. Either the top part of the face lay outside the video or a judgement could not be made because of hair occluding stretches of the upper face outline (for bearded subject this might also be the case for the lower part of the face). In addition there is a general difficulty in deciding what should be assigned to the face and what not (e.g. end with forehead versus including full scull). For the face motion tracking this is almost irrelevant, and even more so for the fitting algorithms: all of them can cope in an excellent way with data points which are scattered over the potential ellipse and completely unbalanced, e.g, data points covering only a very small part of the full ellipse. However, the ellipse is fit to the points marked by the user, whatever their location might be, and not to the face outline.

The combined impact of *i.* and *ii.* could easily lead to a thinner than expected ellipse with an orientation apparently tilted with respect to the perceived face orientation being fit to the unbalanced data point set.

Remarkably, an *ad hoc* ellipse fitting method[1] we had conceived and implemented in an early version of the face motion tracking system

[1] The algorithm attempted to find first the centre of the desired ellipse and then the remaining

(Kroos, Kuratate, and Vatikiotis-Bateson, 2000) that did not return the best fitting ellipse under certain conditions nevertheless seemed to perform subjectively better. In this method any one of the five intrinsic parameters could be fixed and subsequently excluded from the optimisation process. We had applied this on the ellipse orientation by fixing the minor axis to be parallel to the virtual line connecting the outer or inner eye corners of the subject. It seemed to be that this restriction on ellipse orientation was responsible for the subjectively better results.

Accordingly we decided to include the feature in the direct least-squares fitting. The effort for the user does not increase much, after all the eye corners are easy to detect and mark and only two points are necessary. Furthermore saving this information facilitates considerably aligning the motion tracking results of several trials from the same subject or even different subjects later on. Unfortunately it is not straightforward to fix one of the intrinsic parameters, since the Fitzgibbon-Fischer-Pilu algorithm operates on the conic coefficients.

However, setting conic coefficient b to 0 results in the fitting of an ellipse with its axis aligned to the coordinate system axis, i.e. no rotation is attempted. This can be easily shown by looking at the formulae that allow the transformation from the conic representation back to the normal form (see Bronstein, Semendjajew, Musiol, and Mühlig, 1999, for tables that cover all quadratic curves, i.e. conic sections), and in particular the equation for the rotation angle α that rotates the coordinate system given by

$$\tan 2\alpha = \frac{2b}{a - c} \tag{3.12}$$

If $b = 0$, α will always be 0 as well, independent of the values of a and c.

This means that before submitting the data points to the fitting procedure they have to be rotated by the inverse of the angle between the line connecting the eye corners and either the x- or the y-axis with the standard two-dimensional rotation formula

$$\begin{bmatrix} \mathbf{x}_r^T \\ \mathbf{y}_r^T \end{bmatrix} = \begin{bmatrix} \cos -\alpha & -\sin -\alpha \\ \sin -\alpha & \cos -\alpha \end{bmatrix} \begin{bmatrix} \mathbf{x}^T \\ \mathbf{y}^T \end{bmatrix} \tag{3.13}$$

where $\mathbf{x}^T$ and $\mathbf{y}^T$ are row vectors with the x and y coordinates of the original data points and $\mathbf{x}_r^T$ and $\mathbf{y}_r^T$ their rotated counterparts. Then parameter b is removed from the conic coefficient vector $\mathbf{a}$ and the constraint matrix $\mathbf{C}$ adjusted by

---

ellipse parameters in a second run using the ellipse definition as the geometrical place of all points for which the sum of the distances from two given points, the focal points, is constant:

$$r_1 + r_2 = 2a$$

where r_1 is the distance of any point belonging to the ellipse from the first focal point, r_2 the distance from the second focal point, and a the length of the major axis.

Calculating r_1 and r_2 based on the estimated ellipse parameters and the given data points the equation above holds only if all parameters a correctly estimated and no noise corrupts the data points. Otherwise an error remains. This error, or more precisely the mean squared error, was iteratively minimised using the downhill simplex optimisation method (see Nelder and Mead, 1965) to find the best fitting ellipse.

excluding the exigencies for parameter b:

$$\mathbf{C} = \begin{bmatrix} 0 & -2 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.14}$$

Afterwards the direct least-squares fitting algorithm is carried out as before. Upon finishing the found centre coordinates have to be rotated by α (the length of the major and minor axes are of course invariant to rotation). Figure 3.8 shows the previous example (Figure 3.7) with the ellipse fit.

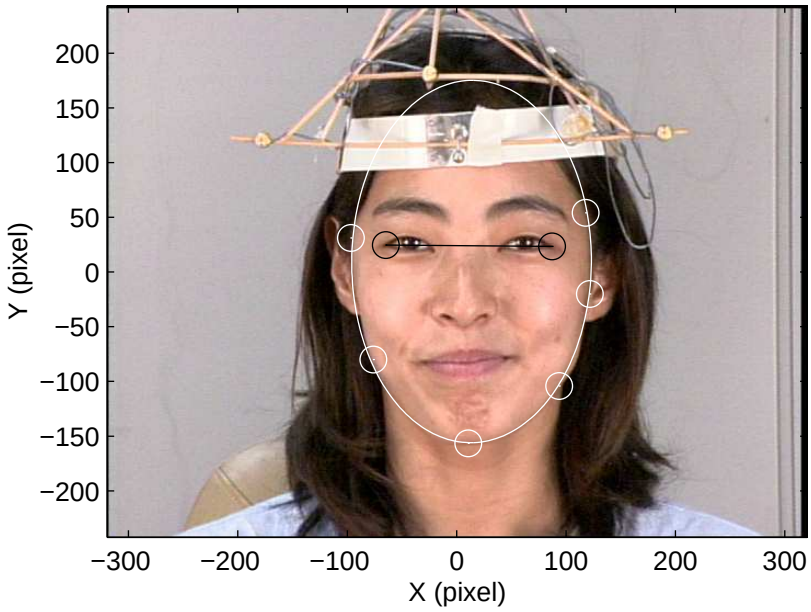


Figure 3.8: Ellipse fit to the face

3.2.5 Camera model and camera calibration

As shown in section 2.2 an algebraic model for perspective imaging is given by the following equation:

$$\mathbf{P}_I = \mathbf{C}_{W \to I} \, \mathbf{P}_W \tag{3.15}$$

where $\mathbf{P}_W$ is a set of points in the 'real world' coordinate system, $\mathbf{P}_I$ the corresponding points in the image, and $\mathbf{C}_{W \to I}$ the camera model matrix. Explicitly

written for a single point this becomes

$$\begin{bmatrix} s\,\mathcal{P}_{Ir} \\ s\,\mathcal{P}_{Ic} \\ s \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & 1 \end{bmatrix} \begin{bmatrix} \mathcal{P}_{Wx} \\ \mathcal{P}_{Wy} \\ \mathcal{P}_{Wz} \\ 1 \end{bmatrix} \tag{3.16}$$

where s is the perspective scaling factor, $\mathcal{P}_{Ir}$ and $\mathcal{P}_{Ic}$ the image row and column coordinates of the transformed three-dimensional 'real world' point with coordinates $\mathcal{P}_{Wx}$, $\mathcal{P}_{Wy}$, and $\mathcal{P}_{Wz}$ respectively, and c the camera model coefficients. Eliminating the homogeneous scaling factor s (using the last row in (3.16) and the dot product of vectors) leads to

$$\begin{aligned} \mathcal{P}_{Ir} &= \frac{[c_{11}\ c_{12}\ c_{13}\ c_{14}] \odot [\mathcal{P}_{Wx}\ \mathcal{P}_{Wy}\ \mathcal{P}_{Wz}\ 1]}{[c_{31}\ c_{32}\ c_{33}\ 1] \odot [\mathcal{P}_{Wx}\ \mathcal{P}_{Wy}\ \mathcal{P}_{Wz}\ 1]} \\ \mathcal{P}_{Ic} &= \frac{[c_{21}\ c_{22}\ c_{23}\ c_{24}] \odot [\mathcal{P}_{Wx}\ \mathcal{P}_{Wy}\ \mathcal{P}_{Wz}\ 1]}{[c_{31}\ c_{32}\ c_{33}\ 1] \odot [\mathcal{P}_{Wx}\ \mathcal{P}_{Wy}\ \mathcal{P}_{Wz}\ 1]} \end{aligned} \tag{3.17}$$

Therefore we have 11 parameters that would have to be estimated. Assuming that we have n points where the 'real world' coordinates and their corresponding image coordinates are known, we could obtain n pairs of calibration equations of the form

$$\begin{aligned} \mathcal{P}_{Ir} &= (c_{11} - c_{31}\,\mathcal{P}_{Ir})\,\mathcal{P}_{Wx} + (c_{12} - c_{32}\,\mathcal{P}_{Ir})\,\mathcal{P}_{Wy} \\ &\quad + (c_{13} - c_{33}\,\mathcal{P}_{Ir})\,\mathcal{P}_{Wz} + c_{14} \\ \mathcal{P}_{Ic} &= (c_{21} - c_{31}\,\mathcal{P}_{Ic})\,\mathcal{P}_{Wx} + (c_{22} - c_{32}\,\mathcal{P}_{Ic})\,\mathcal{P}_{Wy} \\ &\quad + (c_{23} - c_{33}\,\mathcal{P}_{Ic})\,\mathcal{P}_{Wz} + c_{24} \end{aligned} \tag{3.18}$$

Usually we would have more than $ceil(11/2)$, i.e., 6 calibration points and thus an overdetermined linear equation system, though disturbed by measurement noise and noise in the process of determining the point-to-point correspondence between 'real world' and image. Therefore a least-squares solution appears to be appropriate (see Shapiro and Stockman, 2001).

By rearranging equations (3.18) to separate the knowns from the unknowns

we get

$$\begin{bmatrix} \mathcal{P}_{Wx}, \mathcal{P}_{Wy}, \mathcal{P}_{Wz}, 1, 0, 0, 0, 0, -\mathcal{P}_{Wx}\mathcal{P}_{Ir}, -\mathcal{P}_{Wy}\mathcal{P}_{Ir}, -\mathcal{P}_{Wz}\mathcal{P}_{Ir} \\ 0, 0, 0, 0, \mathcal{P}_{Wx}, \mathcal{P}_{Wy}, \mathcal{P}_{Wz}, 1, -\mathcal{P}_{Wx}\mathcal{P}_{Ic}, -\mathcal{P}_{Wy}\mathcal{P}_{Ic}, -\mathcal{P}_{Wz}\mathcal{P}_{Ic} \end{bmatrix} \begin{bmatrix} c_{11} \\ c_{12} \\ c_{13} \\ c_{14} \\ c_{21} \\ c_{22} \\ c_{23} \\ c_{24} \\ c_{31} \\ c_{32} \\ c_{33} \end{bmatrix} = \begin{bmatrix} \mathcal{P}_{Ir} \\ \mathcal{P}_{Ic} \end{bmatrix} \tag{3.19}$$

Gathering all available calibration points in a matrix, equation (3.18) can be represented as

$$\mathbf{A}_{2n\times 11}\,\mathbf{x}_{11\times 1} \approx \mathbf{b}_{2n\times 1} \tag{3.20}$$

where $\mathbf{A}_{2n\times 11}$ is the matrix with knowns corresponding to the leftmost matrix of (3.19), $\mathbf{b}_{2n\times 1}$ the row vector with concatenated known image coordinates of the calibration points corresponding to the right hand side of (3.19) and $\mathbf{x}$ the column vector with the unknowns in exactly the same form as in (3.19). (3.20) can be solved for $\mathbf{x}$ with numerical methods, in MATLAB this is simply stated as

```
x = A \ B
```

(see Shapiro and Stockman, 2001, page 423f, for more details).

Let us now consider the situation encountered in the initialisation phase of our face motion tracking system. In general we assume an uncalibrated camera in order to keep the intended wide range of possible applications of the method. In case of head tracking data coming from OPTOTRAK some or all of the sensors will most likely be visible in the image and could be employed for calibration, though the minimum requirement of 6 calibration points will almost never be fulfilled, either because less than 6 sensors were used on the headmount or not all of them lie within the video frame. In the video data from several experiments used for this thesis never more than 3 sensors were accessible for the calibration, and very often only 2.

Therefore we have to make additional assumptions, for instance on the camera pose relative to the 'real world' coordinate system. Note that of the 11 parameters in (3.19) only 9 are independent; we have three rotational and three translational parameters defining the camera pose, one scaling parameter relating continuous horizontal real image coordinates to pixel columns (d_x), another scaling parameter expressing the relationship for vertical real image coordinates and pixel rows (d_y), and finally focal length (f).

First we discard the external camera parameters by defining the world system as described in section 3.1.4. Of course this does not solve the problem but rather shifts it to determining the pose of the OPTOTRAK device relative to the world coordinate system. However, we may assume that video and OPTOTRAK camera were as closely as possible aligned in the experiment, since they should catch

very similar aspects of the subjects behaviour via a direct optical pathway. Since it is technically impossible that they are exactly in the same spot, some deviation is unavoidable, but being small it could be neglected or otherwise manually measured by the experimenter.

Because we do not model the effects of radial distortion and spherical aberrations we have only three internal parameters, from which the two scaling parameters relating real image coordinates to pixel size (d_x and d_y) are actually the same since we deal only with square pixels at this stage (non-square pixels had been converted in the preprocessing).

Using homogenous coordinates let $\mathbf{F}_{W\to F}$ be

$$\mathbf{F}_{W\to F} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{f} & 1 \end{bmatrix} \tag{3.21}$$

where F is depending only on the focal length, and let $\mathbf{S}_{F\to I}$ be

$$\mathbf{S}_{F\to I} = \begin{bmatrix} 0 & \frac{1}{d_y} & 0 \\ \frac{1}{d_x} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.22}$$

then the relationship between 'real world' coordinates and and camera coordinates in pixels in our simplified case could be expressed as (dropping the third row of $\mathbf{S}_{F\to I}$ as shown for equation (2.12))

$$\mathbf{P}_I = \mathbf{S}_{F\to I}\,\mathbf{F}_{W\to F}\,\mathbf{P}_W \tag{3.23}$$

Accordingly equation (3.16) simply becomes

$$\begin{bmatrix} s\,\mathcal{P}_{Ir} \\ s\,\mathcal{P}_{Ic} \\ s \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{d} & 0 & 0 \\ \frac{1}{d} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 1 \end{bmatrix} \begin{bmatrix} \mathcal{P}_{Wx} \\ \mathcal{P}_{Wy} \\ \mathcal{P}_{Wz} \\ 1 \end{bmatrix} \tag{3.24}$$

The index of the d-parameter was dropped, because of the above mentioned property of square pixels. After the same transformation as above with the full calibration matrix we get analogously to (3.19) for a single point:

$$\begin{bmatrix} \mathcal{P}_{Wy} & -\mathcal{P}_{Wz}\mathcal{P}_{Ir} \\ \mathcal{P}_{Wx} & -\mathcal{P}_{Wz}\mathcal{P}_{Ic} \end{bmatrix} \begin{bmatrix} \frac{1}{d} \\ \frac{1}{f} \end{bmatrix} = \begin{bmatrix} \mathcal{P}_{Ir} \\ \mathcal{P}_{Ic} \end{bmatrix} \tag{3.25}$$

Thus we obtain two equations from every calibration point to estimate two calibration parameters. Using a least-squares method we can now get a solution with only two OPTOTRAK sensors visible in the reference frame, albeit of a little bit doubtful reliability. Three or more sensors, however, should yield sufficiently good results. Note, however, that the least-squares solution tends to become instable, if the distance between the OPTOTRAK and the video camera is not really

very small or not appropriately measured. In this case the calibration can be expanded again to include at least the x- and y-translation parameters.

3.2.6 The ellipsoid mesh

We are now able to reverse the effect of perspective projection for the fit ellipse of section 3.2.4. Keep in mind that all ellipse parameters refer to the image coordinate system, but for the motion tracking it would be highly desirable if the results were independent of the projection. In case of external head tracking data from OPTOTRAK it is actually necessary to create the ellipsoid mesh outside the image space in order to be able to drive it with the motion data obtained in the OPTOTRAK reference frame. With our very primitive camera model this amounts only to scaling of the length of the two principal axes.

The ellipsoid mesh we using is a parametrised half-ellipsoid. The ellipsoid is one of the 17 standard-form types of the *quadratic surface* (also called *quadric*). Its normal form is given by

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \tag{3.26}$$

a,b, and c are the length of the semi-axes. To create the ellipsoid mesh the parametric equation is used

$$\begin{aligned} x &= a \cos\theta \sin\phi \\ y &= b \sin\theta \sin\phi \\ z &= c \cos\phi \end{aligned} \tag{3.27}$$

for azimuth angle θ ranging from 0 to 2π and polar angle ϕ from $-\pi/2$ to $\pi/2$. To approximate the facial surface a half-ellipsoid is sufficient, therefore the azimuth angle can be limited to the interval from 0 to π instead of 2π.

The mesh itself is represented by the coordinates of its nodes, for example in MATLAB with a three-dimensional matrix. The arrangement along the first two dimensions (rows and columns) determines the topology of the mesh, the location of the nodes relative to each other, while the third dimension (we may call it 'slices') represents the three spatial dimensions). Only a few lines of source code are needed to build the mesh from scratch

```
theta=linspace(0,pi,res_long);
phi=linspace(-pi/2,pi/2,res_tran);
```

The above generates vectors with linearly equally spaced azimuth and polar angles. The last input argument is the number of nodes along the longitudinal or transversal axis of the mesh.

```
[PHI,THETA]=meshgrid(phi,theta);
DATA=zeros([size(THETA),3]);
```

Function `meshgrid` generates matrices from the vectors `theta` and `phi` in such a way that the rows of `PHI` are copies of the vector `phi` and the columns of `THETA` are copies of the vector `theta`. The second line initialises the mesh data matrix.

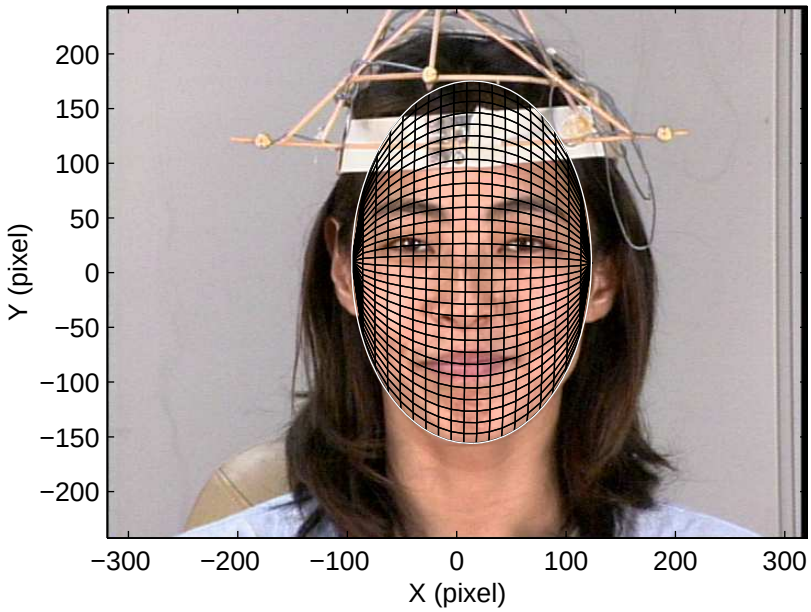


Figure 3.9: Ellipsoid mesh superimposed onto subject's face

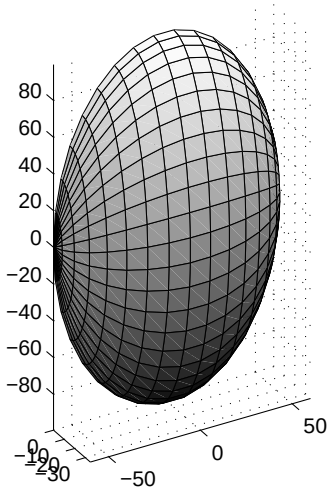
```
DATA(:,:,1)=tran*sin(PHI);
DATA(:,:,2)=ante*sin(THETA).*cos(PHI);
DATA(:,:,3)=long*cos(THETA).*cos(PHI);
```

Finally the parametric formula is applied. `tran`, `ante`, and `long` are the lengths of the transversal, anterior-posterior, and longitudinal semi-axes, respectively. Figure 3.9 shows a sample mesh superimposed onto the subject's face. The parameters were derived from the ellipse in Figure 3.8. For demonstrations purposes it was fit exactly to the ellipse; in a real tracking situation, however, it would be generally enlarged by a proportional factor to make sure that for example the jaw would not move outside of the tracked area during opening gestures.

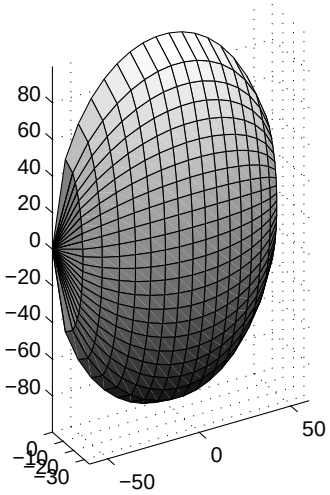
Note that the specific way in which the new data are assigned to the dimensions of the ellipsoid ensures that the vertex points where all longitude lines meet is at the side of the face, an area which does not have to be tracked unlike the important lower chin region.

The length values for the longitudinal and transversal axis are obtained from the principal axis of the ellipse fit to the face. The length of the anterior-posterior axis must be estimated, because there is no information in the image we could utilise to derive it.[2] On account of this the best solution would probably be to use a large database of faces represented as range data (e.g. using a laser range scanner) and examine whether or not the depth parameter could be determined

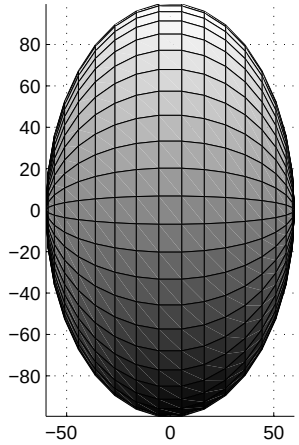
[2] We intentionally do not account here for methods of retrieving three-dimensional data from single images or image sequences, trying to resolve the so-called *shape from shading* and *structure from motion* problem. There are several techniques described in the computer vision literature (see Shapiro and Stockman, 2001; Trucco and Verri, 1998; Faugeras, 1993; Forsyth, 2003, for overviews), however, as far as no stereo camera system is used these methods lack sufficient reliability and more important the effort spent on it would be out of all proportions relative to the gain for our system.



