Reconstructing Human Motion

Kumulative Dissertation

Zur Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Dipl.-Inform. Jan Baumann

Bonn
2017

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn.

Promotionskommission:

- Erstgutachter: Prof. Dr. Andreas Weber
- Zweitgutachter: Prof. Dr. Bernd Eberhardt
- Fachnahes Mitglied: Prof. Dr. Thomas Schultz
- Fachfremdes Mitglied: Priv.-Doz. Dr. med. Rainer Surges

Tag der Promotion: 05.12.2017

Erscheinungsjahr: 2018

Danksagung

An dieser Stelle möchte ich mich bei einigen Menschen herzlich bedanken:
Andreas Weber, für die angenehme und verständnisvolle Betreuung,
Rainer Surges, für die gute Zusammenarbeit und die interessanten Einblicke in die Epilepsieforschung an der Bonner Klinik für Epileptologie,
Bernd Eberhardt dafür, dass ich mich an der HdM Stuttgart immer willkommen und umsorgt gefühlt habe,
Arno Zinke und Jürgen Gall für das außergewöhnliche Sachverständnis und die wertvollen Tipps,
Dirk Schulz, für die Betreuung am Fraunhofer FKIE,
meinen Kollegen Amirhossein Jahanbekam, Björn Krüger, Jochen Tautges, Tomas Lay Herrera und Raoul Wessel.
Für die außerfachliche Unterstützung danke ich meiner lieben Frau Sonja, meinen Mäusen Leon, Noah und Kilian, meiner Mutter, meinen Schwiegereltern und nicht zuletzt meinem guten Freund Michael Knümann.

Abstract

This thesis presents methods for reconstructing human motion in a variety of applications and begins with an introduction to the general motion capture hardware and processing pipeline.

Then, a data-driven method for the completion of corrupted marker-based motion capture data is presented. The approach is especially suitable for challenging cases, e.g., if complete marker sets of multiple body parts are missing over a long period of time. Using a large motion capture database and without the need for extensive preprocessing the method is able to fix missing markers across different actors and motion styles. The approach can be used for incrementally increasing prior-databases, as the underlying search technique for similar motions scales well to huge databases.

The resulting clean motion database could then be used in the next application: a generic data-driven method for recognizing human full body actions from live motion capture data originating from various sources. The method queries an annotated motion capture database for similar motion segments, able to handle temporal deviations from the original motion. The approach is online-capable, works in realtime, requires virtually no preprocessing and is shown to work with a variety of feature sets extracted from input data including positional data, sparse accelerometer signals, skeletons extracted from depth sensors and even video data. Evaluation is done by comparing against a frame-based *Support Vector Machine* approach on a freely available motion database as well as a database containing Judo referee signal motions.

In the last part, a method to indirectly reconstruct the effects of the human heart's pumping motion from video data of the face is applied in the context of epileptic seizures. These episodes usually feature interesting heart rate patterns like a significant increase at seizure start as well as seizure-type dependent drop-offs near the end. The pulse detection method is evaluated for applicability regarding seizure detection in a multitude of scenarios, ranging from videos recorded in a controlled clinical environment to patient supplied videos of seizures filmed with smartphones.

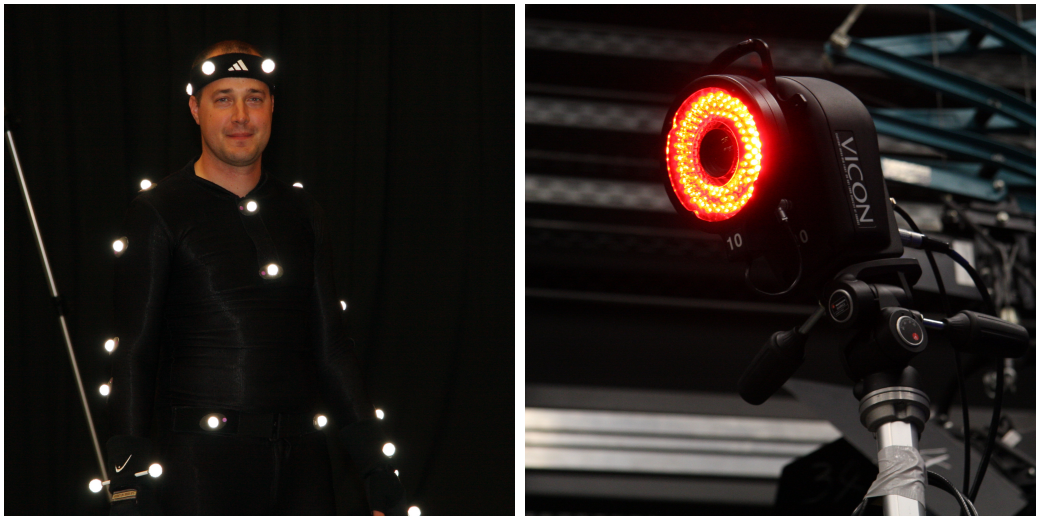
Contents

1 Introduction

Motion capture (mocap) is the process of recording the body and/or facial motion of an actor for further processing using a computer. Motions captured by motion capture systems have found their way into many different industries. Computer animated movies and games might be the first applications that come to mind, but motion capture system are used in our everyday lives as well, e.g., in consumer electronics gaming devices like the Microsoft Kinect. In medical rehabilitation, such systems help people with injuries of the musculoskeletal system and even stroke patients regain their previous abilities of movement and thereby help to increase quality of life itself. Other uses include ergonomics research, biomechanics, sports analysis, robotics and military applications.

The utilized sensor systems range from a single noisy accelerometer, which can be found in almost every smartphone manufactured today, over more complex and sophisticated devices like orientation sensors, wearable exoskeletons, video and depth cameras to high quality and costly optical motion capture systems which can deliver sub-millimeter accuracy at very high frame rates. The individual systems differ in many ways and the type chosen for a specific application depends on several factors, including acquisition and maintenance cost, capture accuracy, ease of use, ability for outdoor recordings and many more.

This work primarily concentrates on passive optical mocap, which is used in applications that need mocap data of high spatial as well as high temporal resolution. In this type of mocap, retro-reflective markers are attached to specific locations on an actor's body who wears a purpose made, tightly fitting motion capture suit (see Fig. 1.1a). The actor's motion performance is simultaneously filmed by multiple



(a) Passive optical markers attached to the author's body. The photo was taken using a flash to emphasize the reflective nature of the markers.

(b) Photo of a Vicon motion capture camera with its outer ring of scene illuminating LEDs.

Figure 1.1: Passive optical motion capture

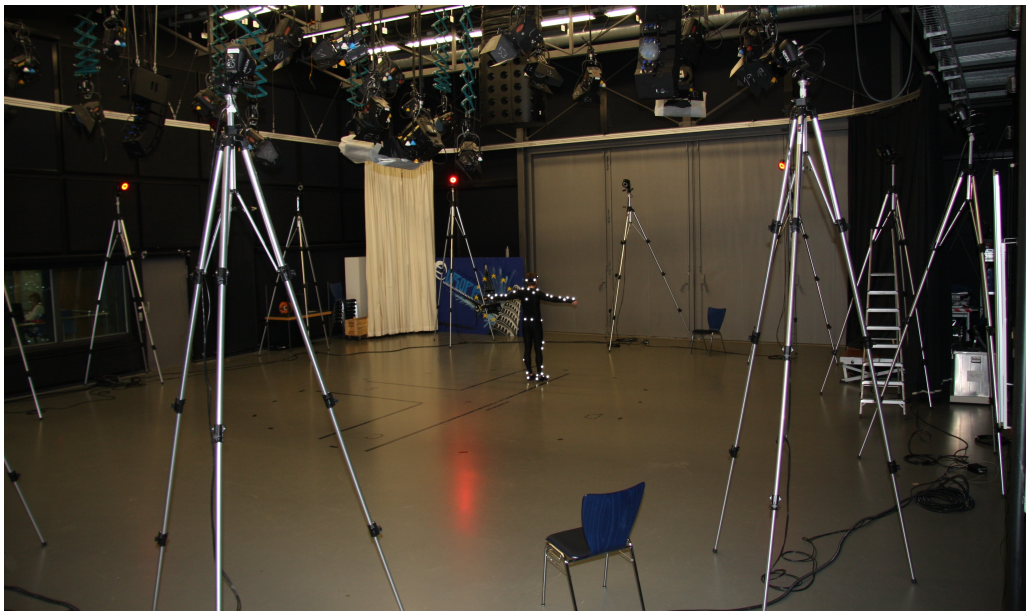


Figure 1.2: Example of a 12 camera motion capture setup using a Vicon system. Cameras are placed in a circular pattern around the center point of the capture volume, whose circular outline is marked as a reference on the floor using tape strips.

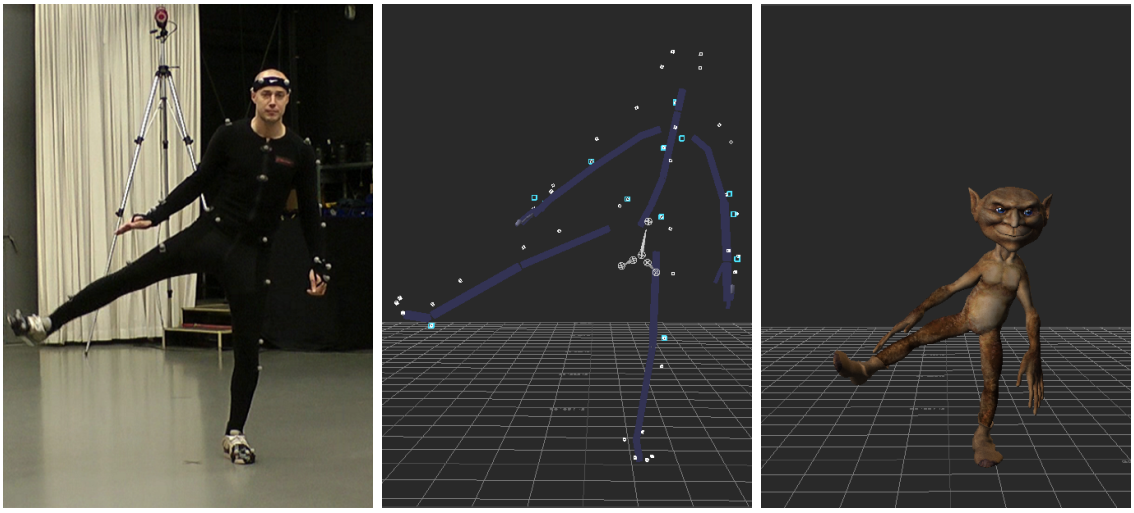


Figure 1.3: Typical passive optical motion capture workflow.
Left: The actor is recorded wearing a set of retroreflective markers.
Center: A skeleton abstraction is computed from the marker data.
Right: The animation data is used to animate a retargeted, virtual character.

scene illuminating infrared cameras, strategically placed in such a way that every part of the capture volume, i.e., the area where the actor is allowed to perform (see Fig. 1.2), is seen by at least three cameras to allow triangulation of each marker's position. Unlike active optical mocap systems, where each marker flashes in a distinct frequency and thus making it relatively easy to identify and label, passive optical systems need to employ sophisticated labeling and tracking algorithms that merge all of the 2D camera inputs to calculate and label the individual marker trajectories. Problems arise when markers are occluded from view for a prolonged time, e.g., the innermost markers of two actors in a close dance. Also, most markers are commonly attached to the suit using hook and loop fasteners, which means they can fly off due to the centrifugal forces generated by highly dynamic performances or even simply because the fastener is worn out.

After recording an actor's motion performance, a skeleton abstraction can be computed from the marker data. In practice, the resulting data is often applied to an animated character with the same skeleton, but different proportions (see Fig. 1.3). In this case, a technique called *retargeting* [Gle98] can be applied to ensure that the resulting animation satisfies various constraints like proper ground contacts (i.e., feet not above or below the ground plane) and step sizes (i.e., no feet

skating artefacts that would result from simply scaling the original skeleton).

Not every new motion has to be recorded with a mocap system. In their classic paper *Motion Graphs* [KGP02], Kovar et al. present a technique to synthesize new motions from a database of existing motions. Here, a directed graph is used where database motions are represented as nodes and possible transitions between these motions as edges. Now, new motions can be generated by simply building walks over the graph.

Regardless of the chosen motion capture system, challenging capture situations rarely produce satisfactory results. The raw recordings often contain missing parts that need to be completed with sophisticated techniques in a post-processing step. A data-driven method for these situations is presented in Chapter 2.

Activity recognition, i.e., the detection of known human actions from motion data, has become very popular in recent years. Applications include fitness tracking software (e.g., how many push-ups, pull-ups or squats were performed in a training session), controlling a game character with the player's actions as well as smart robots, that act according to what a nearby human is currently doing. Chapter 3 presents a technique for action recognition capable of using any combination of sensors. The method is applied to sensor inputs ranging from simple accelerometers to high quality optical motion capture data.

When there is no actual physical sensor attached to an actor's body, vision based motion capture techniques can be applied to video recordings. This can even be taken one step further by using algorithms that indirectly capture the motion of a subject's heart and chest, i.e., heart and respiration rates, and thus enable monitoring of direct and derived body vital signs. In Chapter 4, a video pulse detection technique is evaluated on patients with epilepsy, where distinct heart rate patterns can be observed during seizures.

2

Completion of Motion Capture Data

This chapter presents an extended version of the publication:
Data-Driven Completion of Motion Capture Data [BKZW11].

2.1 Introduction

Optical motion capture is the standard technique for creating realistic human motions in computer animation: Multiple cameras are used to record and track markers that are attached to an actor's body (see Chapter 1). Finally, 3D trajectories of the individual markers are reconstructed from the two dimensional images by triangulation. Using fitting techniques, skeleton abstractions may be computed.

During motion capture, markers are often occluded from the view of too many or even all cameras, resulting in a gap in the final 3D marker trajectory. This is usually unproblematic for small gaps, because these can be filled by a cubic spline interpolation. In larger gaps, when only a single marker of a body segment is missing, the rigid body relationship of the segment's other markers can be used for trajectory reconstruction.

If gaps in several markers occur for a long time period (e.g., several seconds)—a scenario quite common if closely interacting actors are captured simultaneously or interaction with the environment is essential, sophisticated methods for the completion of the maker trajectories need to be employed.

Although the topic of cleaning motion capture data is a classical one and various techniques are available in commercial mocap system software, the problem is far from being solved and has obtained renewed attention in the last years

[LMPF10, LC10, XFH11, FXZ+14, XSZF16]. Unfortunately, the majority of the existing approaches have major limitations, especially if no previously captured motions of *the same actor* which are similar to the one to be cleaned are available. Many approaches are also limited by the size of motion database that can be handled (e.g., Grochow et al. [GMHP04]).

This chapter presents a general framework for data-driven completion of gaps in marker-based mocap data. The novel approach can handle challenging cases, especially if complete marker sets of multiple body parts are missing over a long period of time. Without the need for extensive preprocessing the framework is able to fix missing markers across different actors and motion styles. The results agree with human intuition and key features of the original input motion are greatly retained.

2.2 Related Work

Rudimentary gap filling is available in commercial software systems like Vicon IQ or Blade [Vic]. These methods rely on simple techniques, such as linear or spline interpolation of marker trajectories and thus fail if curvatures change sign. Such simple methods do not account for a correlated motion of markers that occurs when markers are attached to the same body segment. This marker group forms an approximate rigid body of which the inter-marker distances remain nearly constant. For this reason, the above mentioned software systems also provide methods to recover a missing marker from a group of other markers if a rigid relationship between both the marker and the group may be assumed. However, in order to uniquely reconstruct the missing marker's position, at least three other markers or joint positions relative to the missing marker's segment need to be present in the gap.