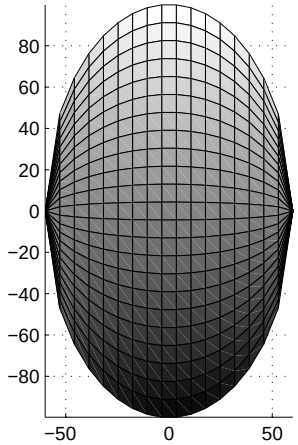
(a) Equal ϕ and θ, lateral view



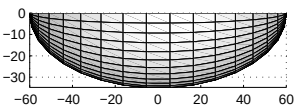
(b) Normalised ϕ and θ, lateral view



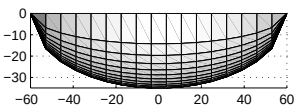
(c) Equal ϕ and θ, front view



(d) Normalised ϕ and θ, front view



(e) Equal ϕ and θ, top view



(f) Normalised ϕ and θ, top view

Figure 3.10: Ellipsoid meshes

depending on face width and length. Since we did not have access to such a database at the time of developing the system, we conceived an *ad hoc* estimation depending on the semi-major and semi-minor axis of the ellipse fit to the face given by

$$d_E = s\, e_{major} \frac{e_{major}}{e_{minor}} \tag{3.28}$$

where s is a scaling constant that can be modified in the program source code and is currently set to 0.2.

(3.28) has the effect to make the ellipsoid flatter for more rounded faces, since the ratio of the semi-major and semi-minor axis approaches 1, and more prolonged for thiner faces. The wisdom in doing so might be arguable, however, very fortunately the depth parameter of the ellipsoid is of subordinate importance in the motion tracking process. Firstly, the half-ellipsoid approximates the facial surface very crudely anyway, the error dwarfing at some locations in the face (e.g., the nose) any error made by using an even grossly wrong anterior-posterior length parameter. Secondly, the tracking is essentially two-dimensional. The ellipsoidal shape is mainly used to predict in a better way the effect that head shape and motion has on the appearance of the face in the video frame (i.e. perspective foreshortening when the angle between the surface normal and the optical axis of the camera becomes relatively large). During the motion tracking the mesh nodes are constraint to lie on the surface of the ellipsoid at all times.

The motion tracking is based on texture map segments whose size and location is defined using the mesh nodes (see section 3.3.4). Therefore the area enclosed by e.g. four neighbouring nodes should stay more or less constant over the mesh, i.e. should not be depending on the location of the area on the ellipsoid surface, to guarantee that approximately the same number of pixels is enclosed in every search segment. As can be seen from Figure 3.10(a) showing the ellipsoid mesh (in an orthographic projection) the area between four neighbouring nodes appears to become smaller as one moves away from the centre towards the edges.

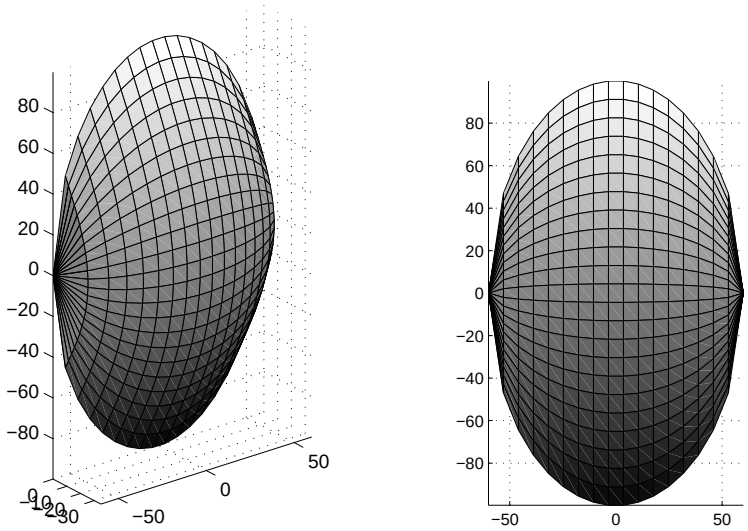
This is partly due to the foreshortening effect and in this way not only wanted but necessary. However, there is a second cause which is a result of the parametrisation. It can be observed when looking at the ellipsoid exactly from the top as in Figure 3.10(e). It appears as if there would be very little foreshortening[3] despite the strong curvature away from the camera exhibited by the ellipsoid from this viewpoint. In the parametrisation the polar angle was linearly equally spaced, but the ellipsoid for our purposes is not - and never will be - a sphere. Thus the lines connecting the nodes are compressed in the regions of higher curvature at the top and the bottom. A similar phenomenon is happening along the transversal axis. Additionally the converging lines towards the vertices at the sides lead to very small segments there. The latter cannot be compensated for without resampling the ellipsoid surface, something we would prefer to avoid for the increased complexity it causes in the further handling during the actual motion tracking.

A partial remedy - albeit not a perfect one - for the above mentioned shortcomings is to replace the equal spacing of the angular values in the basic matrices by a normalised spacing that in some respects is simply the inverse of the parametrisation formula. We used

[3] Remember that in orthographic projection the image plane coordinates equal their 'real world' counterparts: $x_I = x_W$ and $y_I = y_W$

```
theta=linspace(1,-1,res_long);
theta=acos(theta);
phi=linspace(-1,1,res_tran);
phi=asin(phi);
```

`acos` and `asin` are the MATLAB function names for the arccos and arcsin. Figure 3.10 shows a comparison of the described spacing variants from a slightly lateral view and from a front view. Both ellipsoidal meshes have the same number of nodes along corresponding axes. The improvement for the tracking of the normalised version is clearly visible in the front view: the area between four neighbouring nodes at the top and the bottom is about the same size as in the centre. But the side view makes clear that it is achieved by the expense of constancy of the area over the ellipsoid surface. The only resort here would be to give up the ellipsoid and turn to other quadrics, like the somewhat exotic, but beautifully spaced one in Figure 3.11.



(a) Lateral view (b) Front view

Figure 3.11: Ellipsoid alternative

After the ellipsoid is created its location on the anterior-posterior axis has to be estimated. If it was a real world object, we would like it to be as close to the face as the protective masks commonly worn in fencing (which are actually an ellipsoidal gauge mesh). However, in our virtual representation we could fit it even closer to the facial surface. Usually a distance value of the mesh centre to one of the headmount sensors is manually measured or just guessed. The resulting location is then visually checked using trials with considerable amount of head motion (see section 3.2.1), since problems in the positioning become immediately palpable as the head of the subjects turns towards a profile view. It should have become clear that by now we are able to use the OPTOTRAK head tracking data to steer the mesh to follow the head motion of the subject in the video sequence and visualise

the result by superimposing the projected mesh onto the video sequence.[4]

3.3 Motion tracking

A schematic overview over the tracking procedure is presented in Figure 3.12. We will follow the graph with our description in a little bit less rigid way than we did in the initialisation procedure.

3.3.1 Image preprocessing: Filtering with wavelets

The first step applied to any incoming frame is a two-dimensional discrete wavelet transformation. As outlined in section 2.3.2 this corresponds to applying a set of digital halfband filters to the image, where the filters have specific properties. At each level of the DWT the input signal is decomposed into a high frequency and a low frequency part, using a pair of highpass and lowpass filters, which are orthogonal to each other. The lowpass data are then used as the input signal to the next higher level. This insures that there is no redundant information in the low and high frequency components of one level, and hence no redundant information in the high frequency parts of the different levels.

However, human faces or more precisely images of human faces, exhibit static as well as dynamic features that are to a certain degree scale independent.

Figure 3.13 shows the grayscale images of the face of a human (the author of the thesis) and of a gibbon,[5] and their representation at two different wavelet levels: In the images of the centre row the three subbands of the 2nd level are overlaid, thus the images are residing entirely in the high spatial frequency domain. In the images at the bottom the subbands of the 4th level are superimposed and accordingly the frequencies remaining in the images are relatively low. Notice that for instance the lips are clearly recognisable at both scales, though less clearly for the gibbon in the low frequency image (Figure 3.13(f)). The main reason for it is that any reasonably sharp edge bounded by larger uniform image areas receives energy from a whole set of wavelet levels. The highest level still contributing is the one that contains the frequency whose wavelength equals twice the size of the larger of the two bounding areas, the lowest is limited by the sharpness of the edge.

The human eyes constitute a remarkable exception in the way that their overall shape is scale independent, but the interior consisting of pupil, iris and sclera emerges only in the higher spatial frequency domain (Figure 3.13(c)). The gibbon does not possess a white sclera, thus its eye is not as finely structured and does not appear as strongly in the high spatial frequencies (Figure 3.13(d)).

Given the importance for humans of being able to determining the gaze direction of another with great precision (Emery, 2000), it can be assumed that the peak in the high frequency domain aids in that task (while for monkeys and primates the head orientation gives sufficiently detailed clues about the gaze direction). This might in turn have consequences for auditory-visual speech. Because the resolution of human visual perception is varying widely across the field of view, being relatively low in the periphery and relatively high in the centre (fovea), every attempt to resolve high frequency details requires focusing the view on that

[4] We were actually asked once during a demonstration, where a video clip with the undeformed mesh following the head movements was shown, whether or not the subject felt uncomfortable having to wear this tight mask.

[5] The author is the one on the left.

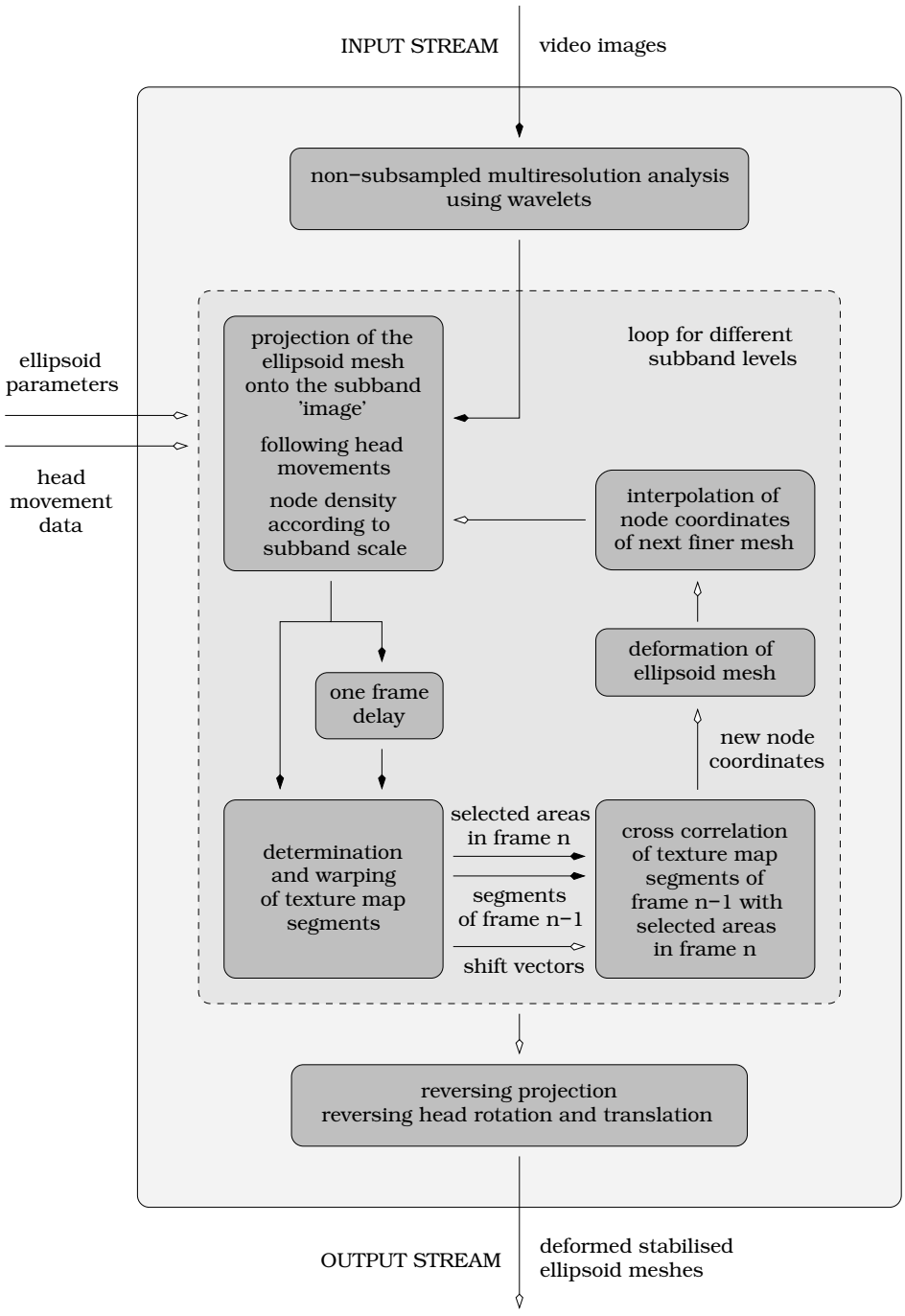


Figure 3.12: Scheme of the motion tracking procedure

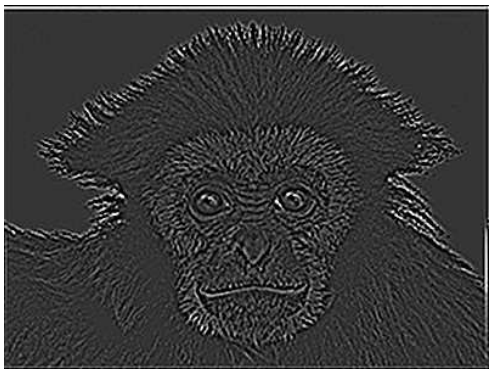
particular area (Wandell, 1995). Thus it would be evolutionarily useful if the perception and neural processing of facial speech movements would not rely on high frequency details to avoid competition with the gaze determining task.

Reasoning along these lines was indirectly confirmed by an exper-

(a) Original image

(b) Original image

(c) Level 2



(d) Level 2

(e) Level 4

(f) Level 4

Figure 3.13: Spatial frequency filtering: comparison between the face of human and gibbon

iment showing that subjects spend less time than expected viewing the speakers mouth even if the audio signal was severely degraded (Vatikiotis-Bateson, Eigsti, Yano, and Munhall, 1998). In spite of the auditorily unfavourable condition the subjects dedicated a significant amount of time to looking at the speakers eyes. Additional confirmation comes from a direct investigation of the role of spatial frequencies that will be presented later in section 4.5.

The above yields a first argument in favour of the multiresolution approach on the image level: it might not be necessary to regard the whole spectral range of the signal for the automatic tracking. Because of the technical difficulties of image motion estimation any reduction of the signal that could help to exclude unwanted information and dampen disturbing factors must be greatly appreciated.

A second argument stems from the comparison with the human visual perception system (or what we think we know about it):

> The spatial frequency theory of image-based vision proposes that early visual processing can be understood in terms of a large number of overlapping psychophysical channels at different spatial frequencies and orientations. (Palmer, 1999)

In particular Marr (1982) strongly advocated the point that early visual processes would operate on several different scales. In his influential book he suggested the second derivative of a Gaussian as the basic operator. The circularly symmetric operator $\nabla^2 G$, where G stands for the two-dimensional Gaussian distribution

$$G(x, y) = e^{-\frac{x^2+y^2}{2\pi\sigma^2}} \tag{3.29}$$

and ∇^2 is the Laplacian operator $(\partial^2/\partial x^2 + \partial^2/\partial y^2)$. resembles from its shape a Mexican hat and constitutes a band-pass filter. Since human beings possess excellent abilities in face motion estimation, it appears promising to probe the aptness of concepts of human visual processing for technical solutions addressing the same problems. However, one should keep in mind that the human visual system in general is not yet fully understood and many details in particular are still completely unclear. One of these questions concerns the bandwidth of the proposed filters. While Marr (1982) suggests a general 1 1/2 octave wide filter, Gold, Bennett, and Sekuler (1999) found in several studies investigating face recognition that faces are processed with a 2 octaves wide filter. However, their use of a rectangular filter in the experiments makes the finding in our view a little bit questionable. Näsänen (1999) concluded from his experiments using Gaussian band-pass filters that the recognition of faces is relatively narrowly tuned, i.e., a bandwidth below 2 Octaves. Note that dyadic wavelets correspond to 1 octave band-pass filters.

The third argument for a multiresolution approach on the image level is purely technical. Firstly, very high spatial frequencies should be excluded because in this way pixel disturbances that are ubiquitous in digital images can be annihilated. Secondly, it was assumed that lighting changes are prevalent in the very low spatial frequencies. This does not hold in many circumstances, for instance shadows with sharp edges, as they might appear if the lighting of the face is not diffuse, certainly have an impact over the whole range of the spectrum. Another violation of the assumption is caused by specular highlights. Since their shape and size depend on surface orientation and curvature relative to the viewer and

relative to the light source, they might not extend too much towards the low spatial frequencies but they are not confined to the domain of the very high spatial frequencies. However, in in-door environments with normal ceiling lighting the assumption may be appropriate.

Thirdly, it was found in pretests that the texture map of small areas of the human face is locally unique, where locally is meant both in the spatial and in the frequency domain. Thus the correspondence problem is weakened or under ideal circumstances solved altogether. This is hardly surprising taking into account what has been said above about the human visual system and the social importance of face motion in human society. Figure 3.14 exemplifies a contrastive case where the two-dimensional cross-correlation (see section 3.3.6) used for the motion tracking fails to return the correct new location of one of the 'search segments', if applied to the unfiltered image. The figure shows two frames which differ more or less only in the degree of mouth opening of the subject. Figure 3.14(a) could be the first frame in typical motion tracking sequence, Figure 3.14(b) the consecutive frame. In reality more time passed between the two frames than just a frame-interval, but the difference in the mouth posture is still realistic (and actually often occurring) in a frame-to-frame situation, albeit fast jaw/lip movements like this would result in additional motion blurring. Since this would aggravate the judgement of the correct tracking in the example, we used these more static frames.

In the left upper corner of 3.14(a) the original source frame is shown. To the right of it the vertical subband on the 4th wavelet level, below the horizontal subband and in the lower right corner the diagonal subband. Two square areas of 48 pixels side length are marked with white and black rectangles in all images in the same way. One is covering the subject's opened mouth, the other one is located at the cheeks. Their centres are indicated by small crosses in the same colour. These are the 'search segments' which should be located in the target frame displayed with its wavelet subbands in Figure 3.14(b), and the rectangle centres correspond to mesh nodes whose potential movements should be determined. The white and black rectangles in the target frame mark the position of the maximum value of a cross-correlation of the enclosed area in the source frame with the whole of the target frame applied separately to the original intensity values and to each wavelet subband.

As can be seen the cheek area is tracked well in the original image as well as in all subbands most likely due to the strong image gradient where the visible surface of the face ends with a sharp edge meeting the head rest or the background room wall. However, the mouth area is completely mistracked using the original intensity values where the most similar area was found centred on the right eyebrow of the subject. The similarity between the curved eyebrow and the dark arc formed by the area between upper lip and teeth on the one side and the tongue on the other side might be the crucial misleading factor. In the wavelet subbands only the vertical subband positions the area wrongly too high and a little bit too much to the left at the nose, the other ones are correct. This is not surprising since the disappearing dark area within the opened mouth is mainly horizontally oriented.

The white and black circles give the centre of the square if the maximum correlation value is determined after summing up the correlation values of *all three* subbands, the procedure we use in the tracking (see below). As can be seen this way the tracking can cope well with the mouth closing since the area of the search segment is big enough to retain enough surrounding area (e.g., lips) and

the change due to the movement is mainly affecting only one orientation, i.e., filtered out in the other orientations, on this wavelet level.

Note that the cross-correlation search through the whole frame is not the usual procedure and not applied in the tracking procedure. It would be too time consuming and result in errors which could be avoided easily by having recourse to constraints on the facial surface, i.e. the places where parts of the face can move due to normal face motion is very limited. But for display purposes we needed here misplacements that are large enough to be spotted with the naked eye in static images.

The wavelet transformation can only be applied to a gray-scale image or to each channel of a multi-channel image, e.g, the red, green and blue channel of an RGB image, separately. Without doubt using the colour information would be highly desirable for the motion tracking procedure. Currently, however, this is not feasible, since it would slow down the already time consuming tracking too much: the whole procedure that is described in the following would have to be applied to each colour channel separately. Therefore incoming video frames are converted to grayscale images using standard perceptual weighting given by

$$I_{gray} = 0.299\, I_r + 0.587\, I_g + 0.114\, I_b \tag{3.30}$$

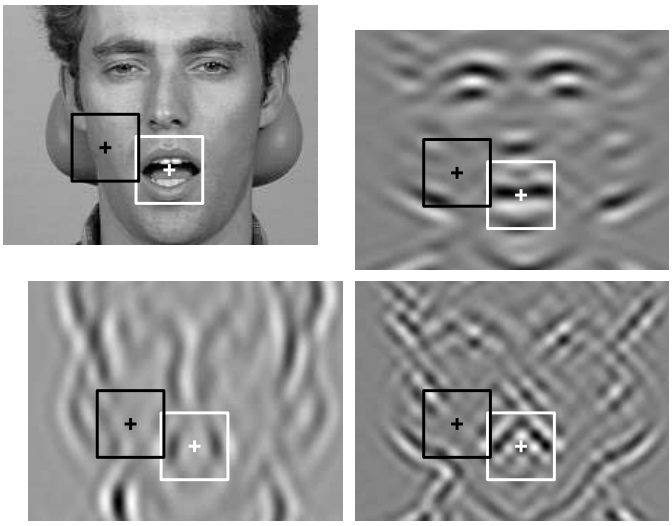
where I_r, I_g, and I_b are the intensity values of the red, green, and blue channels respectively. Alternative ways of weighting like for instance using only the saturation values of the *HSI representation* (**H**ue, **S**aturation, **I**ntensity) are still under investigation.

As described in section 2.3.2 a whole range of different wavelets are at hand for use in applications like ours. The question of what wavelet might be the best suited for a specific application constantly pops up in wavelet related discussion forums. And very often there is no clear answer. For our purpose the linear-phase property of spline-wavelets turned the balance towards them, even though the wavelet corresponding to a set of orthonormal, maximally flat FIR filters described in Vaidyanathan (1993, pages 532-536) proved in pretest to have slightly more favourable filtering characteristics. However the alignment of the different levels was not satisfying even if based on the energy or the mass centre of the filter. The result of the misalignment is that nodes of the ellipsoid mesh do not come to lie on exactly the same location at different tracking/wavelet levels (see below). Most of the time the cross-correlation based tracking can easily correct for the initial displacement caused by initialising the tracking on a higher level with the tracking results of a lower wavelet level (see below), but at points where the tracking runs into near critical problems these are substantially worsened.