Herda et al. [HFP+00] develop a skeleton based marker tracking and reconstruction technique to infer the positions of missing markers by using kinematic information provided by the underlying skeleton and the markers' positional data from previous frames that are attached to the same bone. This method is applicable to short time occlusions of single markers, but fails if entire segments are occluded for extended time periods.

Kalman filters are used in [DU03] to predict the trajectories of missing mark-

ers. However, Kalman filter based approaches fail when markers are missing for an extended time period or are missing entirely.

Li et al. [LMPF10] propose a method for occlusion filling of marker data by learning a linear dynamical system that respects inter-marker distances. However, their method relies on the existence of other markers on the same segment to make inter-marker distance measurements possible at all.

A data-driven method, which uses a piecewise linear modeling approach, is proposed by Liu and McMillan [LM06] for estimating missing markers.

Additional data-driven methods for cleaning motion capture data have been proposed [LC10, XFH11]. Lou and Chai [LC10] are able to filter corrupted motion data by learning a series of spatial-temporal filter bases from prerecorded motion data. Using their filtering approach in a nonlinear optimization framework they are able to reduce noise, remove outliers and fill gaps while keeping the spatial-temporal patterns of the filtered human motion intact. Their method requires creation of a training database in a time consuming pre-processing step which exclusively contains motions similar to the motion to be cleaned. Thus, in contrast to the method in this chapter, it cannot handle different motion styles simultaneously without expensive preprocessing.

Xiao et al. [XFH11] devise a method for filling gaps by representing incomplete poses by a linear combination of a few poses from a training set. Their approach as well as the work in [LC10] requires the training mocap data to be clean and to contain similar motion patterns (of the same actor) as the input motion. Moreover, the robustness of their approach to additional unrelated data in the training set is not discussed.

The problem of pose and motion reconstruction from sparse markers has also been the topic of various papers. In [GMHP04] and its accompanying video, the authors show the reconstruction of motion from sparse marker data. Although the results of their method are visually appealing, it largely depends on a specifically learned model that fails to capture the natural diversity of human motion. In [CH05], Chai and Hodgins show how to transform the positions of a small number of markers to full body poses. They construct a *neighbor graph* with the poses of the prior database as vertices. In a preprocessing step, an edge between two poses is added to the graph if the poses are near each other. This limits the NN-search to poses

already in the database and can only give approximate results if the query pose is not contained within the available example motions. Due to its quadratic preprocessing time, it does not scale well with respect to the size of the database. Moreover, in the optimization step, the synthesized motion depends on the positional information contained in the prior database while completely ignoring the temporal evolution (e.g., velocities and accelerations) of the local model. This might be an issue at turning points in the motion's trajectory. The method presented in this chapter incorporates this additional information yielding smooth and natural results. Krüger et al. [KTWZ10] improve on the method presented in [CH05] by using a kd-tree for determining the neighborhood of a query pose resulting in exact neighborhoods for arbitrary query poses.

Since the method presented in this chapter is data-driven, it uses motions from a mocap database to construct a prior-database. Currently, the largest freely available database is the Carnegie Mellon University mocap database [Car04], which contains 2605 trials in 6 categories and 23 subcategories. Another large database, the HDM05 library [MRC+07], was recorded at the Hochschule der Medien in Stuttgart and contains more than three hours of systematically recorded and well documented motion capture data. Both of these databases provide the data in marker based (c3d) as well as skeleton based (asf/amc) data formats.

2.3 Overview

The presented approach takes advantage of data driven techniques. For that reason, a mocap-database containing motions which are comparable to the clip to be processed is needed. One fundamental assumption of the proposed method is that all poses contained in the database as well as the motion to be cleaned share the same marker set. In this work, the marker set presented in Table 2.1 is used. Furthermore, it is assumed that valid markers — i.e., the set of markers that are assumed to contain reliable positional information — are given for each frame of the input motion to be completed.

In a preprocessing step all mocap data from the prior-database are first normalized with respect to global position and orientation. Then, an efficient spatial indexing structure (kd-tree) is built based on normalized positional data of valid

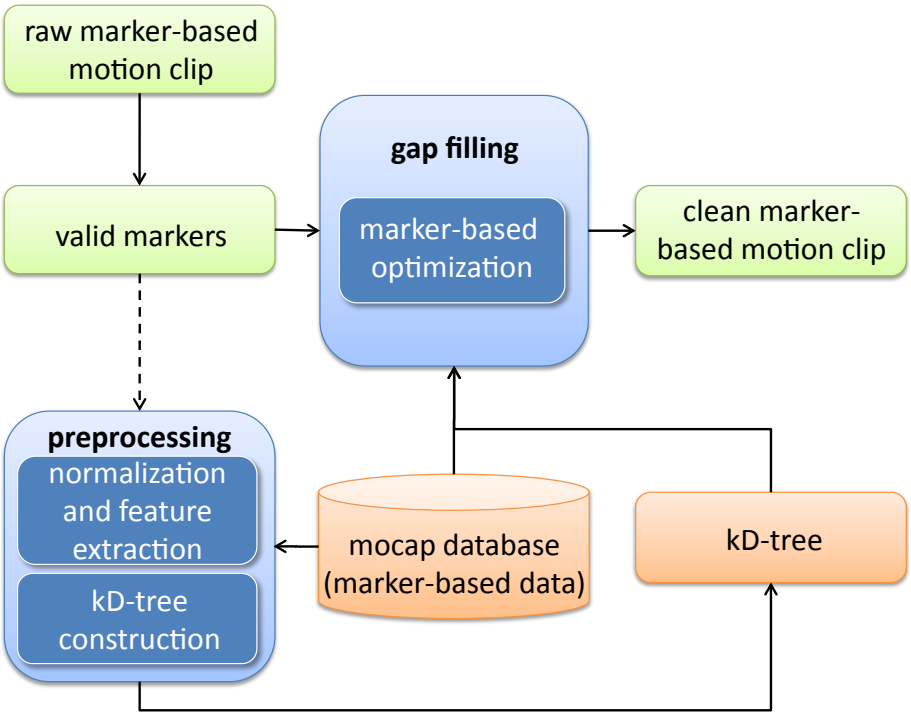


Figure 2.1: Workflow of the proposed method.

markers. In addition, linear marker velocities as well as accelerations are calculated using finite differences and stored in the mocap database. These quantities contribute to continuity and smoothness of the prior-based motion synthesis. The whole preprocessing step is explained in more detail in Section 2.4.1.

Subsequently, missing markers are synthesized for a given motion clip using non-linear optimization. To this end, similar examples from the database are retrieved by kd-tree based nearest neighbor search. These examples serve as priors to drive the synthesis process as discussed in Section 2.4.2. The whole pipeline of the proposed method is sketched in Fig. 2.1.

2.4 Workflow

2.4.1 Preprocessing

The presented method is inspired by the solution to the *pose matching* problem presented by Krüger et al. [KTWZ10]. Here, the key idea is to analyze similarity of poses

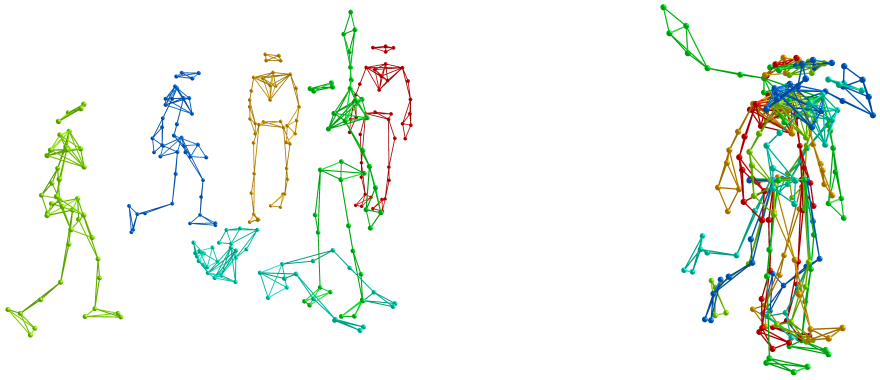


Figure 2.2: Examples of normalizations of poses represented as marker point cloud data. In order to give meaning to positional comparisons, original poses (left) are translated to the global root position and rotated so that the waist faces forward and lies in one plane with the horizontal plane (right).

by employing kd-tree based k-nearest-neighbor-search in dedicated feature spaces. Please note that this approach requires normalization of all poses with respect to global orientation and position. As — in contrast to Krüger et al. [KTWZ10] — no skeleton representation but point cloud marker data is given, normalized poses are estimated by exploiting rigid connectivity between segment markers. Poses can be normalized using the markers of any valid segment. In this work, the centroids of the poses' waist markers are translated to the global root position and rotated so that the waist planes are parallel to the horizontal plane and pointing along a common fixed axis. Examples of poses and their normalized counterparts can be seen in Fig. 2.2.

Let $\boldsymbol{x}$ be a pose vector involving M markers, where components are given by positional marker data. Using the motion database (kd-tree), a search for the k nearest neighboring poses $(\boldsymbol{y}_i), i \in \{1, \ldots, k\}$ is done using a subset of all markers. The actual choice of this subset is based on two different criteria. First, only reliable markers are considered where the placement is well-defined according to the markerset, such as the knee and elbow markers for the standard markersets used in [MRC+07] and [Car04]. This first criterion can be formalized by a static bit vector $(\tilde{\boldsymbol{m}} = \tilde{m}_i), i \in \{1, \ldots, M\}$ that determines if a marker is suitable for k-nn search (one) or not (zero) (see Table 2.1). Second, only markers that are valid according to the capturing logic are considered. Such markers are indicated by another bit

vector $\boldsymbol{m} = (m_i), i \in \{1, \ldots, M\}$. In contrast to $\tilde{\boldsymbol{m}}$, which is independent of the motion to be cleaned, the bit vector $\boldsymbol{m}$ is computed per gap that is to be filled. Once viable markers are selected, i.e., markers with $\tilde{m}_i = m_i = 1$, their respective coordinates form a vector space that is used for building a kd-tree from all motion data included in the database. As missing markers depend on the actual motion, this kd-tree is build from scratch for each cleaning process. Please note that building this tree takes only a few seconds even for the largest currently available databases and thus does not resemble a bottleneck in the method. Using finite differences, linear velocities and accelerations are computed in advance from normalized poses for all motion clips contained in the mocap database used for cleaning. Fig. 2.3 shows a visualization of neighboring poses and their respective local velocity and acceleration vectors on an example pose.

2.4.2 Gap Filling

For each pose that requires to be cleaned by the proposed technique, a search for the k nearest neighbor poses is performed. To this end, each of the given frames is normalized with respect to position and orientation, similar to the data in the knowledge base. A set $(\boldsymbol{y}_i), i = [1..k]$ of k nearest neighbors is retrieved that can be used for the data-driven gap filling procedure, which uses prior-driven optimization to synthesize the positional data of the missing markers. An energy minimization formulation is employed which is frequently used in data driven computer animation. The specific choice of the energy terms to be minimized most closely resembles the one used in [TZK+11]. In this case, the objective function consists of three different terms: pose prior E_{pose} enforcing position and motion priors E_{motion} and E_{smooth} enforcing velocities and accelerations of the missing markers to be comparable to examples retrieved from the database.

$$\boldsymbol{x}_{\text{best}} = \underset{\boldsymbol{x}}{\text{argmin}}(\ E_{\text{pose}}(\boldsymbol{x}) + E_{\text{motion}}(\boldsymbol{x}) + E_{\text{smooth}}(\boldsymbol{x})\ ) \tag{2.1}$$

Prior Terms

Let $(\boldsymbol{y}_i), i = [1..k]$ be the poses retrieved from the database by k-nearest-neighbor search and $(\boldsymbol{\nu}_i), i = [1..k]$ and $(\boldsymbol{\alpha}_i), i = [1..k]$ their respective velocities and acceler-

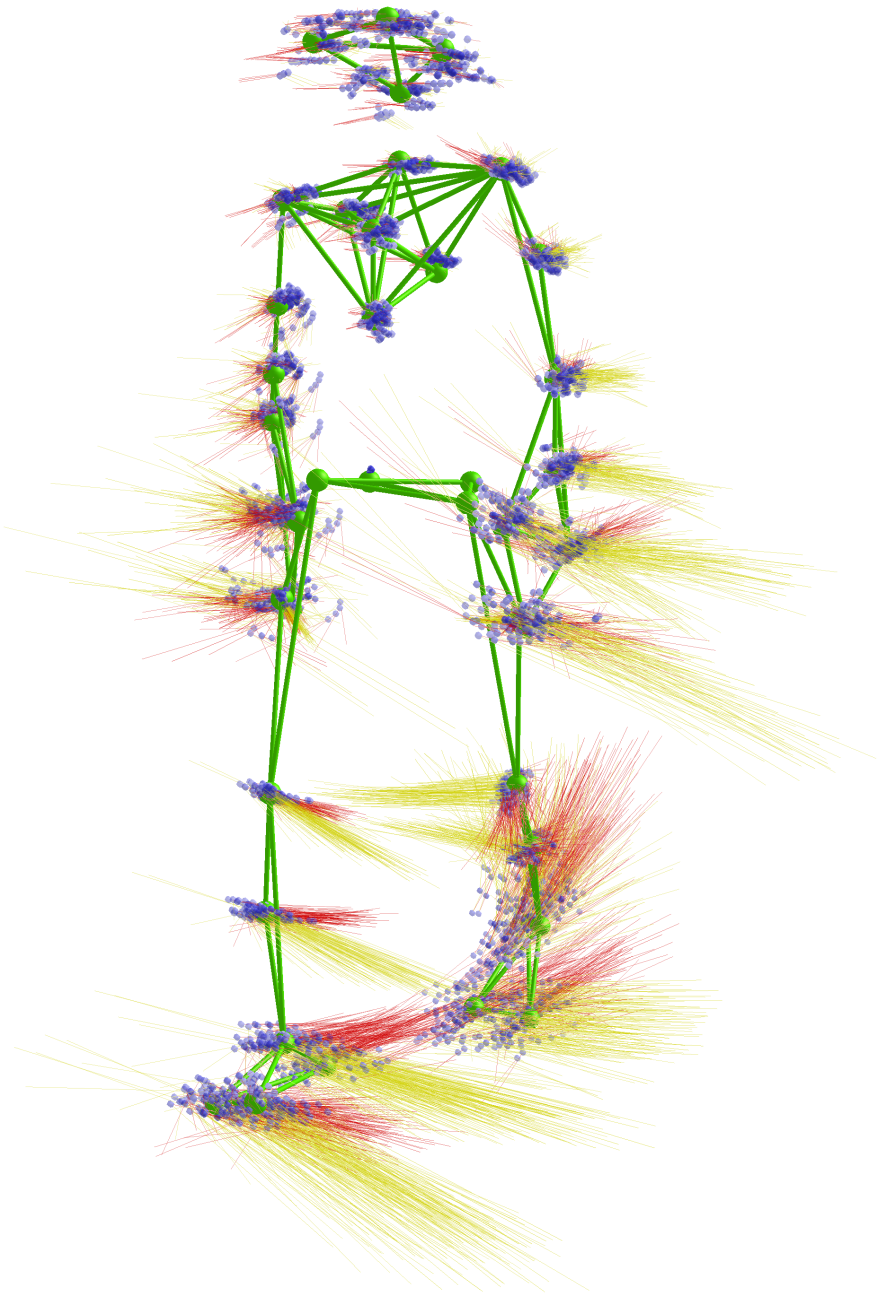


Figure 2.3: Local pose neighborhood visualization of a walking motion. Neighboring poses' marker positions are shown in blue, vectors of local velocities and accelerations in red and yellow, respectively.