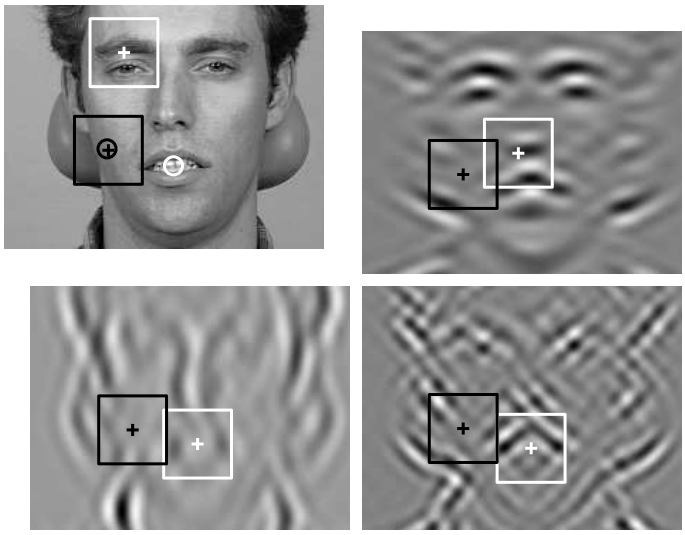
On the implementation side the Uvi_Wave wavelet toolbox [6] (see Sánchez et al., 1996, for details about the implemented algorithms) was used. The toolbox implements the spline wavelet transformation in the usual way as a cascade of biorthogonal filters.

Usually a multiresolution pyramid implies subsampling of the higher levels. Since the high spatial frequency portion is removed no information is lost by adequate subsampling, i.e., in the case of dyadic wavelets by just retaining every other sample point. However, since in our method the higher wavelet levels (lower spatial frequencies) in combination with reduced mesh node densities are used for prediction of initial mesh node locations on lower levels (see below), subsampling

[6] Download is available at `http://www.gts.tsc.uvigo.es/~wavelets/`



(a) Source frame



(b) Target frame

Figure 3.14: Comparison of tracking results using cross-correlation on original image intensity values and wavelet subband coefficients (see explanation in the text on page 68 for details).

would trivially reduce the accuracy of the tracking and thus prediction by the order of the subsampling factor. The lower levels are not in all cases able to catch this error. Interestingly a similar observation in connection with optical flow estimation and Gaussian pyramids was one of the motivation for the non-subsampled wavelet based optical flow determination in Wu et al. (1998), used

among other things for face tracking. Accordingly we do not subsample and use phase differences in the lower spatial frequency domain to track with an accuracy below the wavelength corresponding to these frequencies.

The question which wavelet levels should be used is still open. From the above said it is clear that the lowest wavelet level (ranging from half of the Nyquist frequency to the Nyquist frequency) should be disregarded, and if it only was to filter out pixel disturbances. Almost all image motion estimation methods require at least some smoothing of the raw input image. The inclusion or exclusion of higher level is then more governed by the requirements of the motion tracking described in the following than image processing considerations. For the upper bound run-time considerations and a minimal number of mesh nodes to preserve the ellipsoid topology are decisive, for the lower bound the necessary minimum number of pixels enclosed between the mesh nodes dictates the cut-off: enough pixels[7] must remain in the search segment to make the cross-correlation reliable (see below). As consequence levels 3-5 were chosen (see left column of Figure 3.15, the three subbands are superimposed for display purposes).

The choice was later validated in a perception experiment by Munhall et al. (Munhall, Kroos, and Vatikiotis-Bateson, 2001b,a, in press, see section 4.5).

3.3.2 Coarse-to-fine strategy with different mesh resolutions

With the next step in the tracking procedure a loop through the different wavelet levels is entered. This entails looping through different mesh resolutions and accordingly different tracking resolutions as well. In this way a coarse-to-fine strategy is implemented that is very much the heart of the tracking algorithm.

Coarse-to-fine strategies have a long history in image processing and are now ubiquitous, e.g, the use of Gaussian or Laplacian pyramids (Haberäcker, 1995). However, in our case not only the image processing operates on several resolution levels but the whole tracking process is realized in a multiresolution approach. It is shown with an undeformed mesh in Figure 3.15.

The processing starts on the highest selected wavelet level, i.e., level 5, (low spatial frequencies) with a maximally reduced node density of the ellipsoid mesh. Since the area for which correspondence is determined from one frame to the next (the search segment) is dependent on the mesh node spacing (see below), the tracking starts with determination of the location changes of relatively large parts of the face. Of course this only makes sense because of the constraints of the facial surface and face motion assumed in section 1.3.2. We will refer to this level as the coarse tracking level hereafter.

The results from the coarse level will be used to modify the initial locations of the next finer level that we will call the middle tracking level in the following. This kind of prediction when moving from a coarser to a finer level is one of the most important advantages of the coarse-to-fine strategy. For example, if the whole chin moves in conjunction with a large jaw opening gesture, the motion will be registered already at the highest level and the starting position for the search of – e.g., a part of the lower lip – will be shifted accordingly. Otherwise the tracking procedure on the middle level might attempt to find the corresponding texture map values of the lip in what has become the area within the opened mouth in

[7] Here and in the following we will often speak of pixels despite the fact that the tracking procedure after the wavelet filtering operates only on the wavelet subband coefficients. It just simplifies description and imagination enormously, if we treat the subband coefficients as normal images for the time being. Wherever it could lead to erroneous conclusion, we will abstain from it.

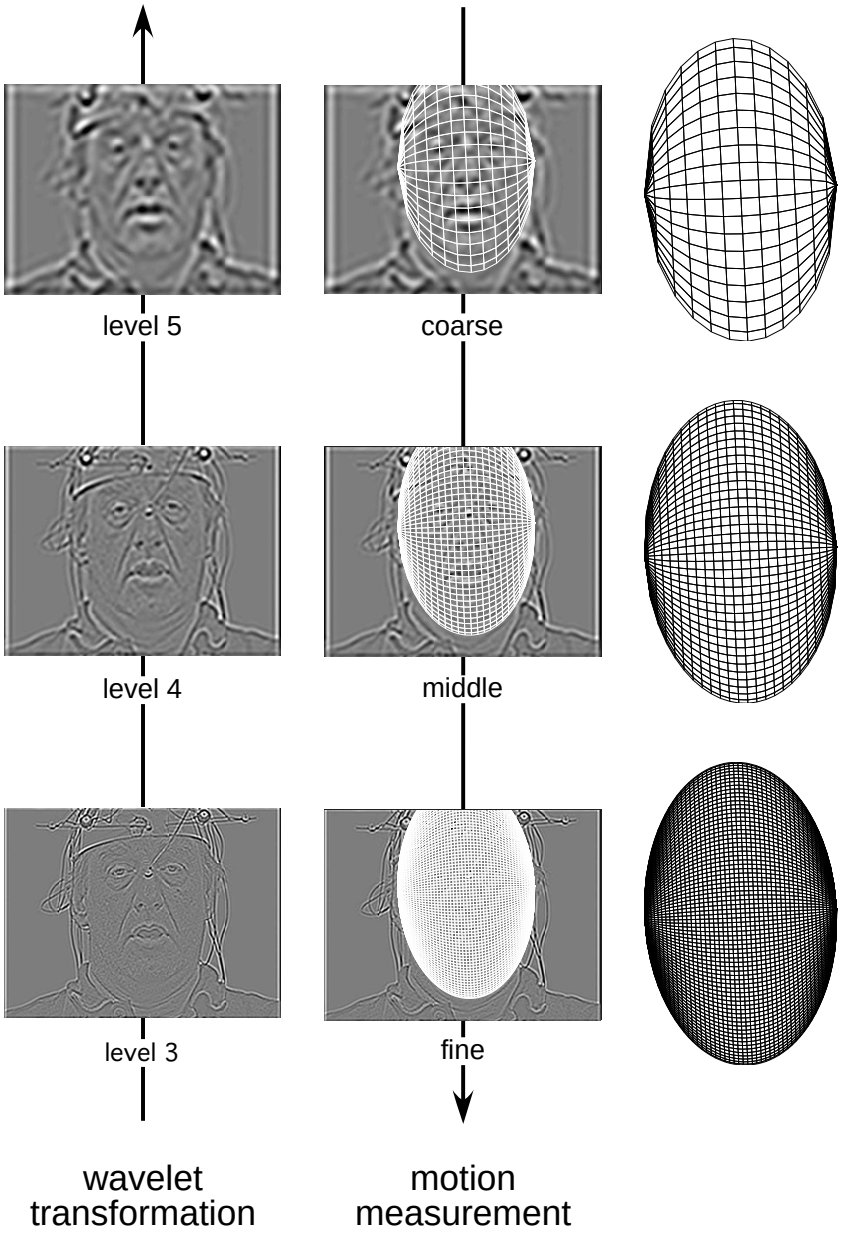


Figure 3.15: The multiresolution approach of the tracking: Different levels of the wavelet transformation (subbands superimposed for illustration purposes, left side) and corresponding mesh models with different node densities (right side). For the motion tracking the mesh is projected onto the 'subband image' (centre).

the incoming video frame. Since the search has to be limited to the immediately surrounding area the procedure would quite certainly fail to return the correct new location.

At the middle level the mesh node density is doubled along both of the sur-

face dimensions of the half ellipsoid. This has its correspondence in moving to the next lower wavelet level (level 4, relatively higher spatial frequencies), since the band of remaining frequencies is now centred on a twice as high centre frequency. From another viewpoint: the length of the sides of the search segment are halved along each axis (the enclosed area comprises now only a fourth or so of the original area) as are by virtue of the wavelet filtering the wavelengths of the frequencies remaining in the image along both image dimensions. Then the tracking is repeated, now with a four times higher number of nodes, yielding on the one hand a refinement of the already obtained tracking results and on the other hand an increase of resolution.

What has been described for the middle level of tracking is then iterated in passing to the last level (wavelet level 3, 'fine tracking level'). The final mesh used here has the full intended resolution. We will dicuss problems with the reliability of the results that specifically come in on this level in the validation chapter (chapter 4).

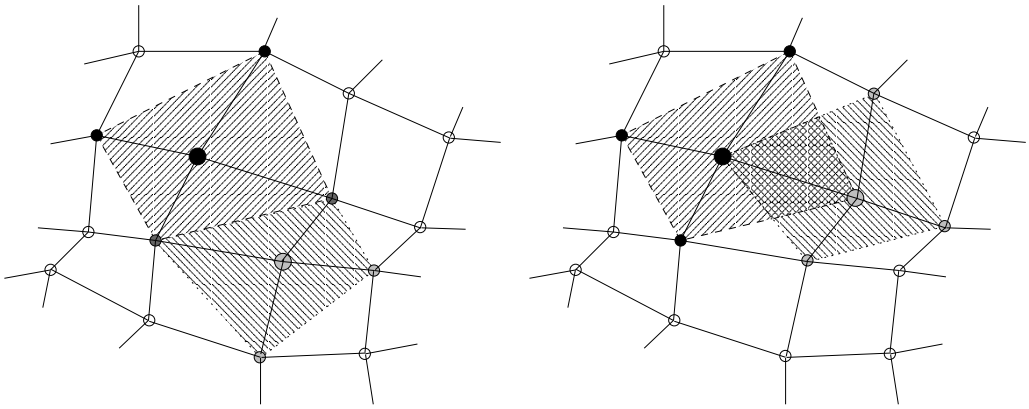
3.3.3 Mesh projection

Using the camera model derived in section 3.2.5 ellipsoid meshes of any resolution can be projected onto the face in the image and kept there during the whole image sequence by steering its three-dimensional pose prior to projection with the head tracking data acquired in section 3.2.1. As said before the different wavelet levels do not lead to any difficulties if spline wavelets are used because of their linear phase. Compensation of the filter delay is straightforward and boundary effects occurring at the image borders are usually far enough away from the face surface area to not impede the tracking process. It should be clear that the mesh never needs to be actually superimposed onto the 'subband image' (as shown in Figure 3.15, middle column) except for human visual inspection. For the tracking we only need the image coordinates of the mesh nodes to determine search segments as explained in the next section.

3.3.4 Determining search segments on the texture map

The motion tracking procedure is frame-to-frame based. Therefore we have to determine the change of location of the face surface from one frame to the next. This and the following step establish the conditions for the comparison of two consecutive video frames. Unfortunately the continuous character of the facial surface implies that there are no rigid objects to track. Even in the case of the jaw that clearly is a rigid object and whose motion can be described with the usual six rigid motion parameters the appearance of its movements in the video frame is that of non-rigid motion due to the layers of muscles and skin above the bone structure. The eyeballs are an exception to this, but eye blinking makes them a less than favourable candidate and they are not very important for speech anyway. The bridge of the nose might be another exception, but even here some people show wrinkles when wrinkling their nose.

Since we were not interested in tracking features, the only remaining solution is to partition the face surface in small parts. The general way of accomplishing it using a multiresolution approach was already discussed in former sections. But what about the practical details? It was suggested by many studies and authors concerned with animating faces in computer graphics (see Parke and Waters, 1996) that different mesh resolutions (different patch sizes) are necessary for



(a) Diagonal neighbours (b) Horizontal neighbours

Figure 3.16: Section of the ellipsoid mesh with two search segments marked. The search segment is defined by the four neighbouring nodes surrounding the centre node.

the animation of different parts of the face, i.e., high resolutions in the mouth and eye region and lower resolution in the cheek area. The same is sometimes claimed though not proved for face motion tracking. One of the longterm goals of our work is to investigate whether or not this claim holds in general and if not whether there are circumstances where it nevertheless is found to be true. Are there maybe differences depending on the kind of face motion, and how is the situation in particular for speech face motion? This requires being able to look at the covariations or correlations between measurements globally distributed over the face. Therefore we chose for the tracking 'atom' on the final fine level of our tracking a relatively small area that is distributed globally over the face and has more or less the same size everywhere. The shape and size of the search areas on the coarser levels are merely a consequence of this.

Of course the 'search segment', as we will call the search area from now on consistently, has to be well-defined everywhere. This is achieved in our algorithm by defining it as the area enclosed by the quadrilateral created by taking the four neighbouring nodes surrounding a centre node as its vertices. Figure 3.16 shows an example.

The search segments of diagonally neighbouring nodes share a border but do not overlap (Fig. 3.16(a)), vertically or horizontally neighbouring nodes share about one fourth of their area (Fig. 3.16(b)). Taking a closer look at all surrounding segments in vertical and horizontal direction reveals that each pixel of the texture map is used twice in the tracking. This redundancy, however, is intentional and its important role in the tracking will become evident later on.

To determine whether a pixel really lies within the quadrilateral the MATLAB function `inpolygon` is used.

Some segments, however, must be excluded: If head motion causes one or more segments to be 'occluded' by the reminder of the ellipsoid their location cannot be determined anymore. If the mesh were a solid 'real world' object the occluded part would be just not visible in the video frame. But the two-dimensional

projection of the virtual mesh nodes still returns image coordinates for these nodes - they are just wrapped around the curve of intersection between the ellipsoid and an arbitrary plane parallel to the image plane. It goes without saying that this would not only result in wrong values for the occluded segments but would interfere with the whole motion tracking. Fortunately those segments can be easily recognised by probing the angle between the optical axis and the vertex normals of the mesh nodes that define the search segment. If the absolute value of the angle is greater than 90 degree for any limiting mesh node, the search segment must be excluded from the tracking for the time being (i.e. for this particular frame-to-frame transition). Since the limiting mesh nodes form the vertices of the search quadrilateral, and the vertices are the quadrilateral extremal points, and further, the intersection curve of the half ellipsoid with any plane parallel to the image plane is convex,[8] no pixel that would not be visible if the ellipsoid was the 'real world' facial surface is included in the tracking.

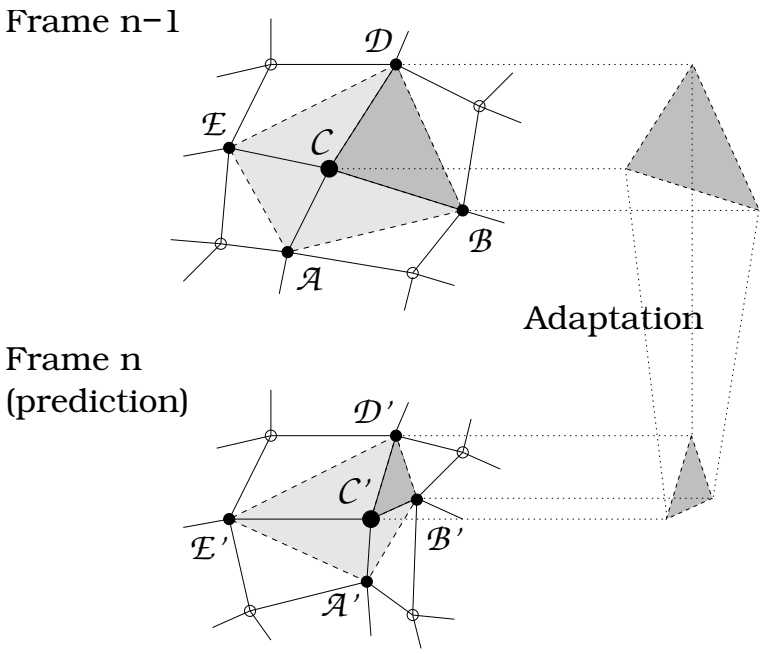


Figure 3.17: Adaptation of the search segment (light gray area) of two successive frames, shown for one 'quadrant' (dark gray area).

3.3.5 Warping the search segments

After having determined a tracking level specific search segment in the first of two consecutive frames we could now try to find the corresponding location of it in the next frame. However this would mean disregarding some of the information gained by the head tracking. Keeping the ellipsoid mesh at a constant

[8] Unless the half ellipsoid is rotated itself more then 90 degrees around the longitudinal and/or transversal axis relative to its 'neutral' starting position (i.e. coronal plane parallel to image plane). In this case, however, the tracking would be suspended altogether because of the lack of facial surface visible in the video frame.

position relative to the subject's face guarantees the best starting location for the correspondence search in the incoming frame. But we would not have considered shape distortions of the search segment due to the projection of the mesh with a potentially new pose. Therefore the search segment has to be adapted by warping the whole area it comprises to fit the shape set by the new mesh node coordinates in the second frame (see Figure 3.17). This is equivalent to extracting a texture map in the first frame and applying it to the mesh with the pose derived from the second frame, i.e., rendering the texture-mapped ellipsoid with the new pose (image registration, see section 2.1.1).

On the middle and the fine level of tracking another exigency for warping of the search segment arises. At those levels motion tracking results from the level one step higher are already available. To roughly a quarter of the mesh nodes new coordinates could be assigned. The remaining ones could be updated using interpolation (see section 3.3.7 below). Thus the likely position and shape of the search segment in the next frame can be predicted based on the information of the already completed tracking on the higher level. However, the tracking is designed to return only results for the mesh nodes as will become clear later on. To take advantage of the already obtained information in solving the correspondence problem the whole area comprised in the search segment must be adapted, i.e., warped. At this point the full strength of the multiresolution approach comes into its own. By warping the search segment according to the predicted shape, the expected texture map of the search segment is also predicted. Taking the example of a large jaw movement, the texture map of a smaller segment located right at the corner of the mouth will undergo dramatic changes. Using the results of the motion tracking on the higher level and a warping procedure these changes will be approximately determined before we start to look for corresponding areas in the incoming frame. Again this can be thought of as a rendering of the texture mapped surface patch that constitutes the search segment based on the new position and shape parameters.

How is the warping accomplished? Clearly it is a *geometrical transformation*, since not only the intensity values of the pixel within the search segment are modified but their spatial relationships as well. As Gonzalez and Woods (2002) points out:

> Geometric transformations often are called *rubber-sheet transformations*, because they may viewed as the process of "printing" an image on a sheet of rubber and then stretching this sheet according to some predefined set of rules. (Gonzalez and Woods, 2002, page 270)

The comparison validates in a way to model face motion with a geometric transformation: Although the facial surface does not really possess the properties of rubber the skin parts have some similarities. The most important exception, the area within the opened mouth, will hence need to be watched closely in the tracking.

Gonzalez and Woods (2002) continue

> In terms of digital image processing, a geometric transformation consists of two basic operations: (1) a *spatial transformation*, which defines the "rearrangement" of pixels in the image plane; and (2) *gray-level interpolation*, which deals with the assignment of gray levels to pixels in the spatially transformed image. (Gonzalez and Woods, 2002, page 271)

The statement given in the context of image restoration can be directly transferred to our search segment warping: In the restoration of a geometrically distorted image normally so-called *tiepoints* are used and these tiepoints are pixels whose location are known in both states, distorted and undistorted. The mesh nodes that define the search segment correspond exactly to the tiepoints.

Let $I(x, y)$ be the intensity function which assigns intensity values to the coordinates x and y in the unwarped search segment and $I'(x', y')$ the equivalent for the warped search segment. Then the spatial transformation could be expressed as

$$\begin{aligned} x' &= g(x, y) \\ y' &= h(x, y) \end{aligned} \tag{3.31}$$

$g(x, y)$ and $h(x, y)$ must be determined based on the location differences of the mesh nodes. There are several ways to accomplish that, since we have 5 mesh nodes that control the search segment.

3.3.5.1 Piecewise affine transformation

In the earlier version of the algorithm we used an *affine transformation* for each 'quadrant' of the search segment. The quadrant is defined here as the triangle between the centre node and two adjacent limiting nodes (see Figure 3.17). This results in a *piecewise affine transformation* for the search segment as a whole. The affine transformation[9] is given by

$$\begin{aligned} x' &= ax + by + p \\ y' &= dx + ey + q \end{aligned} \tag{3.32}$$

It includes rotation, translation, scaling and skewing. Since we have six parameters we need at least three points to compute them using the resulting six equations in the form of (3.32). In our case these are the three mesh nodes.

It can be shown that every affine transformation is composed of a linear transformation and a translation (see Theorem 1 on page 45 in Gomes et al., 1999). Therefore one mesh node, e.g, the centre node, can be utilised to determine the translation, in other words we can assume without loss of generality (Draper and Beveridge, 2002)

$$x_1 = 0 \quad \text{and} \quad y_1 = 0$$

where x_1 and y_1 are the coordinates of one of the mesh nodes. This yields $p = x'_1$ and $q = y'_1$. Then the remaining equations for each spatial direction are solved for the unknown parameter

$$x'_2 = ax_2 + by_2 + x'_1 \tag{3.33}$$

[9] Mathematically more rigorously the affine transformation is defined as the transformation $f : \mathbb{R}^n \to \mathbb{R}^n$ with

$$f((1-t)P + tQ) = (1-t)f(P) + tf(Q)$$

where $P, Q \in \mathbb{R}^n$, and $t \in \mathbb{R}$ (see Gomes, Darsa, Costa, and Velho, 1999, page 45).

$$x'_3 = ax_3 + by_3 + x'_1 \tag{3.34}$$

Solving (3.33) for a gives

$$a = \frac{x'_2 - x'_1 - by_2}{x_2} \tag{3.35}$$

Then substituting a in 3.34 with 3.35 yields

$$x'_3 = \left(\frac{x'_2 - x'_1 - by_2}{x_2}\right) x_3 + by_3 + x'_1 \tag{3.36}$$

By isolating b with

$$\begin{aligned} x'_3 - x'_1 - &= \frac{x'_2x_3 - x'_1x_3 - byx_3 + by_3x_2}{x_2} \\ x_2(x'_3 - x'_1) - x'_2x_3 + x'_1x_3 &= -byx_3 + by_3x_2 \\ x_2(x'_3 - x'_1) - x_3(x'_2 - x'_1) &= b(-yx_3 + by_3x_2) \end{aligned} \tag{3.37}$$

and rearranging the terms we obtain the final solution for b

$$b = \frac{(x'_3 - x'_1)x_2 - (x'_2 - x'_1)x_3}{-x_3y_2 + y_3x_2} \tag{3.38}$$

Trivially by substituting the right side of 3.38 in 3.35 the solution for a is obtained.