Table 2.1: Table of markers, their body locations and if usable as a feature for the presented method. To determine reliable markers that are (if valid) suitable for k-nearest-neighbor-search, a bit vector $\tilde{\boldsymbol{m}}$ is used. Here, for each marker of the markerset, a component indicates if a marker is usable (1) or unreliable (0).

Label	**Location**	$\tilde{\mathbf{m}}$	**Label**	**Location**	$\tilde{\mathbf{m}}$
LFHD	Left Front Head	1	RWRB	Right Wrist B	1
RFHD	Right Front Head	1	RFIN	Right Fingers	1
LBHD	Left Back Head	1	LFWT	Left Front Waist	0
RBHD	Right Back Head	1	RFWT	Right Front Waist	0
C7	Vertebra C7	1	LBWT	Left Back Waist	0
T10	Vertebra T10	0	RBWT	Right Back Waist	0
CLAV	Between Clavicles	0	LTHI	Left Thigh	0
STRN	Sternum	1	LKNE	Left Knee	1
RBAC	Right Back	0	LSHN	Left Shin	0
LSHO	Left Shoulder	0	LANK	Left Ankle	1
LUPA	Left Upper Arm	0	LHEE	Left Heel	1
LELB	Left Elbow	1	LTOE	Left Big Toe	0
LFRM	Left Forearm	0	LMT5	Left Small Toe	0
LWRA	Left Wrist A	1	RTHI	Right Thigh	0
LWRB	Left Wrist B	1	RKNE	Right Knee	1
LFIN	Left Fingers	1	RSHN	Right Shin	0
RSHO	Right Shoulder	0	RANK	Right Ankle	1
RUPA	Right Upper Arm	0	RHEE	Right Heel	1
RELB	Right Elbow	1	RTOE	Right Big Toe	0
RFRM	Right Forearm	0	RMT5	Right Small Toe	0
RWRA	Right Wrist A	1			

ations. Let $\boldsymbol{\nu}(\boldsymbol{x})$ and $\boldsymbol{\alpha}(\boldsymbol{x})$ be the velocity and acceleration of a given pose. Then, kernel regression is used for each of the prior terms along the lines of [TZK+11] considering only markers that are assumed to be invalid:

$$E_{\text{pose}}(\boldsymbol{x}) = \sum_{i=1}^{k} (\bar{\boldsymbol{m}} \circ (\boldsymbol{y}_i - \boldsymbol{x}))^2 \tag{2.2}$$

$$E_{\text{motion}}(\boldsymbol{x}) = \sum_{i=1}^{k} (\bar{\boldsymbol{m}} \circ (\boldsymbol{\nu}_i - \boldsymbol{\nu}(\boldsymbol{x})) \cdot \Delta t)^2 \tag{2.3}$$

$$E_{\text{smooth}}(\boldsymbol{x}) = \sum_{i=1}^{k} (\bar{\boldsymbol{m}} \circ (\boldsymbol{\alpha}_i - \boldsymbol{\alpha}(\boldsymbol{x})) \cdot \Delta t^2)^2 \tag{2.4}$$

with $\bar{\boldsymbol{m}}$ denoting the component wise inversion of the bit vector $\boldsymbol{m}$. Please note that for all the above priors only markers that are assumed to be invalid are considered by the component-wise operating Hadamard product ($\circ$).

2.4.3 Optimization Procedure

The objective function (2.1) is minimized using gradient descent. To improve efficiency, only a subset of all frames is considered during optimization. This includes frames with the highest associated costs as well as neighboring frames indirectly affecting reconstruction results through temporal derivatives occurring in motion and smoothness priors. This is referred to as *scheduling*. Also, to improve the robustness of the method and to speed up the process of optimization, a multi-resolution approach is employed where optimization takes place on subsequently higher temporal resolutions of the motion to be cleaned, starting with the lowest. This requires resampling the motion to a predefined number of lower resolutions. When the error on a certain resolution cannot be improved by at least a certain threshold (set to 1 % in this work), the algorithm upsamples the results and switches to the next higher resolution. Given the number of resolutions n and the highest resolution $r_{\max}$, lower resolutions r_i are calculated by

$$r_i = \frac{r_{\max}}{2^i}. \tag{2.5}$$

For every possible resolution, positions, velocities and accelerations have to be precomputed in the prior-database. Moreover, separate kd-trees have to be created.

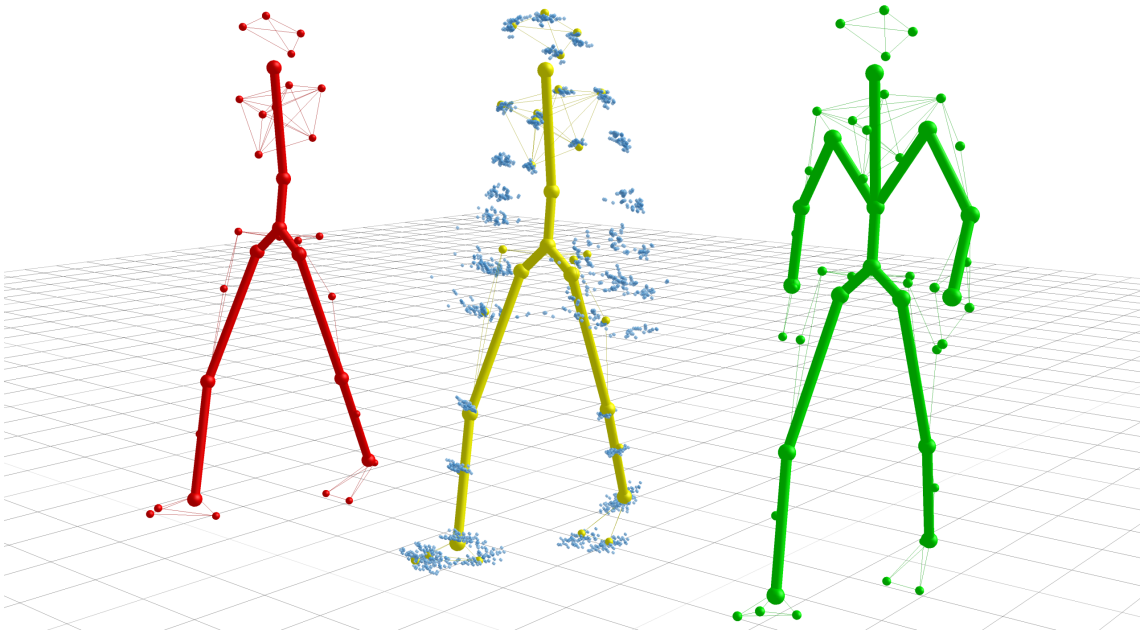


Figure 2.4: Reconstruction example showing a pose of a walking motion missing the markers of both arms (left), the corresponding database retrieved pose neighborhood (center) and the fully reconstructed pose (right).

Please note that the memory requirements of this multiscale approach is bounded by twice the memory consumption of the original data.

An example of the presented gap filling procedure is shown in Fig. 2.4, where the method is used to reconstruct the markers of both arms during the course of a walking motion.

2.5 Results

In order to evaluate the effectiveness of the proposed method, originally artifact-free mocap data is taken and certain markers or sets of markers representing body segments are discarded for time spans of varying lengths. Since ultimately, the visual perception and the possibility of using the resulting mocap data in practice is most important, the reconstruction results are analyzed not only numerically but also visually. Besides synthetic test cases the method is also employed on data containing real gaps.