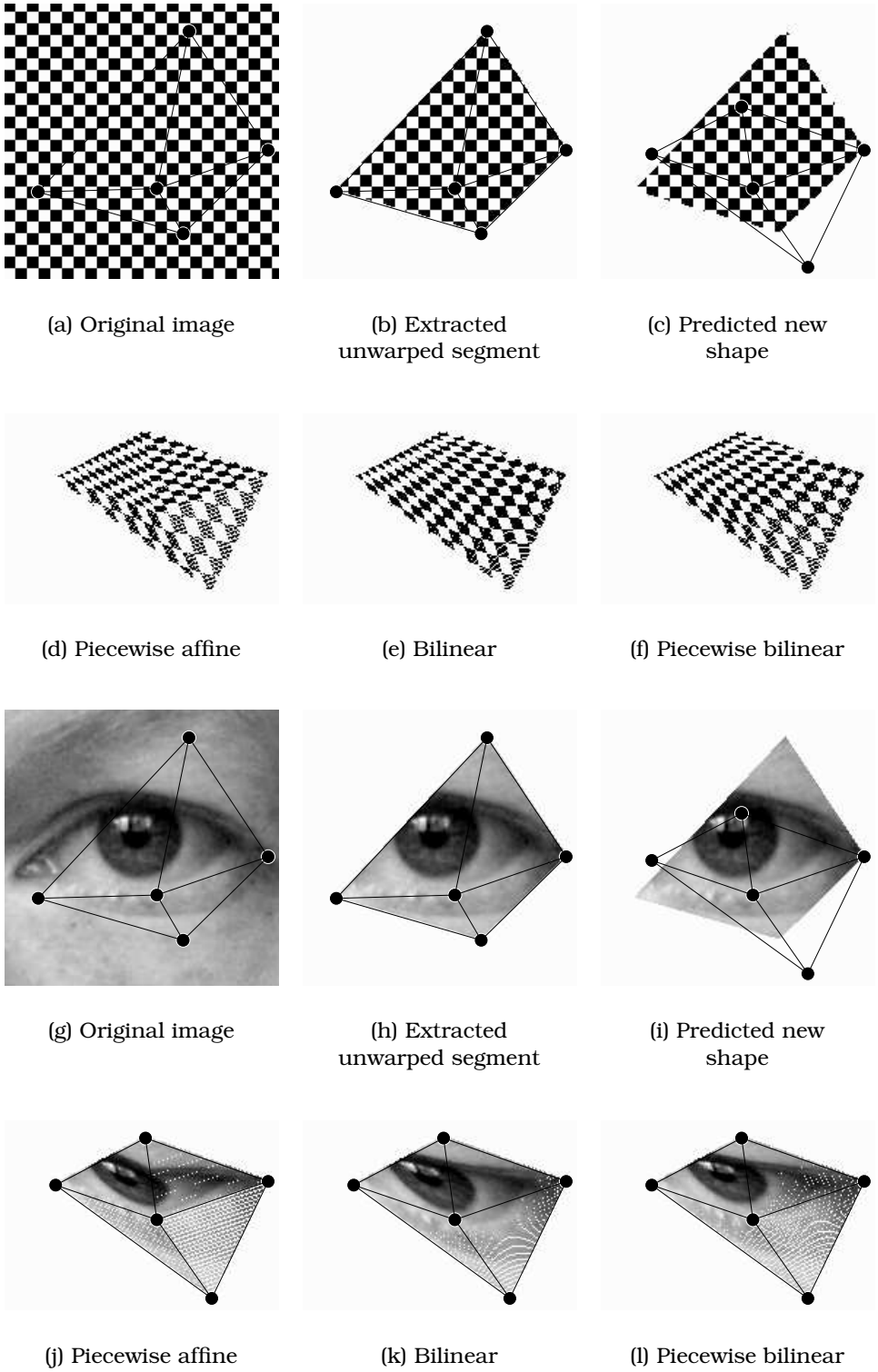
The whole process is equivalent to the following procedure that we implemented in the first version of the algorithm:

i. The mesh nodes of the unwarped and the warped search segment were translated so that each centre node became the origin.

ii. For each quadrant of the unwarped segment the coordinate values of all included pixels in an oblique non-Cartesian coordinate system were calculated, where the coordinate system was defined by considering the two mesh nodes that limit the quadrant as unit vectors of this system.

iii. The resulting coordinate values were treated as if they were coordinates in another oblique coordinate system defined by taking the two mesh nodes which limit the equivalent quadrant in the warped search segment as unit vectors, i.e., to get the new image coordinates of the pixels the oblique coordinates obtained from the *unwarped* segment were transformed back into the Cartesian coordinate system of the image by using the specification of the oblique coordinate system of the *warped* segment.

iv. Then the entire warped segment was translated back into its starting location.



(a) Original image (b) Extracted unwarped segment (c) Predicted new shape

(d) Piecewise affine (e) Bilinear (f) Piecewise bilinear

(g) Original image (h) Extracted unwarped segment (i) Predicted new shape

(j) Piecewise affine (k) Bilinear (l) Piecewise bilinear

Figure 3.18: Comparison of the warping methods

The affine transformation returns most of the time non-integer values for the new pixel locations, but for any further processing we need of course the intensity value at the regular integer positions. Thus they must be interpolated in a step called *intensity interpolation*. We used a so-called *zero-order interpolation*, the simplest form of *nearest neighbour interpolation*, a choice more influenced by considerations to keep the already high computational burden within manageable limits than the quality of the interpolation. Thereby simply every non-integer coordinate within the range of the search segment is rounded to the nearest integer. A promising alternative would be the bilinear interpolation of the four nearest neighbours, but currently it would slow down the tracking procedure too much.

Figure 3.18(d) and 3.18(j) on the page before show the affine transformation of a search segment with a checkerboard-like pattern and a real texture map extracted from the image of a face. The main disadvantage of the affine transformation based on the triangular subparts of the search segment can be observed clearly in the figure. The transformation per se is continuous, but there are discontinuities in the first derivative at the quadrant borders (see Gomes et al., 1999, page 48).

3.3.5.2 Bilinear transformation of the entire search segment

A quite obvious alterative would be to use a *bilinear transformation* of the quadrilateral search segment as a whole. Bilinear transformation is given by

$$\begin{aligned} x' &= ax + by + cxy + p \\ y' &= dx + ey + fxy + q \end{aligned} \tag{3.39}$$

Since there are now eight free parameters we need four tiepoints to obtain the necessary eight linear equations. Using matrix notation and the four neighbouring nodes as tiepoints $[x_1, y_1]$, $[x_2, y_2]$, $[x_3, y_3]$, $[x_4, y_4]$ of the original segment, and $[x'_1, y'_1]$, $[x'_2, y'_2]$, $[x'_3, y'_3]$, $[x'_4, y'_4]$ of the warped segment) the equations can be written as

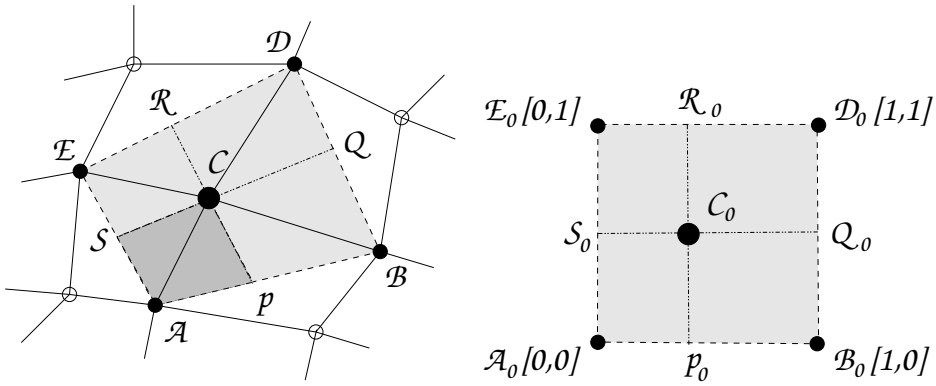
$$\begin{bmatrix} x'_1 & y'_1 \\ x'_2 & y'_2 \\ x'_3 & y'_3 \\ x'_4 & y'_4 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & x_1y_1 & 1 \\ x_2 & y_2 & x_2y_2 & 1 \\ x_3 & y_3 & x_3y_3 & 1 \\ x_4 & y_4 & x_4y_4 & 1 \end{bmatrix} \begin{bmatrix} a & d \\ b & e \\ c & f \\ p & q \end{bmatrix} \tag{3.40}$$

The solution is found analogously to the camera calibration solution of equation (3.19).

However this procedure can only be applied for warping from the unit square to an arbitrary quadrilateral not from one arbitrary quadrilateral to another (see Gomes et al., 1999). In order to achieve the latter first the coefficients for the bilinear transformation from the unit square to the unwarped search segment (backward warping) must be determined as described above. Then they are used to determine the coefficients for the *inverse bilinear transformation* (see below) which warps the search segment into the unit square. After that the coordinates obtained for the unit square can be bilinearly transformed to the new shape of the search segment (forward warping) - again in the way described above. This yields the new spatial locations for all pixels of the search segment. Afterwards the intensity values can be interpolated.

However that would mean sacrificing another key concept of the tracking algorithm: In order to solve the correspondence problem we use a well-defined image region, the search segment. In the comparison procedure the new location of the search segment is ascertained which yields two translation parameters in the image coordinate system. Our tracking model however is the image independent ellipsoid mesh its nodes representing the facial surface. Therefore the two translation values will be assigned to the *projected mesh node in the centre of the search segment*. In other words, the only purpose of the search segment is to have enough 'material' (pixels) to determine the change of location from one frame to the next of the mesh node in its centre.

Accordingly the mesh node must not 'float' within the segment as it would do with bilinear transformation of the whole segment, but should be treated as the fifth and actually most important reference point of the segment. Non-observance would lead to mistracking almost instantly in any case where the search segment is not lying within a homogeneously moving region of the face. In the piecewise affine transformation described above the principle of the fixed centre node is fully employed by virtue of the definition of the quadrants. Figure 3.18(e) and 3.18(k) on page 79 show the bilinear transformation of the entire search segment - again for a checkerboard-like pattern and the image of a human eye. Notice that the centre node in the original search segment - shown in Figure 3.18(h) - is located just on the edge of the lower palpebra, but after the bilinear transformation it can be found within the sclera - in contrary to the piecewise affine (Figure 3.18(d)) and the piecewise bilinear (see next section, Figure 3.18(f)) transformation.



(a) Division of the search segment into four regions for piece-wise bilineatransformation

(b) Division of the unit square into four regions corresponding to Figure 3.19(a)

Figure 3.19: Determination of four qudrilaterals within the search segment

3.3.5.3 Piecewise bilinear transformation

How can the favourable properties of both methods be combined? The answer is to resort to a *piecewise bilinear transformation*. Clearly it cannot be based on triangulars and it should not be based on the 'halfs' of the search segment, e.g.,

the quadrilaterals $\overline{\mathcal{ABDC}}$ or $\overline{\mathcal{ACDE}}$ in Figure 3.19(a), since the centre mesh node might be collinear with two of the other mesh nodes.

However, the quadrilaterals based on the additional tiepoints $\mathcal{P}$, $\mathcal{Q}$, $\mathcal{R}$, and $\mathcal{S}$, e.g., $\overline{\mathcal{APCS}}$ would be suited. $\mathcal{P}$, $\mathcal{Q}$, $\mathcal{R}$, and $\mathcal{S}$ are the projections of the centre node $\mathcal{C}$ onto the borders in the unit square with vertices $[0,0]$, $[1,0]$, $[1,1]$, $[0,1]$ assuming the search segment has been warped to the unit square.

This is shown in Figure 3.19(b) (Note that Figure 3.19(a) and 3.19(b) are schematic and might not represent the real relationships correctly). Here we mapped $\mathcal{A}$ to $[0,0]$.

Now assume we already found the coordinates of the additional tiepoints in the unit square giving us

$$\mathcal{P}_0 = [x_{\mathcal{C}_0}, 0] \qquad \mathcal{Q}_0 = [1, y_{\mathcal{C}_0}] \qquad \mathcal{R}_0 = [x_{\mathcal{C}_0}, 1] \qquad \mathcal{S}_0 = [0, y_{\mathcal{C}_0}]$$

In order to get their position in the original search segment, we would need to warp them back into it. Since we have the corner points of the unit square as tiepoints, (3.40) can be easily solved symbolically yielding in its general form

$$\begin{aligned} a &= x'_2 - x'_1 \\ b &= x'_4 - x'_1 \\ c &= x'_3 - x'_2 + x'_1 - x'_4 \\ p &= x'_1 \\ d &= y'_2 - '_1 \\ e &= y'_4 - y'_1 \\ f &= y'_3 - y'_2 + y'_1 - y'_4 \\ q &= y'_1 \end{aligned} \tag{3.41}$$

Thus the bilinear transformation for $\mathcal{P}_0$, $\mathcal{Q}_0$, $\mathcal{R}_0$, and $\mathcal{S}_0$ would be

$$\begin{bmatrix} x_{\mathcal{P}} & y_{\mathcal{P}} \\ x_{\mathcal{Q}} & y_{\mathcal{Q}} \\ x_{\mathcal{R}} & y_{\mathcal{R}} \\ x_{\mathcal{S}} & y_{\mathcal{S}} \end{bmatrix} = \begin{bmatrix} x_{\mathcal{C}_0} & 0 & 0 & 1 \\ 1 & y_{\mathcal{C}_0} & y_{\mathcal{C}_0} & 1 \\ x_{\mathcal{C}_0} & 1 & x_{\mathcal{C}_0} & 1 \\ 0 & y_{\mathcal{C}_0} & 0 & 1 \end{bmatrix} \begin{bmatrix} a & d \\ b & e \\ c & f \\ p & q \end{bmatrix} \tag{3.42}$$

But we do not know $x_{\mathcal{C}_0}$ and $y_{\mathcal{C}_0}$ yet, thus we are left with only 8 equations for 10 unknowns. The remaining two equations to make the system solvable are provided by the bilinear equations for the centre node $\mathcal{C}$ itself, since clearly

$$x_{\mathcal{C}} = a x_{\mathcal{C}_0} + b y_{\mathcal{C}_0} + c x_{\mathcal{C}_0} y_{\mathcal{C}_0} + p \tag{3.43}$$

and

$$y_{\mathcal{C}} = d x_{\mathcal{C}_0} + e y_{\mathcal{C}_0} + f x_{\mathcal{C}_0} y_{\mathcal{C}_0} + q \tag{3.44}$$

Solving (3.43) for x_{C_0} we obtain

$$x_{C_0} = \frac{x_C - by_{C_0} - p}{a + y_{C_0}} \tag{3.45}$$

Substituting (3.45) in (3.44) gives

$$y_C = d\frac{x_C - by_{C_0} - p}{a + y_{C_0}} + ey_{C_0} + fy_{C_0}\frac{x_C - by_{C_0} - p}{a + y_{C_0}} + q \tag{3.46}$$

which can be rewritten as

$$(a + y_{C_0})(y_C - ey_{C_0} - q) - (d + fy_{C_0})(x_C - by_{C_0} - p) = 0 \tag{3.47}$$

Expanding (3.47) and isolating y_C yields a standard quadratic equation

$$(fb - ce)y_{C_0}^2 + (cy_C - fx_C - ae - cq + db + fp)y_{C_0} + (ay_C - dx_C - aq + dp) = 0 \tag{3.48}$$

which can be solved by means of standard algebra for y_{C_0}. It can be shown that from the resulting two values only one lies within the unit square and hence must be the correct one. By substituting this value for y_{C_0} in (3.45) x_{C_0} is determined as well. Notice that the above solution for the centre node constitutes the inverse of the bilinear transformation (see Gomes et al., 1999, for more details on the inverse of the bilinear transformation and a geometric interpretation of it).

Using (3.42) the coordinates of $\mathcal{P}$, $\mathcal{Q}$, $\mathcal{R}$, and $\mathcal{S}$ in the original search segment can be computed. In the same way the coordinates of the corresponding points in the new search segment, i.e. our warping target, $\mathcal{P}'$, $\mathcal{Q}'$, $\mathcal{R}'$, and $\mathcal{S}'$ can be determined. Having obtained all necessary tiepoints we are now able to warp the pixels within the quadrilaterals $\overline{\mathcal{CSAP}}$, $\overline{\mathcal{CPBQ}}$, $\overline{\mathcal{CQDR}}$, and $\overline{\mathcal{CRES}}$ (Figure 3.19(a)) separately using bilinear transformation. The options for the final determination of the intensity values at the integer pixel coordinates are the same as described for the piecewise affine transformation.

Results for checkerboard pattern (Figure 3.18(f)) and the image of an human eye (Figure 3.18(l)) can be compared with the other warping methods in Figure 3.18 on page 79.

3.3.5.4 Aliasing and completeness

Independently of which one of the above warping methods is used there is in general the danger of aliasing effects introducing noise in the warped segment and thus diminishing the advantage gained by predicting the expected texture map of the search segment in the incoming frame. However, even on the lowest of the selected wavelet levels (level 3) we are already operating on a 4 times oversampled signal in each spatial direction. Thus the chance of aliasing effects caused by e.g., a compression of the segment, is rather minimal.

This leads to interesting considerations on the completeness of the warped segment. In the above example of a compressed segment (or part of it) the transformation and nearest neighbour interpolation might return several values for

the same integer pixel location. For our proposed method of determining correspondence (see next section) this is unproblematic, it would just result in an implicit averaging of these values.[10] In the case that the prediction process forecasts an substantially enlarged search segment, the warping would return new coordinates only for the much smaller number of pixels in the original search segment, i.e., the pixels in the warped segment would be to a certain degree scattered over the area of the warped segment. Of course, the remaining gaps in the warped segment could be interpolated, but for our proposed method of determining correspondence this is again not necessary and since it would increase the computational effort both in the warping and in the correspondence routine without adding any new information, we are very much inclined to abstain from it.

Here another fundamental strength of the algorithm is revealed: If the image was not bandlimited and oversampled, the prediction with the warped segment would lead at some points to low correlation values even at the correct new location of the search segment: since the prediction can never be perfect (the face has no underlying ellipsoidal mesh structure) for instance high frequency details would be likely to be missed by the scattered pixel of an enlarged warped search segment (interpolation of the missing pixel would not make a difference). One could assume that the results would be still better than without any prediction, but here speculation starts. Note, however, that the coarse-to-fine strategy with different mesh resolutions and their inherent limitation of the extent a mesh node can move (see next section) guarantees that the expansion of a segment is contained within a certain range.

3.3.6 Determining correspondence

After having established the conditions for the comparison of to consecutive frames thereby maximising the use of already available information, we are now ready two tackle the correspondence problem itself. Since it is not only a fundamental problem of image motion estimation, but also of *stereopsis*, some well established correlation methods are at hand (Trucco and Verri, 1998).

If we denote by I_1 and I_2 two arbitrary frames, then the similarity criterion r for the correlation window of width (and height) $2k+1$ pixel centred at image coordinates $[x, y]$ in I_1 with its equivalent in I_2 displaced by $[\Delta x, \Delta y]$ is computed according to the following methods:

***i.* Cross-correlation**

$$r_{(\Delta x,\Delta y)}(x,y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} I_1(x+i, y+j)\, I_2(x+i+\Delta x,\, y+j+\Delta y) \tag{3.49}$$

***ii.* Sum of squared distances (SSD)**

$$r_{(\Delta x,\Delta y)}(x,y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} -\big(I_1(x+i, y+j) - I_2(x+i+\Delta x,\, y+j+\Delta y)\big)^2 \tag{3.50}$$

[10] In some cases (i.e., high number of multiple values) it might be recommended to do the averaging explictly in order to cut computational costs.

iii. **Normalised cross-correlation**

$$r_{(\Delta x,\Delta y)}(x,y) = \frac{\sum_{i=-k}^{k}\sum_{j=-k}^{k}\left(I_1(x+i,y+j)-m_1\right)\left(I_2(x+i+\Delta x,\, y+j+\Delta y)-m_2\right)}{\sqrt{\sum_{i=-k}^{k}\sum_{j=-k}^{k}\left(I_1(x+i,y+j)-m_1\right)^2 \sum_{i=-k}^{k}\sum_{j=-k}^{k}\left(I_2(x+i+\Delta x,y+j+\Delta y)-m_2\right)^2}} \tag{3.51}$$

where m_1 and m_2 are the means of the intensity values of the correlation window in the respective frames.

The normalised cross-correlation of (3.51) takes into account the first-order statistics of the involved regions by subtracting the mean of the correlation window and normalising by the product of the standard deviations (the normalisation factor $n-1$ of the standard deviations themselves can be cancelled, its square appears in the numerator as well).

These are the general definitions. They imply a square-shaped window that is centred on a single pixel in one of the images and shifted across a certain area in the other image to calculate some value of correspondence. However, the square shape is not essential for the corresponding problem.

Two important parameters must be set: The size of the correlation window, which naturally has an effect on the lower bound of spatial frequencies contributing to the solution, and the displacement vectors, which limit the maximum distance for which correspondences can be found (if not the whole image is searched through, which is almost always not sensible because of the computational effort involved). The setting of both parameters requires some knowledge or expectations about the input images and the nature of the structures or patterns for which correspondence should be established.

We will now present a slightly modified correspondence procedure to fit the particular needs of our approach. It implicitly solves the problem of determining the window size and the displacement vectors. We will first formally develop the algorithm and then explain its characteristics.