For a visual comparison see the accompanying video contained in the supplemental material, showing:

1. Examples of real gaps in original marker data taken from the HDM05 database [MRC+07].
2. Reconstruction of a motion with missing markers on the left arm.
3. Gap-filled cartwheel motion with leg markers missing.
4. Comparison of databases according to section 2.5.1.
5. Example of a running motion that was presented and reconstructed in [LMPF10]. For this example the complete CMU database [Car04] was used as prior-database.
6. Comparison of reconstructions of a walking motion based on [KTWZ10] and the method presented here.
7. Reconstructions of gaps found in real motion capture data.

In Fig. 2.5, the dependency of the computation times on the length of the filled gaps is shown. As had to be expected, the computation times scale linearly with the durations of the gaps. There are certain variations with respect to the used motion classes and numbers of missing markers, but these effects yield much smaller variations than the primary dependency on the gap size. The computation times are obtained using a single threaded implementation on a Dual Core 3 GHz PC with 8GB of memory. Roughly speaking, the computation times are about 10 times the length of the longest gaps for this implementation. Hence, it is already practical for interactive applications even without having performed code optimization or using multi-threading.

The following conducted experiments show that the method is able to fill gaps in motions ranging from missing a single marker to missing multiple body segments for up to several seconds.

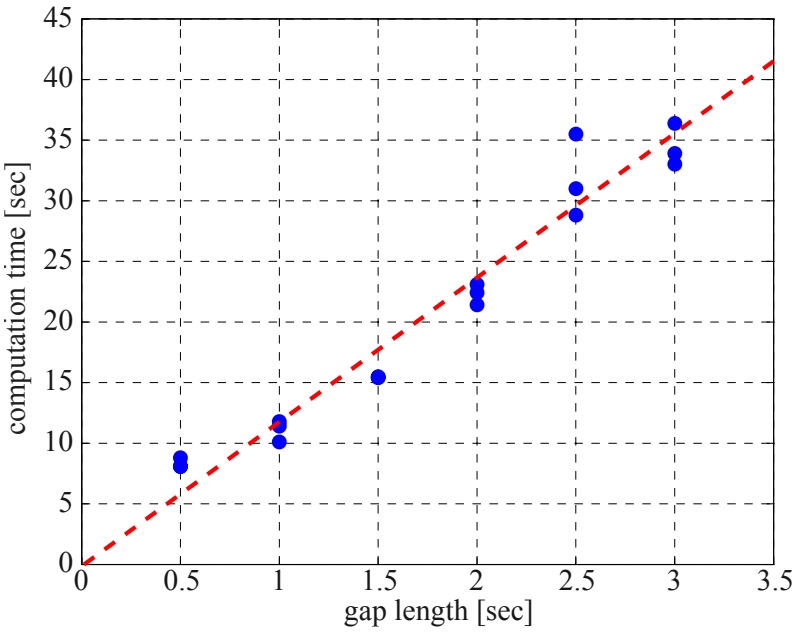


Figure 2.5: Computation times for various examples with respect to the length of the filled gap.

2.5.1 Evaluation on Synthetic Examples

This section reports on a series of tests on synthetic examples. Several aspects of the proposed method are evaluated. For this reason, markers are deliberately removed from intact motion sequences to be able to compare the results with ground truth data. All results are computed on motions resampled to 30 Hz.

Tests on Single Missing Markers

For three test motions taken from the HDM05 database, each marker is systematically removed and reconstructed. These test motions are:

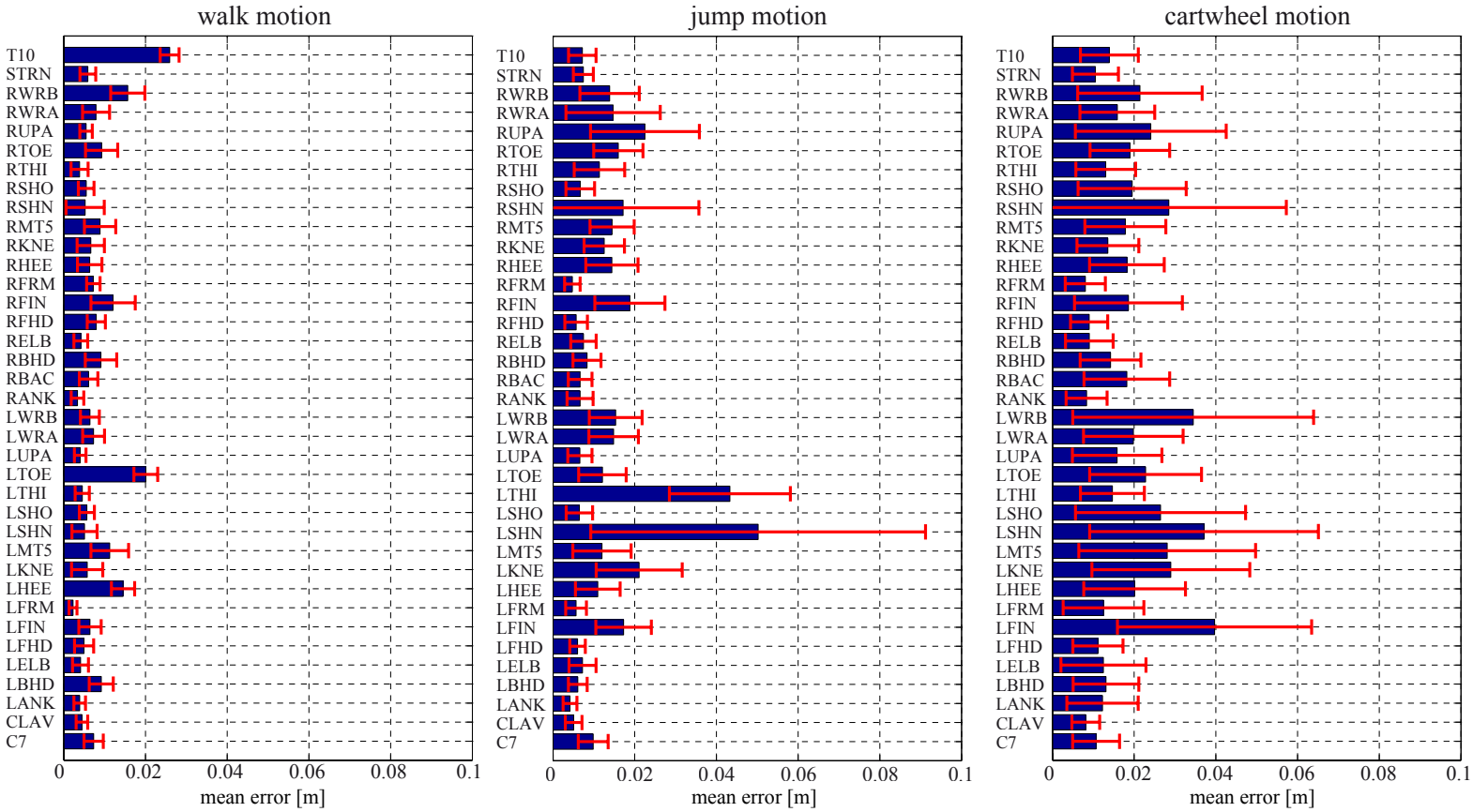


Figure 2.6: Mean distances (blue bars) between original marker trajectories and synthezised trajectories and the corresponding standard deviations (red lines) are shown for three testing motions, where the indicated marker was removed.

1. walk: `HDM_bd_01_01-01_120.c3d`,
   frames 650 – 1100

2. jump: `HDM_tr_01-05_01_120.c3d`,
   frames 1000 – 1350

3. cartwheel: `HDM_tr_05-03_03_120.c3d`,
   frames 2550 – 3000

The database used for these experiments included all motions from the HDM05 database except the whole take the test sequence was taken from. Figure 2.6 shows the results of this test. The mean distance between the original and the synthesized markers as well as the standard deviation of this distance is presented. As can be seen on the left of Figure 2.6, the walking motion gives very good results, showing a mean distance of 0.77 cm over all examples. For the more complex jump and cartwheel motions the means are 1.27 cm and 1.81 cm, respectively.