Let $\mathbf{I}_p$ be the first of the two consecutive frames (the previous one), $\mathbf{I}_a$ the incoming second frame (the actual one), and $\mathbf{S}_{p_w}$ the warped search segment stemming from the previous frame with the centre node translated to the origin using the translation vector $\mathbf{t} = [t_x, t_y]$. Note that $\mathbf{S}_{p_w}$ contains intensity values derived from $\mathbf{I}_p$, but due to the interpolation process included in the warping actually no original value might have been preserved. Note also that the translation vector refers already to the predicted position of $\mathbf{S}_{p_w}$ in the incoming frame. Let $\mathbf{p}_{p_w} = [x_w, y_w]$ be coordinate vectors of the pixels within the search segment (thus the centre node would be [0,0]), and $\mathbf{C}_{p_w}$ the set of all those vectors. Furthermore, let n be the number of pixels in $\mathbf{S}_{p_w}$ and let $\mathbf{D}$ be the set of two-dimensional displacement vectors $\mathbf{d} = [d_x, d_y]$ that shift the cross-correlation window over the designated area in the incoming frame.

i. Set

$$\mathbf{D} = \mathbf{C}_{p_w} \tag{3.52}$$

ii. For every $\mathbf{d}$ compute

$$r(\mathbf{d}) = \sum_{i=1}^{n} f\left(\mathbf{S}_{p_w}(\mathbf{C}_{p_w}(i)),\ \mathbf{I}_a(\mathbf{C}_{p_w}(i) - \mathbf{t} + \mathbf{d})\right) \tag{3.53}$$

where f is one of the functions

$$f(u,v) = uv \tag{3.54a}$$

$$f(u,v) = -(u-v)^2 \tag{3.54b}$$

$$f(u,v) = \frac{(u - m_{p_w})(v - m_a)}{s_{p_w} s_a} \tag{3.54c}$$

with m_{p_w} being the intensity mean of the n pixels in the warped search segment and m_a the intensity mean of its shifted equivalent in the actual frame

$$m_{p_w} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{S}_{p_w}(\mathbf{C}_{p_w}(i)) \qquad m_a = \frac{1}{n}\sum_{i=1}^{n} \mathbf{I}_a(\mathbf{C}_{p_w}(i) - \mathbf{t} + \mathbf{d})$$

and s_{p_w} and s_a the unnormalised standard deviations of the pixels belonging to the warped search segment and its shifted equivalent in the actual frame, respectively:

$$s_{p_w} = \sqrt{\sum_{i=1}^{n}\left(\mathbf{S}_{p_w}(\mathbf{C}_{p_w}(i)) - m_{p_w}\right)^2} \qquad s_a = \sqrt{\sum_{i=1}^{n}\left(\mathbf{I}_a(\mathbf{C}_{p_w}(i) - \mathbf{t} + \mathbf{d}) - m_a\right)^2}$$

iii. Find the displacement vector that produces the highest correspondence score and assign it as motion vector to the centre node

$$\boldsymbol{\nu}_{S_{p\to a}} = \arg\max\{r(\mathbf{d})\} \tag{3.55}$$

With (3.52) we gain our set of shift vectors. Remember that the centre node of the search segment is translated to the origin, thus $\mathbf{C}_{p_w}$ and therfore D contains all pixel coordinates (integer) within the search segment and the maximum displacement included is a shift to the border of the segment in any direction, but not further. This ensures that the mesh topology will always be preserved in the motion tracking, since there can be no overlapping movement during the updating of the mesh node location in the transition from one frame to the next. For the time being this is only true for diagonally neighbouring nodes as can be confirmed in Figure 3.16, but we will show later that with a two-step updating process it covers all neighbouring relationships.

(3.52) has another essential advantage: it allows bigger movements in directions where the limiting nodes are far away and only smaller movements in directions where they are close, instead of searching as usual uniformly within a square-shaped area. In this way it is made sure that areas that were expanded very much in the past motion tracking can easily contract again (and the other way round via the coarse-to-fine strategy) without necessarily changing the location of a centre node at the border of the expanding area. For instance a mesh node situated at the upper lip must not be moved from the lip position during mouth opening. We will return to this example shortly. This models a continuous surface that can be stretched and compressed, but will not disintegrate. Note that we do not employ virtual springs between the mesh nodes, since the computational effort would exceed any manageable measure. With the proper coefficient setting assumed it would probably be advantageous for all skin areas, however in case of the opened/closed mouth or eye blinking it might drive the tracking system into critical errors.

Alternatively to (3.52) one could set $\mathbf{D} = \mathbf{C}_p$, where $\mathbf{C}_p$ the set of all coordinate vectors $\mathbf{p}_p = [x, y]$ of the *unwarped* search segment $\mathbf{S}_p$. This allows the correlation procedure to search around the centre node's predicted location in the incoming frame according to the borders of the unwarped segment. Thus it could for example move across the limits of the interpolated borders of the warped search segment. In this way interpolation errors can be compensated more radically, but if one assumes that the prediction based on head tracking and/or motion tracking on a higher level and subsequent interpolation is relatively reliable one would possibly prefer the more restrictive approach given in (3.52). Very generally speaking the tracking on the higher levels is indeed more reliable, since more pixels contribute to the solution, however, at least a fine tuning of the interpolated nodes is always absolutely necessary and sometimes amounts to a bit more than 'fine', leaving us in some kind of stalemate here. Note that the just described approach has as well as (3.52) the property of avoiding overlap with diagonally neighbouring nodes. It was implemented in the earlier version of the algorithm, for the current one both alternatives are available.

Equation (3.53) looks surprising at first glance: Having a two-dimensional signal, why is there only one summation? But $\mathbf{C}_{p_w}$ contains of course vectors with two-dimensional coordinates, therefore by running through all elements of $\mathbf{C}_{p_w}$ we move through a part of the image plane. It means that our correlation window is no longer square-shaped but rather assumes the shape of *each individual search segment*.

It probably needs no further comment that a correlation window that is adapted to the task outperforms a traditional square-shaped one, even if some of the favourable properties of the cross-correlation in the frequency domain are sacrificed. Nevertheless let us return to the example of a mesh node located directly on the upper lip. Now assume that in a particular frame transition the subject's mouth is opening. That means that in the second frame of the transition a texture map patch appears (usually a dark, almost black area) for which we do not have an equivalent in the first frame. On the coarsest level of the tracking the change is not significant enough to disturb the tracking, that is, mesh nodes situated above the mouth will stay there and mesh nodes on the chin will move downwards with the chin. On the finer levels the neighbouring mesh node below the upper lip centre node will be moved far away due to interpolation (see section 3.3.7), thus the search segment will be warped to have a relatively long wedge-like extension downwards. The upper half of the search segment will be

still undeformed and contain the texture map values of the lip and maybe a bit of the skin above the lip, while the lower half will consist mainly of interpolated values. If the mouth was only very slightly opened before, the downward extension will already have a significant amount of the almost black values that characterised the area within the opened mouth in our example. Trying to determine the correspondence with a thus shaped correlation window improves the chances for a successful location in the second frame enormously.

This is even more important in the case of a mouth closuring gesture and a node located on the upper lip but somewhat off-centre. There the combination of the coarse-to-fine strategy and the individually shaped correlation window helps to thwart the danger of a spurious movement to the side: Since a whole area of intensity values below the mesh node disappears, an undeformed square-shaped correlation window would certainly move towards the centre of the upper lip simply because there - if the mouth is not closed completely - more of the almost black intensity values are preserved below the upper lip.

Clearly using (3.54a) as the function to calculate the correspondence gives the ordinary cross-correlation, (3.54b) the SSD (**S**um of **S**quared **D**istances) and (3.54c) the normalised cross-correlation. So far we used only the normalised cross-correlation for a very simple reason: The correspondence procedure has to be applied to all subbands of a wavelet level (as shown in Figure 3.14) and the three resulting sets of values have to be combined in some way or another to get the final motion vector. Since the normalised cross-correlation returns a value between -1 and 1, the values can be simply added and afterwards (3.55) applied. With the other methods the results must be ranked for each subband (requiring a computationally expensive sorting process) and then the minimum combined rank must be found. The interesting question as to how the results would differ, has not been addressed yet. Neither have we investigated what impact different ways of combining the results of the normalised cross-correlation would have. For instance multiplying the correlation results from each subband would punish candidates that perform badly in only one of the subband, weighting could be used to give a single subband more salience, and it is not clear whether the ranking process described above applied to the normalised cross-correlation values would lead to the same result as if the values are just summed up.

In the earlier implementation the intensity values of the warped search segment and the shifted equivalent areas in the incoming frame were not preprocessed before being submitted to the cross-correlation. However this might not be the optimal choice for the following reasons:

i. Since we assign the motion vector of the search segment to its centre node, it appears sensible to weight the area close to the centre node more strongly than the periphery.

ii. Severe mistracking of a specific node would be less likely to propagate to its neighbours in the next frame-to-frame transition (or by passing from one wavelet level to the next), if the area surrounding the centre node is more strongly weighted than the periphery: Each limiting node of a search segment is a centre node of another one. Therefore after completing the tracking for a specific frame-to-frame transition (or just a single wavelet level) the search segment looks slightly different for the following frame-to-frame transition (or wavelet level). This is of course intentional, in fact, it is one of the strengths of the algorithm. However in case of severe mistracking it can turn into a problem, not directly, because it cannot penetrate into the

correspondence process, but indirectly, by changing the shape of the search segment to an inappropriate form. Imagine that because of a grave lighting change the system would have placed the lower neighbour of our example node at the upper lip *much* too low. Then in the next transition the lower half of the search segment would be the wedge-like elongated triangle. In the correspondence process it would gather correctly the intensity values, which would include in this case probably texture map patches from the lower lip and even the chin. If there would be no movement in this area no problem would arise, but if for example the lips would open the segment would be quite likely located incorrectly in the next frame, because it became unreasonably large in one direction, i.e., a region of the mesh which was supposed to be non-rigid at the specific resolution becomes rigid and thus cannot recover the non-rigid movement. A stronger weighting of the area around the centre node would dampen the propagating effect of the error without sacrificing the positive impact during correct tracking (described above).

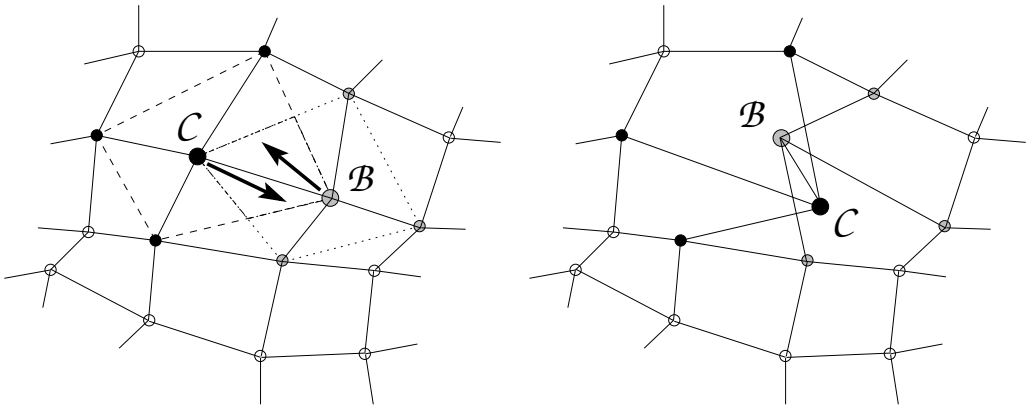
iii. The half-ellipsoid mesh with movable nodes is of course not a very accurate model of the facial surface, which is much too complicated to be modelled in any simple way. With the piecewise affine warping the model approximates more a set of arbitrarily flexible rubber sheets suspended between a wire mesh with joint-like behaviour at the mesh nodes and expandable/contractible but otherwise solid links between the nodes, while with the piecewise bilinear warping the links can bend, too. Since the face has no underlying mesh structure the deviation of model and original due to the impact of the links between the nodes will be more severe at a distance from mesh nodes than close to them, or in other words, only with a mesh resolution approaching infinity we could model any surface behaviour correctly. Therefore weighting the intensity values with some function inversely proportional to the radial distance from the centre node should counteract this deficiency of our model.

Therefore we applied a windowing function to the search segment in the new implementation of the algorithm. Lacking any criterion pointing towards a special window a two-dimensional circular Gaussian window (3.29) appears to be the most natural choice. For the free parameter σ several alternatives are conceivable, e.g., matching it to some fraction of the maximum length of each individual search segment, i.e., forcing the coefficients close to the search segment border to become sufficiently small. We decided to set σ uniformly for all search segments to half of the wavelength corresponding to the centre frequency of the respective wavelet level. Note that the infinite support of the Gaussian function can be considered an advantage in this context, since remote pixels in an exorbitantly enlarged sector still contribute to the correlation albeit very little.

As already mentioned the displacement vector that produced the maximum correlation is finally assigned as motion vector to the centre node of the search segment. We must now return once more to the problem of mesh node movement producing overlap and thus destroying the mesh structure.

Figure 3.20(a) shows the motion vector of two horizontally neighbouring nodes for a movement which rarely occurs but is still within the allowed limits of the tracking algorithm. As depicted in Figure 3.20(a) such a movement would lead to the disintegration of the mesh.

In the earlier implementation of the tracking system our remedy for problems caused by those extreme cases was to limit the movement of a mesh node to half



(a) Movement of mesh nodes (b) Overlap destroying mesh structure

Figure 3.20: Critical mesh node movement

of the distance in each direction, i.e., the shift vectors were extracted from the quadrilateral created by taking the points as vertices that lie halfway on the line between the centre node and its neighbouring nodes. The main disadvantage is of course that the movement resolution does not match the mesh resolution, thus for instance an area could not be compressed (almost) completely from one frame to the next. Given the fundamental assumption for all image motion estimation procedure that the changes from one frame to the next should be small, this is unproblematic. However in the notably less than ideal world of speaking faces captured on video, mouth closuring gestures indeed happen during a single frame transition, though not very often. In these cases the procedure leads to an underestimation of the movement.

Accordingly we used an alternative procedure in the current implementation: The motion tracking on each wavelet level is broken down into two steps. In the first step all mutually diagonally neighbouring nodes relative to an arbitrary starting node are tracked, that is, roughly half of all mesh nodes. As shown in Figure 3.16(a) they are not able to produce any overlap in a single frame transition. The node positions of all tracked nodes are updated and the ones not yet tracked are interpolated yielding a new deformed mesh. Then in step two the set of the remaining nodes is tracked. Since it contains only diagonally neighbouring nodes as well, the danger of overlapping movements is avoided again. After the updating of these node's positions the tracking on the particular level is completed. One could object that the process creates two classes of nodes, since for the latter half the shape of the search segment is changed just before they are submitted to the warping and correlation procedure. However the search segments of all nodes almost always undergo some changes because of the effect of head motion or tracking results on a higher level. What remains constant and ensures consistency in the tracking over a sequence of frames is that some area around the centre node is used for the tracking and that different shapes of the search segment cannot seep into the correspondence procedure itself.

3.3.7 Interpolation of mesh node coordinates of next finer mesh

The last step in the loop through the wavelet and tracking levels is the interpolation of the location of the nodes that come in on the next resolution levels. Unlike the search segment the mesh coordinates are always full matrices, though not containing regularly spaced values, which permits usual two-dimensional interpolation methods to be used. For example, the MATLAB function `ZI = GRIDDATA(X,Y,Z,XI,YI)`

> ... fits a surface of the form Z = F(X,Y) to the data in the (usually) nonuniformly-spaced vectors (X,Y,Z) GRIDDATA interpolates this surface at the points specified by (XI,YI) to produce ZI. The surface always goes through the data points.

as the `help` command points out. In our case `X` und `Y` are the image coordinates of the tracked nodes, `Z` is one of the two components of their respective motion vectors, and `XI` und `YI` the image coordinates of *all* mesh nodes of the next resolution level. The function has to be executed twice to get the `x` and the `y` component of the motion vectors. The obtained full motion vectors for all mesh nodes are than just added to the image coordinates in their original state before the motion tracking of this level.

3.3.8 Reversing the mesh projection

After completion of the actual motion tracking the effects of perspective projection and head motion (translation and rotation) on the mesh must be reversed, since our final result should be a sequence of stabilised meshes (one for each frame) preferably in physically meaningful unit. Thus we step here from mere tracking to relative (uncalibrated camera) or absolute (calibrated camera) measurement of face motion. In principle this is straightforward since for the combined transformation matrix

$$\mathbf{Q} = \mathbf{S}\,\mathbf{R}\,\mathbf{T} \tag{3.56}$$

accounting for scaling $\mathbf{S}$, rotation $\mathbf{R}$ and translation $\mathbf{T}$ of the ellipsoid the inverse $\mathbf{Q}^{-1}$, i.e.,

$$\mathbf{I} = \mathbf{Q}\,\mathbf{Q}^{-1} \tag{3.57}$$

always exists. But unfortunately the non-linear part of the perspective projection (division by the depth coordinate) makes the process a bit more complicated: For all mesh nodes that were moved by the tracking we simply do not have their new depth coordinate in the image coordinate system. If the ellipsoid was still in its undeformed starting state they could be easily recovered by solving the ellipsoid normal form given in (3.26) for the desired depth coordinate z:

$$z = \sqrt{c^2 \left(1 - \frac{x^2}{a^2} - \frac{y^2}{b^2}\right)} \tag{3.58}$$

However, this is not the fact, rather the ellipsoid was translated to a different origin (the calculated centre of rotation of the head), rotated and translated according to head movements and projected. Since the actual x and the y image coordinates of the ellipsoid are depending on the z coordinates, the z coordinates cannot be recovered directly with (3.58). But since we know all rotation, translation, etc. parameters there should be a unique solution, we only need to account for the effects they have. Thus re-formulate (3.26) as follows:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} - 1 = 0 \tag{3.59}$$

Let us now look for simplicity at one of the terms only:

$$\frac{x^2}{a^2} + \dots \tag{3.60}$$

If nothing would have changed the ellipsoid except the translation to a different origin, (3.60) could be expressed as

$$\frac{(x_o + o_x)^2}{a^2} + \dots \tag{3.61}$$

where o_x is the inverse (negative) of the x translation component. Now we add rotation which is dependent on the coordinates in all three components

$$\frac{(r_{1_x} x_r + r_{2_x} y_r + r_{3_x} z_r + o_x)^2}{a^2} + \dots \tag{3.62}$$

where r_{1_x},r_{2_x}, and r_{3_x} are rotation coefficients from the inverse of the rotation matrix $\mathbf{R}$ that we used to rotate the ellipsoid according to the rotational head motion parameters (the structure of the matrix was developed in section 2.2.2). r_{1_x},r_{2_x}, and r_{3_x} constitute the first row of this matrix, i.e., the coefficients that have an impact on the x-coordinate. x_r,y_r, and z_r are the mesh node coordinates after origin translation and rotation.

We continue in the same way with the head motion translation component

$$\frac{(r_{1_x}(x_t + s_x) + r_{2_x}(y_t + s_y) + r_{3_x}(z_t + s_z) + o_x)^2}{a^2} + \dots \tag{3.63}$$

with s_x,s_y, and s_z being the inverse (negative) of the respective translation. In the next to last step we have to consider the scaling, i.e., the focal length. Since in the projection it affects only the depth component $z_i = 1/f\, z_t$ we have

$$\frac{(r_{1_x}(x_t + s_x) + r_{2_x}(y_t + s_y) + r_{3_x}(z_i f + s_z) + o_x)^2}{a^2} + \dots \tag{3.64}$$

And finally the division by the unknown depth coordinates z_i is reversed into a multiplication

$$\frac{(r_{1_x}(x_i z_i + s_x) + r_{2_x}(y_i z_i + s_y) + r_{3_x}(z_i f + s_z) + o_x)^2}{a^2} + \dots \tag{3.65}$$

Now we have incorporated all changes in the normal form (assuming that the y-term and the z-term were treated in the same way) and we can start to solve for z_i.

Isolating z_i within the basis of the exponent gives

$$\frac{(r_{1_x} x_i + r_{2_x} y_i + r_{3_x} f)\, z_i + (r_{1_x} s_x + r_{2_x} s_y + r_{3_x} s_z + o_x))^2}{a^2} + \ldots \tag{3.66}$$

Expanding the square and returning to the full equation yields

$$\begin{aligned} \frac{q_1^2 z_i^2 + 2\, q_1\, q_2 z_i + {q_2}^2}{a^2} + \frac{q_3^2 z_i^2 + 2\, q_3\, q_4 z_i + {q_4}^2}{b^2} + \\ \frac{q_5^2 z_i^2 + 2\, q_5\, q_6 z_i + {q_6}^2}{c^2} - 1 = 0 \end{aligned} \tag{3.67}$$

where

$$\begin{aligned} q_1 &= r_{1_x} x_i + r_{2_x} y_i + r_{3_x} f \\ q_2 &= r_{1_x} s_x + r_{2_x} s_y + r_{3_x} s_z + o_x \\ q_3 &= r_{1_y} x_i + r_{2_y} y_i + r_{3_y} f \\ q_4 &= r_{1_y} s_x + r_{2_y} s_y + r_{3_y} s_z + o_y \\ q_5 &= r_{1_z} x_i + r_{2_z} y_i + r_{3_z} f \\ q_6 &= r_{1_z} s_x + r_{2_z} s_y + r_{3_z} s_z + o_z \end{aligned}$$

Rewriting (3.67) into a standard quadratic equation we obtain

$$\begin{aligned} \left(\frac{q_1^2}{a^2} + \frac{q_3^2}{b^2} + \frac{q_5^2}{c^2}\right) z_i^2 + 2\left(\frac{q_1 q_2}{a^2} + \frac{q_3 q_4}{b^2} + \frac{q_5 q_6}{c^2}\right) z_i + \\ \left(\frac{q_2^2}{a^2} + \frac{q_4^2}{b^2} + \frac{q_6^2}{c^2} - 1\right) = 0 \end{aligned} \tag{3.68}$$

which can be solved for z_i by means of standard algebra

$$\begin{aligned} z_{i_{1,2}} = \frac{-2\left(\frac{q_1 q_2}{a^2} + \frac{q_3 q_4}{b^2} + \frac{q_5 q_6}{c^2}\right)}{2\left(\frac{q_1^2}{a^2} + \frac{q_3^2}{b^2} + \frac{q_5^2}{c^2}\right)} \pm \\ \frac{\sqrt{4\left(\frac{q_1 q_2}{a^2} + \frac{q_3 q_4}{b^2} + \frac{q_5 q_6}{c^2}\right)^2 - 4\left(\frac{q_1^2}{a^2} + \frac{q_3^2}{b^2} + \frac{q_5^2}{c^2}\right)\left(\frac{q_2^2}{a^2} + \frac{q_4^2}{b^2} + \frac{q_6^2}{c^2} - 1\right)}}{2\left(\frac{q_1^2}{a^2} + \frac{q_3^2}{b^2} + \frac{q_5^2}{c^2}\right)} \end{aligned} \tag{3.69}$$

The plus and minus in the equation gives, so to speak, the front and the back side z_i coordinates of the ellipsoid. Since the general orientation of the ellipsoid

is usually fixed (the half used for tracking faces the camera) it can be determined beforehand which part of the solution is matching the image x and y coordinates of the mesh.

After determining the depth coordinates the ellipsoid mesh can be transformed into standard form and written to file, its nodes now representing the intrinsic face motion relative to the state of the face in the reference frame.

3.4 Putting it all together

Figure 3.21 shows a schematic overview over the data flow of the whole system. On the left hand side the initialisation is shown, which needs the video sequence and the head motion data to derive the mesh parameters based on a reference frame and to calibrate the camera. Arrows indicate where the initialisation procedure contributes information to the motion tracking procedure, e.g., the head motion parameters are just passed to the tracking routine.

The main motion tracking box stands for the loop through the frames that were selected for tracking. It contains itself three nested loops, one for the wavelet levels, one for the mesh nodes, and the final one for the correspondence determination on each subband (horizontal, vertical and diagonal). Note that the first frame that is really tracked has as its frame n-1 the reference frame.

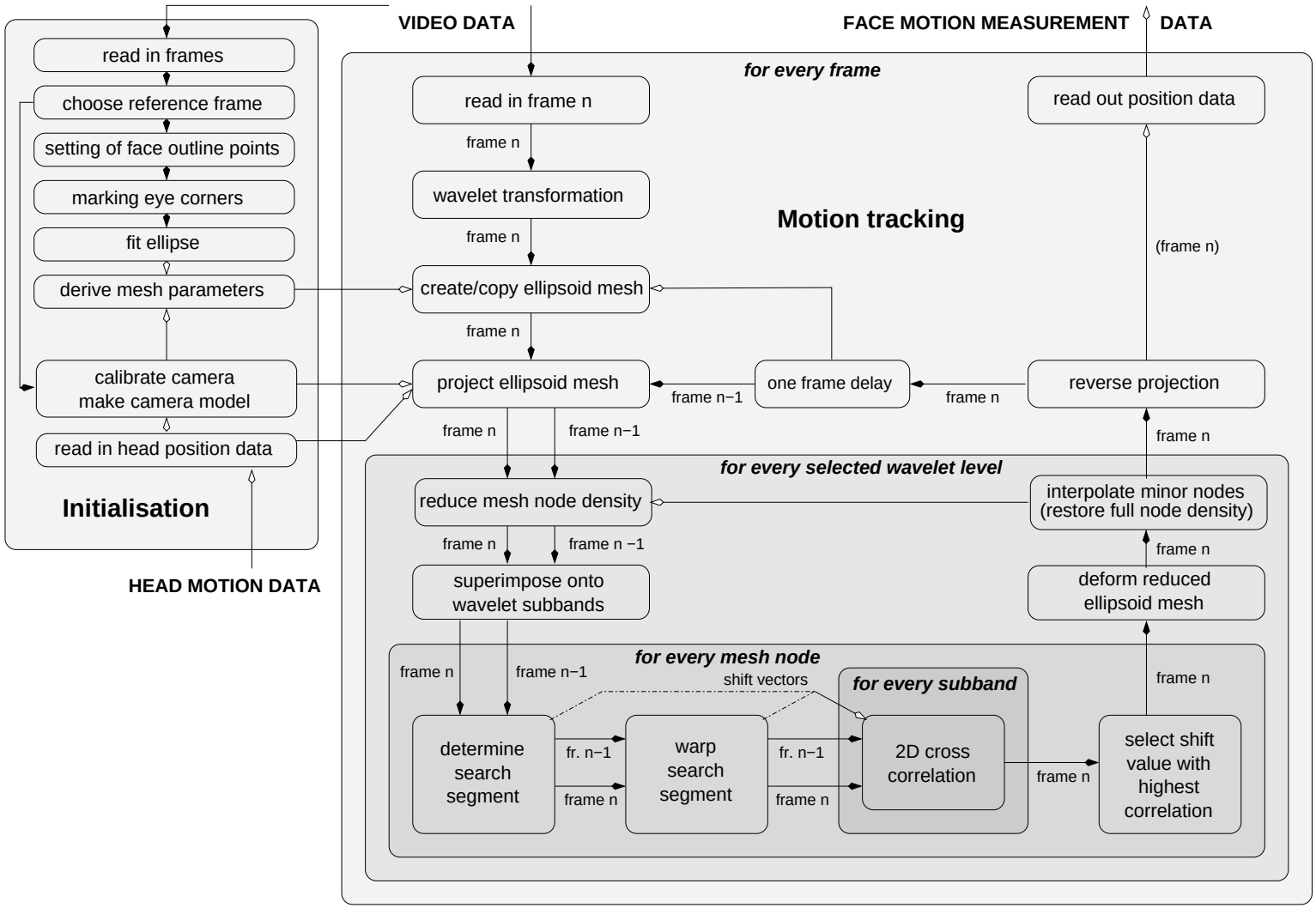


Figure 3.21: System overview

Chapter 4

Validation

Video-based methods to measure face motion are in general intricate to evaluate for several reasons:

i. The location change of parts of the face surface cannot be known beforehand. We do not understand face motion enough to construct a reliable model for prediction, not even for very simple experimental tasks like mouth opening.

ii. It is difficult to apply other measuring methods without interfering with the video-based method. Reliable marker-based methods are at hand, but the makers introduce a strong image gradient at their location and thus facilitate the tracking considerably (they act as passive markers for the video-based tracking). Potential wires attached to active markers on the other hand disturb video-based methods.

iii. Face motion is not symmetric. Numerous studies including our own have shown this. That prevents using a marker based method at one side of the face and the video-based on the other half during the same experiment and evaluate the comparison. As an approximation it might be acceptable, but the discrepancy will be to high to base an evaluation on it.

iv. Manual tracking is difficult and error-prone. Human observers have excellent abilities to perceive location changes of the face surface in motion sequences. But if selected points in still images should be tracked the task becomes extremely challenging.

v. 'Nothing compares' to the facial surface. Morphological structure and textural appearance are highly complex and unique, making it very demanding, not so say impossible, to find a sufficiently similar substitute with known motion parameters or build a comprehensive simulation (the latter at least for a foreseeable future). Up to now animated faces had to concede to the complexity of real faces and had to allow strong simplification, for instance, a grossly smoothed or idealised texture map of the artificial face.

As consequence of this we certainly will not get a single validation number asserting the goodness of the tracking and allowing comparison to other tracking methods. Rather more qualitative methods relying on human visual inspection or comparisons of large amounts of data acquired asynchronously with different methods will take a centre stage in the following. However, even here strong differences exist in the usefulness of the method, e.g., just presenting the flow field

superimposed on the image sequence as in the online demo of the tracking results described in Wu et al. (1998)[1] does not allow a serious verification of the tracking goodness.

Table 4.1 summarises some of the properties of the data sets we will refer to in the next sections. All recordings were made at the speech production laboratories at ATR near Kyoto (Japan). The subjects were seated approximately 3 meters away from the recording Betacam video camera and wore a headmount with attached OPTOTRAK sensors to track head movements. During the experiment they were reading each sentence from a display monitor. The sound was recorded with a directional microphone. The video data were digitised and the single frames/fields were stored as lossless compressed TIF-files.

Table 4.1: Data sets used in the development and evaluation of the tracking algorithm

Code	**Corpus**	**Speaker**	**Language**	**Remark**
EVB-TEST5	5 test sentences	Male American English native speaker	American English	Experimental, used for algorithm development, normal room lighting
EVB-CID100	Full CID[a] corpus	Male American English native speaker	American English	Normal room lighting
EVB-CID15	First 15 CID[a] sentences	Male American English native speaker	American English	normal room lighting
SAE-KAN12	12 sentences[b]	Female Japanese native speaker	Japanese	Additional lighting
ACA-KAN12	12 sentences[b]	Female Japanese native speaker	Japanese	Additional lighting
DCA-MOB12	12 sentences[c]	Male American English native speaker	American English	Additional lighting
ENA-DOM12	12 sentences[d]	Male Brazilian Portuguese native speaker	Brazilian Portuguese	Additional lighting
CHK-MOO12	12 sentences[e]	Male German native speaker	German	Additional lighting

[a] The **C**entral **I**nstitute for the **D**eaf (CID) 'Everyday Sentences', 100 phonetically balanced sentences (see Davis and Silverman, 1970).
[b] First 12 sentences from the phonetically balanced Kanzaki corpus, developed by Rika Kanzaki at ATR-HIP, Kyoto, Japan.
[c] First 12 sentences from the novel 'Moby Dick' by US-American novelist Herman Melville.
[d] First 12 sentences from the novel 'Dom Casmurro' by Brazilian novelist Joaquim Maria Machado de Assis
[e] First 12 sentences from the 'Moorsoldaten' story, written by the author for articulatory experiments within the 'DFG-Schwerpunktprogramm Sprachproduktion'

4.1 Animation

Using the face motion tracking results to create a photo-realistic animation of the talker's face can only be accomplished, if the tracking points are truly globally distributed over the face. If this is not the case at least a sufficiently detailed

[1] Available at `http://www-2.cs.cmu.edu/afs/cs/project/face/www/Facial.htm`

computer graphic animation model must be available. In the latter case motion parameter e.g,. derived from feature tracking, would drive certain portions of the model only and the inherent model structure would ideally take care of the rest. In the former case, which is ours, the texture map of the face taken from an arbitrary frame can be deformed according to the measured motion.

Thus as a prerequisite, a high resolution texture map of the ellipsoid mesh model must be extracted. There are several ways to accomplish that, two of them will be described here.

i. The first method was used in Kroos, Masuda, Kuratate, and Vatikiotis-Bateson (2001). The texture map is taken (arbitrarily) from the input frame where the motion tracking started. At this stage the tracking mesh is translated and rotated in the image plane to fit the face and also projected, but is not deformed with respect to the motion tracking. That is, the coordinates of all mesh nodes in the image are already known and exist in a normalised, unfolded two-dimensional matrix corresponding to the two angular coordinate matrices that were used to generate the ellipsoid in the initialisation process. These correspond to azimuth angle θ and polar angle ϕ in the parameterisation formula for the ellipsoid from standard analytical geometry (equation (3.27)).

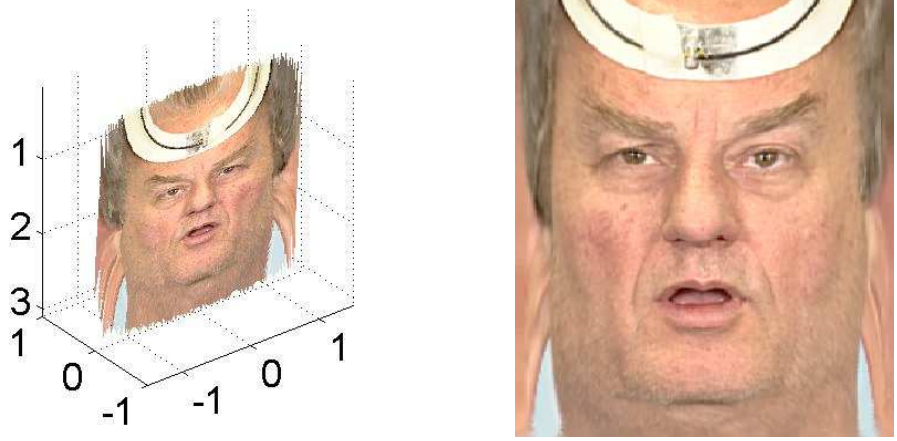
Now for any pixel lying inside the projected mesh's outline, its image plane coordinates are easily obtained by reversing translation, rotation and projection. However, the corresponding depth values have be computed by solving the ellipsoid normal equation taking into account the rotation, translation and projection of the mesh in the reference frame (see section 3.3.8). Then, reversing the parameterisation formulae and putting in the three-dimensional pixel coordinates 'unfolds' the ellipsoid surface yielding normalised two-dimensional angular coordinates for every pixel (see Figure 4.1(a)).

Using those and the θ and ϕ values of the mesh nodes, arbitrarily coarse or fine texture maps can be obtained via interpolation. Of course, the information content of the texture maps is limited by the pixel density of the video frame. Figure 4.1(b) shows an example.

ii. If the tracking mesh is undeformed, that is, the texture map is taken from the reference frame, an arbitrarily coarse or fine mesh is generated. Usually it would be much finer than the one used for tracking. It is then translated, rotated and projected according to the known parameter values for the reference frame. Then by interpolation the intensity value at each node in the image is determined. If zero-order interpolation (see 3.3.5) is applied this amounts only to rounding of the coordinates of the mesh nodes and assigning the intensity value of the pixel with the same coordinates to them. After that the mesh is just 'unfolded' as described under item *i.*.

If the tracking mesh is already deformed, it must be brought in standard form first and must then be unfolded. Now different resolutions can be obtained by interpolation. The remainder of the procedure is the same as for the undeformed mesh.

The texture map is then applied to the sequence of deformed ellipsoid meshes using the graphics capabilities of the 3D rendering subroutines of MATLAB. Creating a new movie sequence by juxtaposing the original video frame and the animation can be considered the best of all quantitative, visual evaluation methods.



(a) The ellipsoid surface 'unfolded'

(b) Full interpolated texture map

Figure 4.1: Texture map extraction

Any deviation is is immediately spotted. Figure 4.2 shows some stills from a thus obtained movie.

Animations could also be used to do an evaluation based on perception experiments with humans. It could be tested whether or not the reconstructed motion sequence elicit the same speech related phenomena as the original, for example improving intelligibility in acoustically noisy environments or causing the McGurk-Effect.

4.2 Difference images

The first suggestion for an evaluation at presentations of parts of the earlier algorithm was that we should calculate difference images between the original images and reconstructed images based on the motion tracking. However, again this is merely a qualitative evaluation and moreover it depends critically on the reconstruction method, in particular on the chosen resolution of the texture map and whether the reference frame comprises all areas that become visible later on as consequence of the tracking. For instance, if the mouth is opened only slightly in the reference frame, the teeth might be not visible, because not enough light can pass into the oral cavity. As a result there would appear strong peaks in the difference images when later in the sequence the mouth is opened completely, even if the tracking were perfect. On the other hand a human perceiver might not be disturbed at all by the missing visible teeth, since the effect could be caused equally by reduced front lighting.

In addition difference images emphasise correct location of edges, since a slight mispositioning of an edge creates a strong peak. The paramount perceptual relevance of 'hard' edges, to use this term from modern art, i.e., scale-space independent edges where all frequencies contribute, is at least doubtful. The state-

Figure 4.2: Pairwise comparison of the original video image (right side) and reconstructed image (left side) by applying the texture map of the reference frame to the deformed mesh. Frames 1, 8 (first row), 16, 22 (second row), 49, and 66 (third row).

ment could actually be generalised for difference images per se: No arguments are known to us showing that the perception of a difference image has any fixed relationship to the perception of the similarity between the original images used to create the difference image. Therefore we did not use difference images for evaluation.

4.3 Comparison of Principal Components

While a direct comparison between results from the video-based method and OPTOTRAK (i.e. using the same trials) is not feasible (see above), a comparison is still possible by looking at the overall spatial behaviour of larger sets of data reduced by statistical means to essential components, e.g., by using Principal Component Analysis. Of course it is only sensible if the material is as sufficiently similar be-

yond being produced by the same speaker. This includes using the same corpus and recording in a comparable experimental situation. These conditions were fulfilled for the EVB-CID100 data set, of which 71 sentences were tracked with the video-based method and a comparable OPTOTRAK data set for the same speaker containing a subset of 58 of the 71 sentences was available from an earlier experiment.

Even then the globally distributed but relatively coarse (low image resolution) video-based tracking might show different statistical behaviour if compared with the sparsely distributed but very fine OPTOTRAK tracking. Also the video-based tracking is only two-dimensional while OPTOTRAK returns three-dimensional coordinates. Accordingly PCA was performed based on the covariance matrix of four different data subsets. The first data subset contained movement data from the video-based method for a region of the lower face (242 mesh nodes) corresponding to the area measured by 18 OPTOTRAK markers. Figure 4.3 shows the measurement locations.

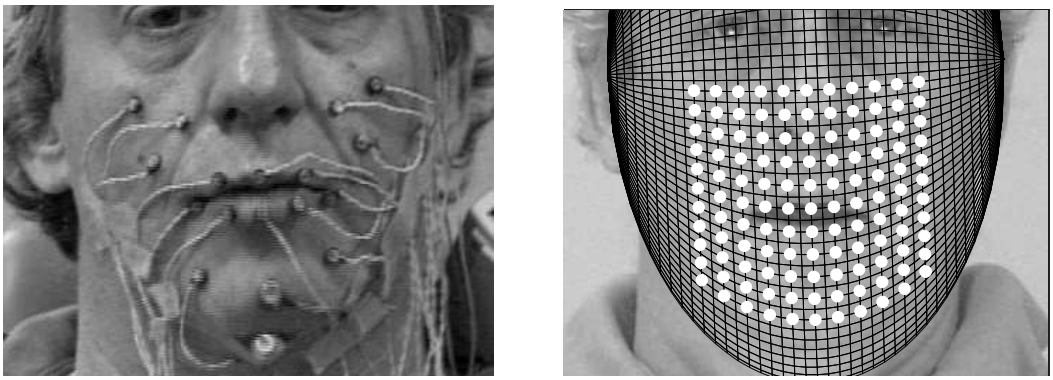


Figure 4.3: Location of the OPTOTRAK markers (left side) and the selected mesh nodes of the video-based method (white dots, right side)

The second data set consisted only of the 18 mesh nodes that were at similar locations on the face as the OPTOTRAK markers. The third data set included the OPTOTRAK movement data, as did the fourth, however this time the 'depth' dimension (perpendicular to the video image plane) was disregarded.

Figure 4.4 shows the cumulative amount of variance recovered by the first 36 components for all four data sets. Note that 36 is the maximum number of components that can be derived from 18 two-dimensional marker or mesh node measures. Obviously, many more components are needed to account for the same level of variance in the video-based method as in the OPTOTRAK data. In part this is due to the large difference in measurement resolution of the two systems; compare 0.6 mm pixel resolution of the video image with .01 mm position accuracy of the OPTOTRAK.

However, despite this difference in accuracy, the two analyses capture the same motion characteristics; e.g., the first components are fairly similar for the two methods. This can be seen in Figure 4.5 which shows a comparison of the first three components by projecting them two standard deviations on each side

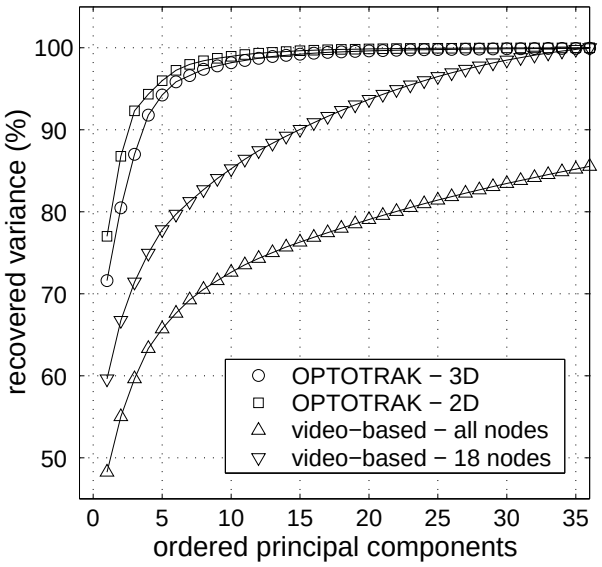


Figure 4.4: Recovered variance by the first 36 principal components: three-dimensional OPTOTRAK data (first 36 of 54 components), two-dimensional OPTOTRAK data (image plane, all components), video-based method using all originally selected nodes (first 36 of 242 components) and video-based method using 18 nodes corresponding to OPTOTRAK marker locations (all components).

of the mean.

The uppermost part of the figure depicts the results from the video-based analysis with all originally selected nodes included and the texture map applied to the deformed mesh. The three components account for 48.2, 6.8 and 4.6 % of the total variance in the data set. The centre part shows the results from the video-based analysis limited to 18 nodes; the three components account for 59.6, 7.1, and 4.6 % of the total variance. And the lower part displays the results from the two-dimensional OPTOTRAK data set with the components accounting for 77.0, 9.8 and 5.6 % of the total variance. The first component clearly corresponds to the contribution of the jaw to the face motion. This does not mean, however, that jaw motion effects are uniformly oriented. The jaw moves the whole chin area vertically, but its effect on the soft tissue of the cheeks and mouth corners is more lateral than vertical, due to structural constraints.

Less clear is why the percentage of recovered variance for the jaw's contribution is so different for each analysis. One possibility is that positioning of the small set of markers on the chin and around the lower lip overemphasises the contribution of the jaw. However, in previous work (Yehia et al., 1998), it was shown that the jaw component was the major contributor to the motion of *all* markers, not just those of the lower face region. Another possibility is that the video-based method underestimates the jaw's contribution to face motion at high speeds, even though the distribution of the effect on the various nodes has been shown to be coherent through examination of the node velocities.

Note that the second and the third component in the video-based method apparently changed their order in the OPTOTRAK data. The second component does

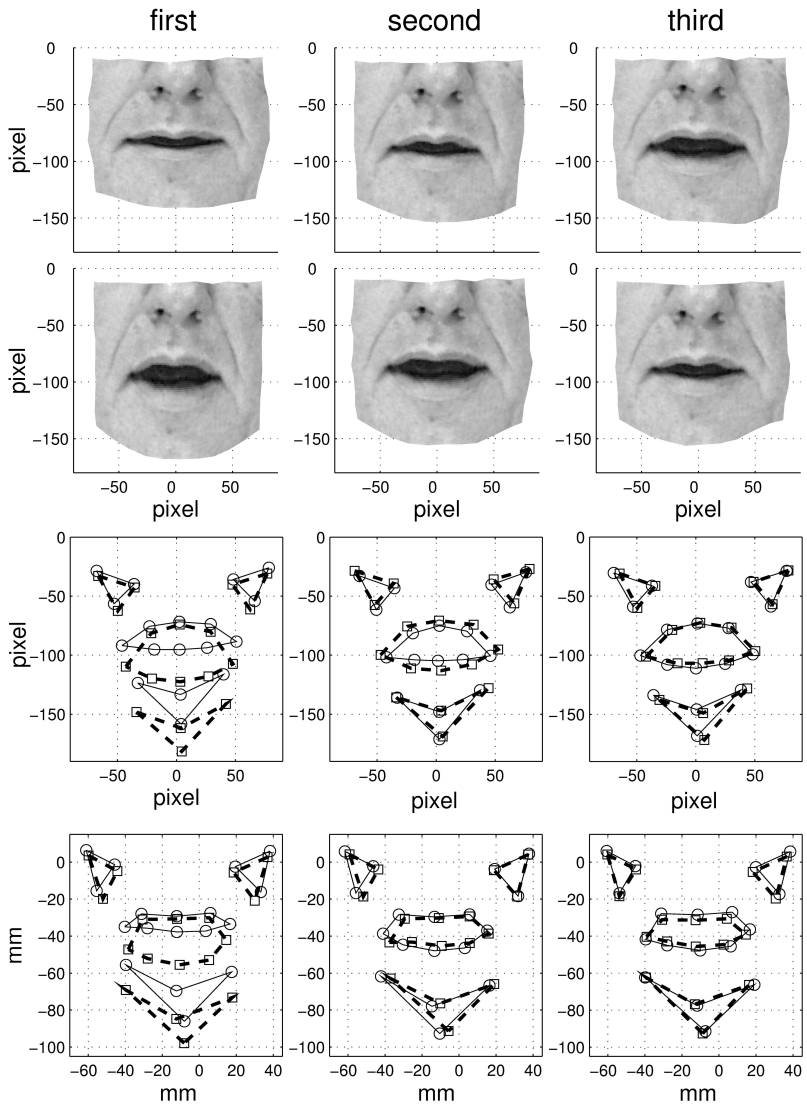


Figure 4.5: Visualisation of the first three principal components by calculating the coordinates of the data points, which are plus/minus two standard deviations apart from the average face in each component's direction. The upper part of the figure shows results from the video-based method using all mesh nodes, the centre part shows results form the video-based method using only 18 nodes corresponding to OPTOTRAK marker locations and the lower part shows results from OPTOTRAK data, where only the two image plane dimensions were regarded. In the upper part the top row corresponds to +2 stds and the bottom row to -2 stds. In the centre part and lower part circles and solid lines correspond to +2 stds and squares and dashed lines to -2 stds. The connecting lines are drawn only in order to facilitate the comparison.