Tests on Groups of Missing Markers

On the three motions that were used on the single marker leave-out evaluation, more tests were performed where several groups of markers were removed simultaneously. For each test, six markers of the segments of the left arm were removed. For the walk and jump motion, these markers will not be in contact with the ground, whereas for the cartwheel motion a contact of the left arm with the ground occurs. For these tests, the following scenarios were regarded

1. The prior-database does not contain motions of the actor of the test motion.

2. The prior-database contains motions of various performers, including motions of the actor (other than the test motion).

3. Only motions of the actor were included in the prior-database.

The results of the tests are summarized in Fig. 2.7. If motions of the actor are not contained in the prior-database, the average reconstruction errors are more than twice as high as in the other cases for all three examples. However, the reconstruction results are still good, with a mean error ranging from 2.5 cm for the walking motion

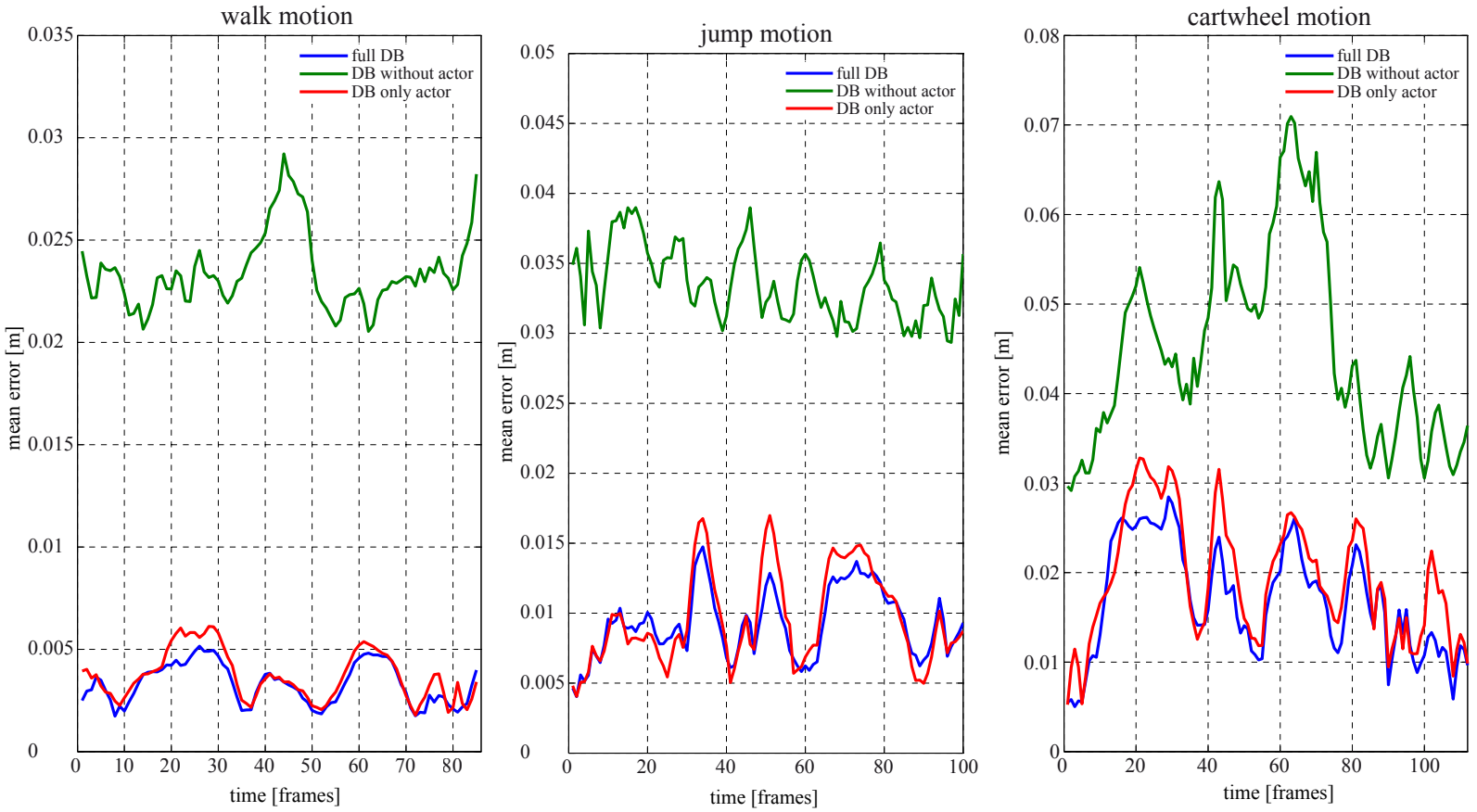


Figure 2.7: Mean distances over all markers for three motion sequences. The distances are presented for three different databases: The full database, excluding only the test motion (blue), a database where the actor was completely removed (green) and a database where only motions from the actor were included (red).

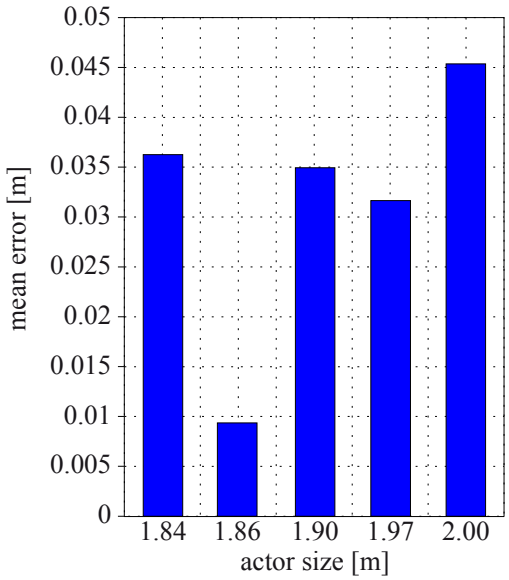


Figure 2.8: Results of tests with actors of different sizes. The mean reconstruction error is plotted for a walking sequence (`HDM_**_01_01_01_120.C3D`) versus the body size of the actors. The actors were not included in the database for this experiment.

to 5 cm for the cartwheel motion. Moreover, the reconstructed motions have a high visual fidelity (see the accompanying video).

In another test suite to estimate the influence of properties of the performing actor, leave-out tests are performed for any of the five actors doing walking motions in the HDM05 database. Again, marker positions for the left arm are first removed and then reconstructed. For this experiment, the takes `HDM_**_01_01_01_120.C3D` were used which were performed by each actor for this test. The results are summarized in Fig. 2.8.

2.5.2 Comparison with Previous Work

The results of the presented method are compared with the motion reconstruction technique described by Krüger et al. [KTWZ10], which is an extension, based on fast similarity searches, of the technique originally described by Chai and Hodgins [CH05]. For this comparison, six scenarios of missing data were regarded: left arm, right arm, both arms, left leg, right leg and both legs. Since the presented method only reconstructs missing marker positions, a skeleton was fit to the ground truth and reconstructed marker data using the method of de Aguiar et al. [dATS06].

To make the marker fit skeleton comparable to the standard asf/amc skeleton used in the HDM05 library, a set of joints was selected as the intersecting set of joints of both skeleton topologies. An obvious error measure to compare different reconstructions is to calculate the average distance between this set of joints of the reconstructed and the ground truth motion for every frame. The mean reconstruction error is then calculated as the mean over all individual frame errors. The numeric results of both reconstructions are given in Table 2.2.

The approach presented in this work uses multi resolution optimization combined with scheduling and a more sophisticated prior model to produce natural looking results while Krüger et al. use ad-hoc smoothness and framewise optimization combined with a kd-tree nearest neighborhood search. In most cases, the presented method proves to be numerically better using the standard error measure and the difference in the visually perceived quality of the results can also be seen in the accompanying video.