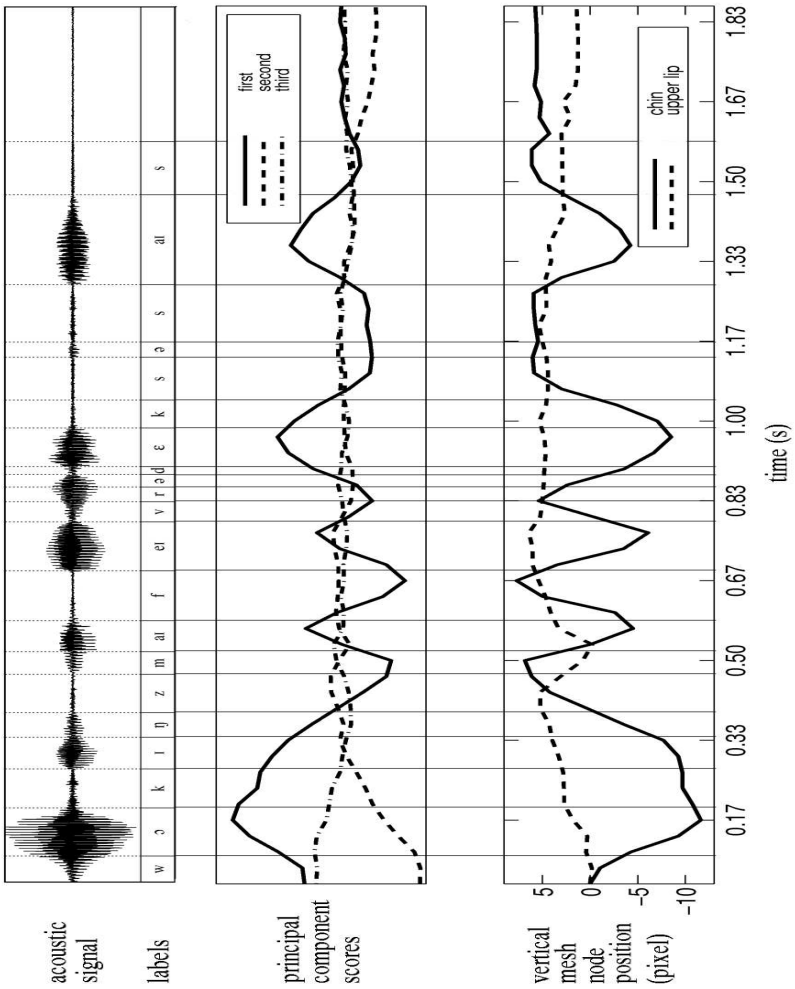


Figure 4.6: Scores of the first three principal components and vertical coordinates of two selected mesh nodes are shown over the time course of the first sentence of the CID corpus ('Walking is my favourite exercise'). The top panel shows the acoustic signal and a broad phonetic transcription. The centre panel shows the scores of the first three principal components. The scores of the component are not scaled, i.e. their variances equal their respective eigenvalues, thus showing their relative importance (the unit, however, is meaningless). The bottom panel shows the vertical coordinates of two of the 18 mesh nodes corresponding to OPTOTRAK marker locations (see Figure 4.5): 'chin' corresponds to the upper centre of the four chin nodes, 'upper lip' corresponds to the upper centre of the eight mouth nodes.

not account for any jaw motion; it corresponds entirely to deformation of the mouth and lip area. This influences the cheek tissue in a manner complementary to the effect of the first component and results in a tightening of the lip oval. This can be seen most clearly in the second component of the 18-nodes video-based analysis and the third component of the OPTOTRAK data, and may account for a substantial part, but not all, of the lip rounding/spreading mechanism. The third component is also independent of jaw position and contributes to mouth and lip shape in a way similar to the second component. The major difference between the second and third components is in their opposite effects on the cheek behaviour associated with a particular mouth/lip shape. This is most evident on the right side of the image (left hand side of the subject).

It is exemplified in Figure 4.6, which shows the scoring of the first three components for the first sentence of the CID corpus, 'Walking is my favourite exercise', and the vertical coordinates of two selected mesh nodes. In order to evaluate the extent that the principal components can be interpreted in a physical sense, we segmented and labelled all vowels in the data set used for the video-based tracking and examined the scoring of the components for each segment by its phonological category. However, no conclusive results were found for any analysed phoneme. Manual inspection of a subset of segments confirmed that the speaker showed very little 'expected' behaviour so long as the whole segment was taken in to account and not just a 'magical' moment picked out within the segment. For example, lip rounding in rounded vowels was not very prominent.

4.4 Comparison with manual tracking

Since visual inspection of the resulting animations is not sufficient to judge the accuracy of the tracking algorithm, and PCA based comparisons with OPTOTRAK only access the general behaviour, we decided to track manually nine points on the subject's face for the first 8 of 15 sentences (877 of 1363 image fields) of the EVB-CID15 data set.

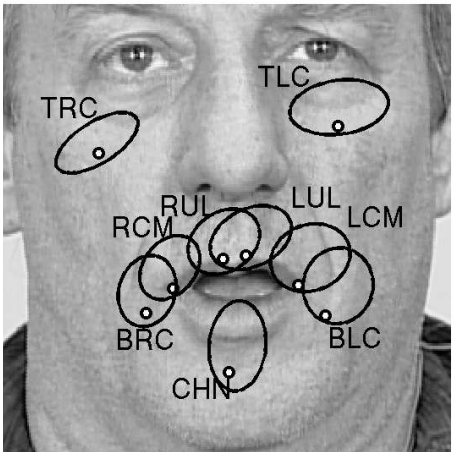


Figure 4.7: Location and movement range of the manually marked points (head motion not removed)

Ideally for this test, the points should be distributed randomly on the face, but it was impossible to find enough arbitrarily assigned points that could be identified reliably in every frame. Therefore, landmark coordinates were used comprising the mouth corners, two points on the upper lip, and five points on the cheeks and chin marked by small blemishes (not visible in the images reproduced for this thesis). Figure 4.7 shows the locations (filled circles) and the movement ranges (including head motion). For the ellipses enclosing each location, size was three times the standard deviation and axis orientation was derived from PCA of the manually tracked position coordinates.

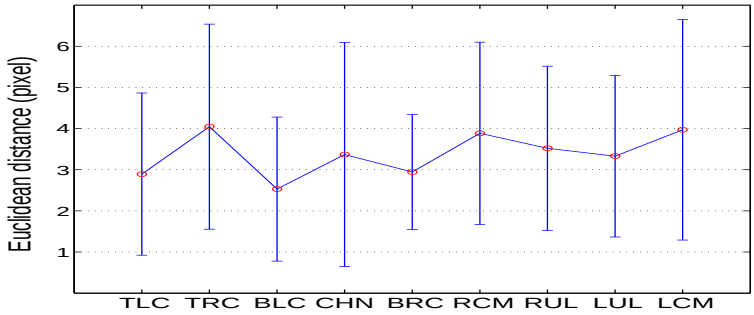


Figure 4.8: Means and standard deviations of the discrepancy between manual and automatic tracking of nine points on the cheeks (TLC, TRC, BRC, BLC), upper lip (RUL, LUL), lip corners (RCM, LCM), and the chin (CHN).

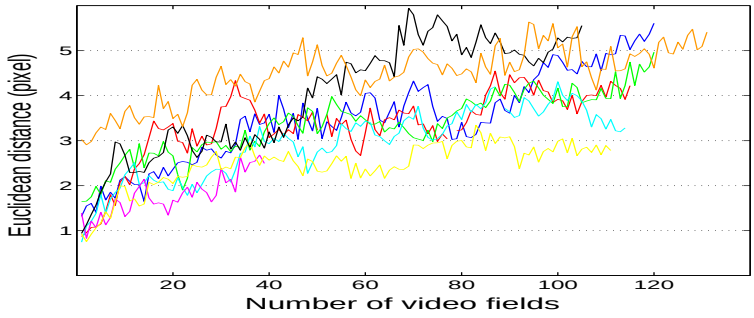


Figure 4.9: Mean discrepancy of all points over time for each sentence

The manual and the automatic tracking results were compared by calculating the Euclidian Distance between each manually marked point and the mesh node closest to it in the global reference frame, taking it as reference (i.e. zero), and then computing the distances between node-point pairs over time. Figure 4.8 shows the discrepancy between the methods for the nine points for all image fields tested. As can be seen, there is a mean discrepancy (change in Euclidean Distance) of 3-4 pixels. Neither the mean nor the standard deviation of discrepancy seems to depend on the location or degree of face motion e.g., compare the rela-

	Bandwidth (cycles/face width)	Centre frequency (cycles/face width)	Centre frequency (cycles/degree visual angle)
F1	1.8-3.7	2.7	0.36
F2	3.7-7.3	5.5	0.73
F3	7.3-15	11	1.46
F4	15-29	22	2.92
F5	29-59	44.1	5.85

Table 4.2: Spatial frequency bands (from Munhall et al., in press)

tively motionless upper cheek with highly mobile chin. It is clear, however, that the discrepancy increases over time. Figure 4.9 shows the generally monotonic growth in mean discrepancy for all markers over the time course of each sentence. The tendency is common to almost all more complex image motion estimations methods if they are frame-to-frame based: the error accumulates slowly.

4.5 Comparison with human auditory-visual speech perception

Beyond direct evaluation interesting insight was won in Munhall et al. (in press) in which spatial frequencies are used by human observers in auditory-visual speech perception, thereby providing a convincing argument for the wavelet-based approach in machine tracking of face motion during speech. Munhall and colleagues tested spatial frequency bandpass filtered image sequences of a talker in an audiovisual speech-in-noise task in two experiments.

As already mentioned in the introduction seeing the speakers face enhances intelligibility in acoustically unfavourable environments. In the first experiment the performance of subjects in recognising correctly keywords of the CID corpus was tested where the original audio track was severely degraded by superimposed multi-speaker babble. The original video sequence was converted to grayscale and filtered with five different one octave-wide bandpass filters yielding together with the audio-only and the full video (grayscale) seven test conditions. Table 4.2 shows the bandwidth and the centre frequencies.

Table 4.3 shows the mean percentage of correct answers over all 42 subjects.

Presentation condition	Percent correct keywords
Full face	66.38
F1	49.52
F2	46.76
F3	55.9
F4	53.81
F5	31.9
Auditory only	36.67

Table 4.3: Identification results (from Munhall et al., in press)

In the second experiment with 90 subjects the viewing distance was varied.
Munhall et al. (in press) summarises:

> Experiment 1 showed that all but the lowest spatial frequency band that we tested enhanced auditory speech perception, however, none of the individual frequency bands reached the accuracy level of the unfiltered images. The band-pass conditions showed a quadratic intelligibility pattern with the peak intelligibility occurring in the mid-range filter band with center frequency of 11 c/face. Experiment 2 showed that this pattern did not vary as a function of viewing distance and thus that object-based spatial frequency best characterized the data.

Note that the spatial frequency conditions denoted as F2, F3, and F4 correspond exactly to the wavelet levels 5, 4, and 3, respectively used in the motion tracking. Actually the stimuli images were produced with the (slightly modified) routine implemented for the tracking procedure. Therefore we can be sure that most of the phonetic information humans can infer from image sequences can be extracted from the spatial frequency domain we are relying on for the tracking (see also MacDonald, Andersen, and Bachmann, 2000).

Chapter 5

Conclusion

5.1 Summary

We presented a system for video-based analysis of face motion during speech. The core of it consists of an algorithm to measure face motion from image sequences. Additional features ensure that the audio track, the acoustical speech signal, is synchronised with the face motion measurement, external head motion data can be integrated and the measurement data itself can be accessed at will for further analysis.

The tracking algorithm has two stages - an initialisation phase and the actual frame-to-frame image motion tracking. The initialisation procedure generates a parametrised ellipsoid mesh, scales it to the size of the subject's face in a user-chosen reference frame and places it to cover the face area. The mesh's function in the subsequent motion tracking is to provide anchor points for the tracking and record location changes of small parts of the facial surface. To achieve size accommodation and placement of the ellipsoid, the user is required to mark a few points at the outline of the face and the inner or outer eye corners. An ellipse is then fit to the outline points, the orientation of which is constrained to the slope angle of the line connecting the eye corner points. From the ellipse the ellipsoid parameters are derived. Also in the initialisation phase a camera model is instantiated, an adapted version of the ideal pinhole camera model.

The motion tracking procedure uses a multiresolution analysis in the strict sense for the image data and - adapted to it, but formulated in a less strict sense - for the mesh resolution, i.e., a set of ellipsoids meshes with varying node density is applied for the tracking in a coarse-to-fine strategy and the tracking results are refined at each step. The goal of the multiresolution decomposition of the video images is to obtain spatial frequency band-limited subbands that are mutually orthogonal and orientation sensitive in three major directions (horizontal, vertical, and diagonal). This is accomplished by a discrete-space wavelet transform of the image data realized as cascade filter bank with pairwise low and high pass half-band filters. The algorithm then loops through selected levels of the wavelet transform projecting the ellipsoid mesh onto the subband 'images' using the camera model and following head movements by considering external head motion data.

The mesh resolution is reduced at the beginning of the tracking of a frame-to-frame transition and superimposed onto a higher wavelet level decomposition of each of the two frames under investigation. Search segments are defined as the

partial face texture map contained in quadrilaterals created by the four neighbouring nodes surrounding an arbitrary centre node. The set of all search segments covers the entire facial surface with overlapping to ensure on the one hand that enough area is included and on the other hand that a relatively high density of measurement points is maintained distributed globally over the face surface. Due to the reduced mesh node density the size of the search segment area corresponds to the low spatial frequencies remaining in the higher level subband.

The search segment is then warped to integrate already known information about its appearance in the next frame (e.g., the effect of head movements). After that correspondence is established using normalised cross-correlation. This yields a motion vector characterising the change of location of the search segment from one frame to the next, which is assigned to the centre node of the search segment. After in a two-step procedure motion vectors for all mesh nodes have been determined the whole mesh is deformed accordingly.

Moving to the next finer tracking level the coordinates of intermediate nodes not tracked yet are interpolated, and then the tracking procedure described above is repeated using the wavelet subbands of a lower scale (higher spatial frequencies). On finishing the run through the updating loop one more time, the entire refining step is once more repeated to yield the final fine-grained result.

In order to allow comparison and analysis of the measured face motion independent from the original video sequence a stabilised version of the mesh must be produced. Therefore the effect of projection is reversed and the translation and rotation of the mesh because of head movements is inverted. Once the tracking of a video sequence is finished the resulting sequence of ellipsoid meshes (one per frame) represents the intrinsic face motion.

The evaluation with several different methods showed that our speech face motion tracking system picked up the essential speech-related facial behaviour and the tracking error remained within acceptable limits so long as the video sequence did not become too long.

5.2 Outlook

Neither the evaluation nor the refinement of the algorithm is finished yet. In that respect the thesis documents work still in progress, even though the fundamental concepts and the layout of the algorithm do not need to be changed anymore. On the evaluation side one of the most interesting studies will be to examine to what extent animations based on the face motion measurements increase intelligibility in speech-in-noise-tasks in the same way as the original video sequence. Note, however, that even this kind of experiment has limited validity concerning the accurateness of the tracking, since the animations would be based on a texture map extracted once for the entire video sequence and thus the animation might be unable to reproduce - just by deforming - certain appearance effects of the facial surface, for instance creases that appear due to face motion (Revèret, Bailly, and Badin, 2000).

On the improvement side video-based head motion tracking should be integrated into the system to make it independent from any data source other than video and allow its use in a wide range of applications as outlined in the introduction. Concerning the tracking algorithm itself one of the most promising extensions would be to include *Kalman filtering* (see for instance Maybeck, 1979; Brown and Hwang, 1997; Grewal and Andrews, 2001). Kalman filtering is a statistical method from modern *control theory* that allows combining one or more

measurements of the same process variable compounded with different degrees of noise with a model prediction, which can include a noise or uncertainty factor as well. Kalman prooved that the method conceived by him to estimate the state of the variable based on the available information leads to the statistically optimal result, i.e. no other method could do better. There are some restrictions on the nature of the noise distribution, but for many, if not most, applications they are acceptable.

In case of the face motion tracking the most difficult part, however, is to develop an appropriate prediction model. In the simplest form this could possibly consist of a smoothed extrapolation using already tracked frames. Essa, Darrell, and Pentland (1994) created a finite-element-based simulation of the muscles and the skin/tissue of the face and used its dynamic properties as a model for face motion from which the prediction was obtained. However, the enormous simplifications made because of the computational costs of such an approach renders in our eyes the otherwise promising approach almost useless, unless only comic-style exaggerated face motion is to be tracked.

Interestingly the authors claim concerning face motion of all kinds, e.g., speech face motion, emotional expression, etc.

> The number of degrees of freedoms required for tracking facial articulations is limited, especially as most of the facial expressions are linear combinations of simpler motions. One can think of tracking being limited to a fixed, relatively small set of "control knobs," one for each type of motion, and then tracking the change in facial expression by moving these control knobs appropriately.

In absence of any proof for the claim we opt decidedly for something quite close to the opposite. Given the complexity of the facial anatomy (and in particular of the insertion points of facial muscles), the prominent importance of the human face in social interactions, the wide variety of facial behaviour, and the distinguished ability of human observers to perceive subtle changes of the face surface, 'simple' seems not a good description for any underlying mechanism. We are aware that Essa et al. (1994) speaks only of face motion *tracking*, but we are just not convinced that the tracking could be done by 'simple' means, if the phenomenon to be tracked is highly complex both in appearance and underlying control mechanisms.

We think that currently our knowledge concerning production and perception of face motion is at a superficial level and that the research into it has still a long way to go before reaching a deeper understanding. As far as speech face motion is concerned this might include better insights into how speech production and perception in general is controlled and processed. We hope that our system will contribute to new findings especially by allowing the analysis of video footage recorded outside the laboratory.

Bibliography

F. Auger, P. Flandrin, P. Gonçalvès, and O. Lemoine. *Time-Frequency Toolbox Tutorial*. CNRS (France) and Rice University (USA), July 1997. 28

J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12:1:43–77, 1994. 13

S. Basu, I. Essa, and A. Pentland. Motion regularization for model-based head tracking. In *Conference on Pattern Recognition (ICPR '96)*, 1996. 14

M. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. *Int. Journal of Computer Vision*, 25(1):23–48, 1997. 14

J. Bortz. *Statistik. Für Sozialwissenschaftler*. Springer, Berlin, Heidelberg, New York, 4th edition, 1993. 40

I. N. Bronstein, K. A. Semendjajew, G. Musiol, and H. Mühlig. *Taschenbuch der Mathematik*. Verlag Harri Deutsch, Frankfurt am Main, fourth edition, 1999. 54

R. G. Brown and P. Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley & Sons, New York, 1997. 112

B. Burke Hubbard. *The World According to Wavelets*. A K Peters, Natick, Massachusetts, 1998. 20, 27

C. S. Burrus, R. A. Gopinath, and H. Guo. *Introducton to Wavelets and Wavelet Transforms*. Prentice Hall, Upper Saddle River, New Jersey, 1998. 27, 36

D. Callan, A. Callan, and E. Vatikiotis-Bateson. Neural areas underlying the processing of visual speech information under conditions of degraded auditory information. In *International Conference on Auditory-Visual Speech Processing (AVSP 2001)*, pages 45–49, Aarlborg/Denmark, 2001. 3

J. Carter, C. Shadle, and C. Davies. On the use of structured light in speech research. In *ETRW - 4th Speech Prod. Seminar*, pages 229–232, Autrans, May 1996. 4

J. Cole. *About Face*. MIT Press, Cambridge, Massachusetts, 1998. v

I. Daubechies. *Ten Lectures on Wavelets*. SIAM, Philadelphia, Pennsylvania, 1992. 26, 30, 31, 32, 33, 38

H. Davis and S. R. Silverman, editors. *Hearing and Deafness*. Holt, Rinehart, & Winston, New York, 3rd edition, 1970. 98

D. DeLillo. *White Noise*. Viking Penguin, New York, 1984.

B. A. Draper and J. R. Beveridge. Affine and perspective projection, 2002. URL `http://www.dai.ed.ac.uk/CVonline/`. 77

P. Ekman and W. V. Friesen. *The facial action coding system (FACS): A technique*

for the measurement of facial action. Consulting Psychologists Press, Palo Alto, CA, 1978. 5

P. Ekman and E. Rosenberg, editors. *What the face reveals: basic and applied studies of spontaneous expression using the facial action coding system (FACS)*. Oxford University Press, Oxford, 1997. 6

N. J. Emery. The eyes have it: the neuroethology, function and evolution of social gaze. *Neuroscience and Biobehavioral Reviews*, 24:581–604, 2000. v, 64

I. Essa, T. Darrell, and A. Pentland. Tracking facial motion. In *Proceedings of IEEE Nonrigid and Articulated Motion Workshop 1994*, Austin, Texas, November 1994. 113

I. Essa and A. Pentland. Coding, analysis, interpretation and recognition of facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19 (7):757–763, July 1997. 14

O. Faugeras. *Three-Dimensional Computer Vision. A Geometric Viewpoint*. MIT Press, Cambridge, Massachusetts and London, England, 1993. 17, 60

A. Fitzgibbon, M. Pilu, and R. Fisher. Direct least-square fitting of ellipses. In *International Conference on Pattern Recognition*, Vienna, August 1996. 51, 53

A. W. Fitzgibbon, M. Pilu, and R. B. Fisher. Direct least square fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):476–480, 1999. 52, 53

D. A. Forsyth. *Computer Vision - A Modern Approach*. Prentice Hall, Upper Saddle River, New Jersey, 2003. 15, 17, 60

W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Trans. Patt. Anal. Mach. Intell.*, 13:891–906, September 1991. 8

R. Goecke, B. Millar, A. Zelinsky, and J. Robert-Ribes. Analysis of audio-video correlation in vowels in Australian English. In *International Conference on Auditory-Visual Speech Processing (AVSP 2001)*, pages 115–119, Aarlborg/Denmark, 2001.

J. Gold, P. Bennett, and A. Sekuler. Identification of band-pass filtered letters and faces by human and ideal observers. *Vision Research*, 39:3537–3560, 1999. 67

J. Gomes, L. Darsa, B. Costa, and L. Velho. *Warping and Morphing of Graphical Objects*. Morgan Kaufmann, San Francisco,California, 1999. 77, 80, 83

R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, Upper Saddle River, New Jersey, 2002. 76

M. S. Grewal and A. P. Andrews. *Kalman Filtering: Theory and Practice*. John Wiley & Sons, New York, 2001. 112

P. Haberäcker. *Praxis der Digitalen Bildverarbeitung*. Carl Hanser Verlag, München/Wien, 1995. 71

R. Halir and J. Flusser. Numerically stable direct least squares fitting of ellipses, 2000. URL citeseer.nj.nec.com/350661.html. 53

M. Heller and V. Haynal. Depression and suicide faces. In P. Ekman and E. Rosenberg, editors, *What the face reveals: basic and applied studies of spontaneous expression using the facial action coding system (FACS)*. Oxford University Press, Oxford, 1997. 5

J. E. Jackson. *A user's guide to principal components*. John Wiley & Sons, New York, 1991. 38, 39, 40

B. Jawerth and W. Sweldens. An overview of wavelet based mul-

tiresolution analyses. Technical report, Februar 1993. URL citeseer.nj.nec.com/article/jawerth94overview.html. 27

J. Jiang, A. Alwan, L. E. Bernstein, P. Keating, and E. Auer. On the correlation between facial movements, tongue movements and speech acoustics. In *International Conference on Spoken Language Processing*, volume 1, pages 42–45, Bejing, China, 2000. 3

S. X. Ju, M. J. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated image motion. In *International Conference on Automatic Face and Gesture Recognition*, pages 38–44, Killington, Vermont, 1996. 13

G. Kaiser. *A friendly guide to wavelets*. Birkhäuser, Boston, 1994. 27

C. Kroos, T. Kuratate, and E. Vatikiotis-Bateson. Loosing track? Video-based measurement of face motion. In *HIP99-36*, pages 19–24, Okinawa, Japan, 2000. IEICE. 54

C. Kroos, T. Kuratate, and E. Vatikiotis-Bateson. Video-based face motion measurement. *Journal of Phonetics (special issue)*, 30:569–590, 2002.

C. Kroos, S. Masuda, T. Kuratate, and E. Vatikiotis-Bateson. Towards the 'face-coder': Dynamic face synthesis based on image motion estimation in speech. In *International Conference on Auditory-Visual Speech Processing (AVSP 2001)*, pages 24–29, Aarlborg/Denmark, 2001. 99

T. Kuratate, H. Yehia, and E. Vatikiotis-Bateson. Kinematics-based synthesis of realistic talking faces. In *D. Burnham, J. Robert-Ribes, and E. Vatikiotis- Bateson (Ed.), International Conference on Auditory-Visual Speech Processing (AVSP'98)*, pages 185–190, Terrigal-Sydney, Australia, 1998. Causal Productions. 6

M. La Cascia, J. Isidoro, and S. Sclaroff. Head tracking via robust registration in texture map images. In *IEEE Conf. on Computer Vision and Pattern Recognition*, Santa Barbara, CA (USA), 1998. 14

J. Lien, T. Kanade, J. Cohn, and C. Li. Detection, tracking, and classification of action units in facial expression. *Journal of Robotics and Autonomous Systems*, 31 (3):131–146, July 1999. 14

J. MacDonald, S. Andersen, and T. Bachmann. Hearing by eye: How much spatial degradation can be tolerated? *Perception*, 29:1155–1168, 2000. 109

S. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11:674–693, 1989. 32

D. Marr. *Vision*. Freeman, New York, 1982. 11, 67

K. Mase. Recognition of facial expression from optical flow. *IEICE Transactions*, E74:3474–3483, 1991. 14

K. Mase and A. Pentland. Automatic lipreading by optical-flow analysis. *Systems and Computers in Japan*, 22:67–76, 1991. 1

P. S. Maybeck. *Stochastic Models, Estimation, and Control*, volume I. Academic Press, New York, 1979. 112

H. McGurk and J. MacDonald. Hearing lips and seeing voices. *Nature*, 264:746–748, 1976. 2, 3

K. G. Munhall, C. Kroos, and E. Vatikiotis-Bateson. Audiovisual perception of band-pass filtered faces. In *Autumn Meeting of the Acoustical Society of Japan*, pages 519–520, Oita, 2001a. 71

K. G. Munhall, C. Kroos, and E. Vatikiotis-Bateson. Band-pass filtered faces and audiovisual speech perception. *Journal of the Acoustical Society of America*, 109

(Suppl.1):2314, 2001b. 71

K. G. Munhall, C. Kroos, and E. Vatikiotis-Bateson. Spatial frequency requirements for audiovisual speech perception. *Perception & Psychophysics*, in press. 71, 108, 109

R. Näsänen. Spatial frequency bandwidth used in the recognition of facial images. *Vision Research*, 39:3824–3833, 1999. 67

Digital Imaging and Communications in Medicine (DICOM). National Electrical Manufacturers Association, PS 3 edition, 2003. URL http://medical.nema.org/dicom/2003.html. 45

A. Nefian, M. Khosravi, and M. Hayes. Realtime detection of human faces in uncontrolled environments. In *Proceedings of SPIE conference on Visual Communications and Image Processing, Vol. 3024*, pages 211–219, San Jose, California, USA, 1997. 50

J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965. 54

Y. Nievergelt. *Wavelets Made Easy*. Birkhäuser, Boston, Basel, Berlin, 1999. 27

S. E. Palmer. *Vision Science - Photons to Phenomenology*. MIT Press, Cambridge, Massachusetts, 1999. 11, 67

F. I. Parke and K. Waters. *Computer Facial Animation*. A K Peters, Wellesley, Massachusetts, 1996. 73

L. Prasad and S. S. Iyengar. *Wavelet analysis with applications to image processing*. CRC Press LLC, Boca Raton, 1997. 27

J. G. Proakis and D. G. Manolakis. *Digital Signal Processing*. Prentice Hall, Upper Saddle River, New Jersey, 1996. 23

M. R. Raghuveer and A. S. Bopardikar. *Wavelet transforms: introduction to theory and applications*. Addison Wesley, Reading, 1998. 27, 30, 31, 32, 33, 35

L. Revèret, G. Bailly, and P. Badin. MOTHER: a new generation of talking heads providing a flexible articulatory control for video-realistic speech animation. In *International Conference on Speech and Language Processing*, Beijing, China, October 2000. 112

L. Revèret, F. Garcia, C. Benoit, and E. Vatikiotis-Bateson. An hybrid approach to orientation-free liptracking. In *Proc. of the First ESCA Workshop on Audio-Visual Speech Processing, AVSP'97*, pages 117–120, Rhodes, Greece, September 1997. 14

P. L. Rosin. Further five-point fit ellipse fitting. *Graphical Models and Image Processing*, 61(5):245–259, 1999. 53

G. S. Sánchez, N. G. Prelic, and S. J. G. Galán. *Uvi_Wave. Wavelet Toolbox for use with Matlab*. Departamento de Tecnoloxías das Comunicacións. Universidade de Vigo, Vigo, second edition, July 1996. 38, 69

A. Schödl, A. Haro, and I. A. Essa. Head tracking using a textured polygonal model. Technical report, GIT-GVU-98-24, 1998. 14

L. G. Shapiro and G. C. Stockman. *Computer Vision*. Prentice Hall, Upper Saddle River, New Jersey, 2001. 17, 25, 56, 57, 60

R. H. Stetson. *Motor phonetics a study of speech movements in action*. North Holland Publishing, Amsterdam, 2nd edition, 1951. 2

C. Stiller and J. Konrad. Estimating motion in image sequences. *Signal Processing Magazine*, 16 (4):70–91, July 1999. 13

G. Strang and T. Nguyen. *Wavelets and Filter Banks*. Wellesley-Cambridge-Press, Wellesley, Massachusetts, 1997. 27, 31, 34, 36

W. Sumby and I. Pollack. Visual contribution to speech intelligibility in noise. *Journal of the Acoustical Society of America*, 26:212–215, 1954. 3

E. Trucco and A. Verri. *Introductory techniques for 3-D computer vision*. Prentice Hall, Upper Saddle River, New Jersey, 1998. 21, 51, 52, 60, 84

A. Ude. Robust human motion estimation using detailed shape and texture models. *Journal of the Robotics Society of Japan*, 19(5):15–18, July 2001. 13

P. P. Vaidyanathan. *Multirate systems and filter banks*. Prentice Hall, 1993. 32, 69

E. Vatikiotis-Bateson, I.-M. Eigsti, S. Yano, and K. G. Munhall. Eye movement of perceivers during audiovisual speech perception. *Perception and Psychophysics*, 60:926–940, 1998. 67

B. A. Wandell. *Foundations of Vision*. Sinauer Associates, Sunderland, Massachusetts, 1995. 11, 65

E. W. Weisstein. Nyquist frequency. Eric Weisstein's world of mathematics, 1999. URL http://mathworld.wolfram.com/NyquistFrequency.html. 21, 33

Y.-T. Wu, T. Kanade, J. Cohn, and C. Li. Optical flow estimation using wavelet motion model. In *International Conference on Computer Vision*, pages 992–998, Bombay, India, January 1998. 14, 70, 98

H. C. Yehia, P. E. Rubin, and E. Vatikiotis-Bateson. Quantitative association of vocal-tract and facial behavior. *Speech Communication*, 26:23–44, 1998. 3, 6, 8, 103

V. M. Zatsiorsky. *Kinematics of human motion*. Human Kinetics, 1998. 48

W. D. Zimmer and E. Bonz. *Objektorientierte Bildverarbeitung*. Carl Hanser Verlag, München/Wien, 1995. 47

Appendix A

Deutsche Zusammenfassung

A.1 Einführung

Im Bereich der Analyse von Gesichtsmimik steht traditionell die Analyse *emotionalen* Gesichtsausdrucks stark im Vordergrund. Zwar ist inzwischen das Interesse an der genuinen Sprechmimik stark angestiegen dank vielfältiger Anwendungsmöglichkeiten (computergesteuerte Animationen von sprechenden Gesichtern halten immer mehr Einzug in Werbung, Unterhaltung und als Benutzerverkörperung in virtuellen Räumen), jedoch ging das nicht immer einher mit einer Nutzung des unter anderem in der Phonetik vorhanden Wissens über die beim Sprechen ablaufenden Vorgänge. Beispiele aus Veröffentlichungen im Bereich *Maschinelles Sehen* belegen dies. Auf der andern Seite wurde und wird die menschliche *auditorisch-visuelle* Sprachverarbeitung in der Phonetik zum Teil immer noch nur als bloßes Kuriosum angesehen.

Die vorliegende Doktorarbeit präsentiert ein System, das die Analyse von (Sprech-)Mimik erlauben soll, ohne dass auf sensorbasierte Verfolgungssysteme zurückgegriffen werden muss. Diese haben zwar den Vorteil von großer Genauigkeit und zeitlich und räumlich guter Auflösung, aber den Nachteil, dass sie verhältnismäßig teuer sind und im Allgemeinen nicht außerhalb des Labors eingesetzt werden können. Letzteres verhindert kombiniert mit der Erfordernis, Sensoren an der Versuchsperson anbringen zu müssen, die Analyse spontaneren Sprechverhaltens.

Entsprechend wäre ein Verfahren, das Messdaten der Gesichtsmimik aus Standardvideosequenzen extrahieren kann, höchst wünschenswert. Dem stehen jedoch die eminenten Probleme im Bereich der Bildbewegungsschätzung (rechnerische Ermittlung der Bewegung von Bildobjekten in Bildsequenzen, die realen Objekten entsprechen mögen oder auch nicht) entgegen, ganz besonders, da es sich beim menschliche Gesicht um einen höchst komplexen Untersuchungsgegenstand handelt.

Im Bildakquisitionsprozess werden kontinuierliche dreidimensionale Trajektorien der realen Szene nicht-linear auf zweidimensionale zeit-diskrete Trajektorien abgebildet. Die erhaltenen Bildintensitätswerte aber sind eine Kombination aus den Reflektionscharakteristika der abgebildeten Oberflächen, dem einzigen Objektmerkmal, auf dem die Bewegungsverfolgung aufbauen kann, und anderen Einflussfaktoren wie der allgemeinen Szenenbeleuchtung.

Zudem besteht die Gesichtsmimik aus der eigentlichen oder *intrinsischen* Mimik und der überlagerten Bewegung des gesamten Kopfes. Während letztere

in die Kategorie der Bewegung von starren Körpern fällt und deshalb in einfacher Weise mit drei Translations- und drei Rotationsparametern parametrisiert werden kann, lässt sich erstere nur als nicht-starre Bewegung beschreiben mit im Prinzip unendlich vielen Freiheitsgraden.

Die genannten Probleme lassen sich jedoch bis zu einem gewissen Grad lösen durch Rückgriff auf Einschränkungen, denen die Gesichtsoberfläche unterliegt, z.B. dass sie mit Ausnahme des Mundes eine zusammenhängende Fläche bildet, von der sich nicht einfach Teile ablösen können.

A.2 Theoretische und empirische Grundlagen

Das Herzstück des in der Doktorarbeit beschriebenen Systems ist ein Algorithmus zur videobasierten Bewegungsverfolgung von Gesichtsmimik. Aufgrund der oben beschriebenen inhärenten Schwierigkeiten musste ein neuer Ansatz zur Lösung gefunden werden. Dieser benutzt aber natürlich zum Teil bereits bekannte Techniken. Besondere Bedeutung haben dabei:

i. Bildregistierung
Von den realen Objekten, die in der Bildsequenz verfolgt werden sollen, wird ein Erscheinungsmodell erstellt, gewöhnlich aus einer geometrischen Beschreibung und einer Oberflächentextur bestehend. Die Objektmodelle werden entsprechend der zu erwartenden Bewegung animiert und die so generierten Bilder mit denen der realen Videosequenz mit Hilfe eines geeigneten Abstandsmaßes verglichen. Die Bewegungsparameter, die ein Minimum im Abstand des jeweiligen realen und generierten Bildes erreichen, werden dem verfolgten Objekt zugewiesen. Um nicht einen möglicherweise hochdimensionalen Suchraum komplett durchsuchen zu müssen, wird üblicherweise ein Optimierungsverfahren verwendet.

ii. Perspektivische Projektion
Perspektivische Projektion erlaubt den Vorgang der Abbildung der realen dreidimensionalen Szene auf das zweidimensionale Bild vereinfacht mathematisch zu beschreiben. In dieser Weise kann ein Kameramodell erstellt werden. Ein sehr einfaches aber durchaus für viele Anwendung ausreichendes derartiges Modell ist ein idealisiertes Modell der *camera obscura*.

iii. Wavelets in Verbindung mit einer Vielfachauflösung-Analyse
Ortsfrequenzen können im großen Ganzen in der selben Weise behandelt werden wie Frequenzen im zeitlichen Bereich. Bewährte Verfahren wie die *Fouriertransformation* können ohne Schwierigkeiten auch auf zweidimensionale Signale erweitert werden, wobei nun aber neben Amplitude und Phase der Frequenzkomponenten auch deren Orientierung ein Rolle spielt.

Die diskrete *Wavelet-Transformation* ist im Gegensatz zur Fouriertransformation zeitlich/räumlich *und* frequenzmäßig gut lokalisiert und zwar in einer sinnvollen Abhängigkeit vom Frequenzbereich. Dabei wird das Signal nicht wie bei der Fouriertransformation in Sinus- und Kosinuskomponenten zerlegt, vielmehr sind die Bausteine 'wavelets', 'kleine Wellen', Funktionen, deren Energie finit ist, die also außerhalb eines kurzen Intervalls meist nur Funktionswerte gleich Null aufweisen.

Der mathematisch geklärte Zusammenhang zwischen Wavelet-Transformation und einer strikt formulierten Vielfachauflösung-Analyse

erlaubt es, die zweidimensionale Wavelet-Transformation als kaskadische Filterbank mit Halbbandfiltern zu realisieren. Dabei entstehen bei jedem Filterungsschritt (der jeweils einer Wavelet-Ebene entspricht) vier Signale: eine Approximation und drei orientierte Seitenbänder (horizontal, vertikal und diagonal.

A.3 Videobasierte Messung der Gesichtsmimik

Der Algorithmus umfasst zwei zentrale Module: die Initialisierung und die eigentliche auf Einzelbildern operierende Bewegungsverfolgung.

A.3.1 Initialisierung

In dieser Phase wird ein parametrisiertes ellipsoidales Maschennetzwerk generiert, auf die Größe des Gesichtes der Versuchsperson in einem vom Benutzer ausgewählten Referenzvideobild skaliert und über dem Gesicht plaziert, so dass es dieses vollständig abdeckt. Das Maschennetzwerk liefert in der nachfolgenden Bewegungsmessung Ankerpunkte für die Verfolgung der Mimik und dient der Erfassung von Ortsveränderungen von kleinen über die Netzknoten definierten Segmenten der Gesichtsoberfläche. Um die Größenanpassung und Positionierung zu erreichen, wird der Benutzer aufgefordert, die Gesichtsaußenkontur und die Augenwinkel der Versuchperson im Bild mit eine paar Punkten zu markieren. Anschließend wird eine Ellipse berechnet, so dass sie in ihrer Größe optimal auf die Konturpunkte angepasst ist, ihre Orientierung aber durch den Neigungswinkel der Geraden zwischen den Augenwinkelpunkten fixiert ist. Die Ellipse liefert die Parameter zur Generierung des Ellipsoids. Außerdem wird in der Initialisierungphase ein einfaches Kameramodell erstellt und, wenn möglich, kalibriert.

Die Kopfbewegungen der Versuchsperson werden in der jetzigen Implementierung noch nicht videobasiert verfolgt, denn eine der zentralen Fragen in der Entwicklung des Verfahren war, welche Präzision in der Messung der intrinsischen Mimik zu erreichen ist, vorausgesetzt die Kopfbewegungserfassung liefert nahezu perfekte Resultate. Demzufolge liest die Initialisierungroutine Kopfbewegungsdaten ein, die mit Hilfe eines sensorbasierten kommerziellen Gerätes und einer Kopfhalterung für die Sensoren erhoben wurden.

A.3.2 Bewegungsmessung

Die Bewegungsverfolgung verwendet in zweifacher Hinsicht eine Vielfachauflösung-Analyse, zum einen im strikten Sinne für die Bilddaten, zum anderen weniger rigoros formuliert für das Maschennetzwerk, in dem seine Netzknotendichte variiert wird. Im Bildverarbeitungsbereich ist das Ziel bandbegrenzte Signale zu erhalten und die Vielfachauflösung-Analyse wird durch eine Wavelet-Transformation realisiert. Mit der Vielfachauflösung-Analyse das Netzwerk betreffend wird eine Grob-zu-fein-Strategie umgesetzt, die es möglich macht, erst größere Bewegungen mit einem groben Netzwerk zu verfolgen und die Ergebnisse dann schrittweise mit feineren Netzwerken zu präzisieren.

Das ellipsoidale Maschennetzwerk wird mit Hilfe des Kameramodells direkt den Seitenbändern aus der Wavelet-Filterung überlagert. Dabei folgt es - gesteuert von den externen Kopfbewegungsdaten - den Kopfbewegungen der Versuchsperson, so dass das Netzwerk als Ganzes immer über derselben Stelle des Gesichtes verharrt.

Suchsegmente werden über benachbarte Knotenpunkte, die einen beliebigen zentralen Knoten umgeben, definiert. Das Suchsegement registriert Intensitätswerte an seinen Pixelkoordinaten im ersten von zwei aufeinanderfolgenden Videobildern. Um Informationen zu berücksichtigen, die über seine Form und Position im nächsten Videobild schon vorliegen (aufgrund der Kopfbewegungsdaten oder der schon erfolgten Bewegungsmessung auf einer gröberen Ebene), wird es entsprechend verzerrt.

Dann wird mit Hilfe der Kreuzkorrelationsmethode seine korrespondierende Position im zweiten Videobild ermittelt. Der entsprechende Verschiebungsvektor wird dem zentralen Knoten des Suchsegments als Bewegungsvektor zugewiesen. Ist dies für alle Knotenpunkte erfolgt, wird das gesamte Netzwerk entsprechend verformt. Solange noch nicht die feinste ausgewählte Netzwerkdichte erreicht ist, wird der Vorgang auf der nächst feineren Ebene wiederholt, wobei die hinzugekommenen Knoten bilinear interpoliert werden, um Startwerte für die Korrespondenzermittlung zu erhalten.

Um stabilisierte, von der speziellen Videosituation unabhängige Messungen als Ausgabe zu erhalten, werden am Schluss Kopfbewegungen ausgeglichen und die perspektivische Projektion reversiert. Das Ergebnis ist eine Sequenz von stabilisierten, verformten Netzwerken, eines pro Videobild, die die in der Videosequenz sichtbare Gesichtsmimik repräsentieren.

A.4 Validierung

Eine Reihe von unterschiedlichen Evaluierungsmethoden, sowohl qualitativer wie quantitativer Natur, hat gezeigt, dass der vorgestellte Algorithmus die wesentlichen Charakteristiken der Sprechmimik erfasst und der Messfehler im akzeptablen Rahmen bleibt, solange die analysierte Videosequenz nicht zu lang ist.

Appendix B

Lebenslauf

Christian Kroos

23. März 1964
Geboren in München (Bayern) als Sohn von Liselotte Kroos und Friedrich-Karl Kroos.

1970-74
Grundschule in Starnberg (Bayern).

1974-1983
Gymnasium in Starnberg.

29. Juni 1983
Abitur.

1983-1988
Studium der Kommunikationswissenschaft und Philosophie an der Ludwig-Maximilian-Universität München.

1989-1990
Zivildienst im Krankenhaus 'München-Schwabing'.

1991-1997
Studium der Phonetik (Professor Tillmann), Logik und Theaterwissenschaft an der Ludwig-Maximilian-Universität München. 1992 Mitarbeit als studentische Hilfskraft im PhonDat-Projekt, von 1992 bis 1996 im DFG Projekt 'Das Vokalsystem des Deutschen - kinematische Analyse' (Supervisor: Dr. Philip Hoole)

20. Februar 1997
Magister artium (M.A.) in Phonetik, Logik und Theaterwissenschaft.
Titel der Magisterarbeit: 'Eingipflige und zweigipflige Vokale des Deutschen? Kinematische Analyse der Gespanntheitsopposition im Standarddeutschen'.

1997-1999
Wissenschaftlicher Mitarbeiter am Institut für Phonetik, München, im DFG Projekt 'Compensatory articulation and the nature of phonetic goals'.
Supervisor: Dr. Philip Hoole.

März 1999 - März 2002
Gastwissenschaftler am ATR-International (Kyoto/Japan). Entwicklung eines Algorithmus zur videobasierten Messung der Gesichtsmimik (Patent für Japan und anhängig für USA).
Supervisor: Dr. Eric Vatikiotis-Bateson.

Seit April 2000
Verfassung der Promotionsschrift *'A system for video-based analysis of face motion during speech'*. Doktoranden-Stipendium der DFG im Rahmen des Graduiertenkollegs 'Sprache, Mimik und Gestik im Kontext technischer Informationssysteme'.