2.6 Conclusion and Future Work

This chapter presented a data driven method for filling large gaps in marker based mocap data. The method works well, even for large gaps of multiple seconds from the perspective of required computational resources as well as quality of results—provided that there are sufficiently similar motions available in the prior-database. The basic mechanism can be extended to other cleaning and reconstruction tasks, such as optimal skeleton fitting and correcting marker-mislabelings. These extensions will be one topic of future work.

In contrast to previous approaches, all available and cleaned motion capture data can be kept in the prior-database, and the approach scales well to huge prior-databases. The quality of the gap filling methods depends on the similarity of data contained in the prior-database and somewhat better results are obtained if motions of the performing actor of the clip to be cleaned are already contained in the prior-database. Nevertheless, the method also works quite well if such data is not available. In principle, in this approach it is possible to incorporate model knowledge about skeleton constraints and contact constraints. Using a good algorithmic heuristic to estimate contact constraints from motion data—e.g., the method pre-

Table 2.2: Results for motion reconstructions based on the method presented in this paper (this), compared to reconstructions based on the method of Krüger et al. [KTWZ10]. The table gives the mean reconstruction error in centimeters.

		scenario		
motion	**method**	**left arm**	**right arm**	**both arms**
HDM_bd_01-01_01_120	this	0.64	0.59	1.23
Frames: 650 – 1100	KTWZ10	1.00	1.56	2.29
HDM_bd_01-02_03_120	this	1.20	1.18	3.63
Frames: 450 – 750	KTWZ10	1.81	2.43	4.71
HDM_bk_01-01_03_120	this	1.29	1.47	4.80
Frames: 6300 – 6600	KTWZ10	2.05	2.44	5.91
HDM_mm_02-03_02_120	this	1.00	3.50	10.20
Frames: 450 – 750	KTWZ10	2.89	4.46	12.07
HDM_dg_01-06_01_120	this	1.20	0.94	1.81
Frames: 1000 – 1300	KTWZ10	1.68	1.26	3.69

		scenario		
		left leg	**right leg**	**both legs**
HDM_bd_01-01_01_120	this	0.88	0.82	2.47
Frames: 650 – 1100	KTWZ10	1.35	1.48	4.55
HDM_bd_01-02_03_120	this	1.60	2.20	4.00
Frames: 450 – 750	KTWZ10	1.63	2.19	4.62
HDM_bk_01-01_03_120	this	1.27	1.61	4.24
Frames: 6300 – 6600	KTWZ10	1.86	2.65	6.26
HDM_mm_02-03_02_120	this	0.95	1.11	2.00
Frames: 450 – 750	KTWZ10	0.97	1.37	1.85
HDM_dg_01-06_01_120	this	1.70	1.58	3.56
Frames: 1000 – 1300	KTWZ10	1.42	1.59	4.68

sented in [LCB06]—the contact information can be incorporated into the search and all defined constraints can be incorporated into the optimization procedure. One can presume that such information is useful in all settings and might be crucial if for a gap-filling the information of body segments such as lower or upper body parts only are considered. Such restrictions to body parts allow an extension of the notion of "similar motion" to ones being similar for body parts only.

Future work will explore the algorithmic techniques and should perform empirical investigations for incremental extension of the prior-databases: cleaned motion clips can be incrementally added to the prior-database potentially allowing a step-wise extension of the expressibility of the prior-database. With such extensions, motions which could not be handled by an original prior-database might become tractable by the newly added clips.

The scenario of missing markers on entire body segments for longer time periods is a common challenge even for single user capture using practical low-cost equipment such as the Kinect. Here, instead of markers, gaps in skeleton node trajectories could be reconstructed using database information. The integration of the presented algorithms into a capturing and processing pipeline for such low-cost devices could be a topic of future work.

3

Detection of Human Actions

This chapter presents an extended version of the publication:
Action Graph: A Versatile Data Structure for Action Recognition [BWKW14].

3.1 Introduction

Consumer motion capture systems (like Kinect, WiiMote, EyeToys, accelerometers) have received a lot of attention in recent years, primarily because they enable the user to interact with an application in a very natural way using low cost consumer hardware. The field of usage exceeds replacing the classic game controller in computer games. New applications beyond the field of computer games are emerging. This chapter is motivated by such a novel example application: The automated detection of Judo referee signals, i.e., the recognition of full body movements (of Judo referees) as belonging to a set of small motion segments which are detected as certain referee signals (usually denoted by their Japanese names). Taking the developed method to the gym would allow for cost-effective automatic score counting and time keeping and greatly reduce the administrative overhead required at Judo competitions.

Technically, a fully data-driven action recognition scheme is devised, where motion sequences can be detected in real-time. The method is very flexible concerning the used sensor input data, which can range from high quality optical motion capture data, over medium quality Kinect skeletons to highly noisy accelerometer readings. Adding robust feature extraction from video data to the recognition pipeline even enables the approach to detect actions from video input. All this various input

data can be compared in real-time with previously recorded sample motions in the database. The framework detects if the performed motion is similar (and possibly time-warped) to an annotated motion contained in the database.

The method requires very little preprocessing—only sample motions for each action to be recognized have to be labeled by the name of the action. No further explicit learning phases are required. Additional flexibility comes from the ability to add action templates to the used action database in an online manner, requiring only minimal processing.

For the purpose of evaluating different aspects of the method it is applied to prerecorded high quality motion capture data, to live captured low quality motion data obtained by a Microsoft Kinect sensor, to features extracted from video data and to a sparse accelerometer sensor setup with only four sensors attached to the actor's body. Interestingly, even for these very sparsely distributed accelerometers, the method is able to detect actions, making it very effective in a low-cost sensor setup.

The approach uses a framework for motion matching using k-nearest neighbor searches. It is shown that these previously devised techniques can be adapted and are then also very suitable for the task of action recognition.

3.2 Related Work

Related work for the method can be divided into four groups, image-based action recognition, 3D point trajectory action recognition methods, methods using accelerometers as sensors and data-driven techniques in the field of computer animation.

The first group of techniques uses 2D information such as images coming from a video camera to infer information about the actions taking place. The work by Bobick and Davis [BDSS01] presents a view-based approach to action recognition using *temporal templates*, which are static vector-images where the vector value at each point is a function of the motion properties at the corresponding spatial location in an image sequence. In [SLC04], Schüldt et al. use local space-time features in combination with SVM classification schemes for action recognition.

The second group works directly on 3D point trajectory data. Barbič et al.

[BSP+04] show methods for automatically segmenting motion capture data into distinct behaviours. Campbell and Bobick [CB95] present techniques for representing movements based on space curves in subspaces called *phase spaces*, recognizing actions by calculating distances between these curves at every time step.

Arikan et al. [AFO03] use an interactively guided *Support Vector Machine* to generalize example annotations made by a user to the entire motion capture database. Their approach works well on the small (7 minutes) motion capture database presented in their paper. The method presented in this chapter uses a similar SVM approach for comparison with the developed method.

Data-driven k-nearest neighbor approaches have been quite popular in the field of computer animation in recent years. In the context of synthesizing motions, Chai and Hodgins [CH05] show how to transform the positions of a small number of markers to full body poses. For nearest neighborhood pose searches, they construct a *neighbor graph*, allowing approximate NN-searches and requiring a quadratic preprocessing time in the size of the number of poses in the database. Krüger et al. [KTWZ10] improve the method presented in [CH05] by querying a kd-tree for determining the neighborhood of a query pose resulting in exact neighborhoods for arbitrary query poses.

A novel and very intuitive puppet interface is used by Numaguchi et al. in [NNSH11] to retrieve motions from a motion capture database. By sketching actions with the 17-degree of freedom puppet, the method matches the puppet's sensor readings retargeted to human motion to behaviour primitives stored in the motion database.

In [RKH11], Raptis et al. develop a method to classify human dance gestures by using a special angular skeleton representation designed for recognition robustness under noisy input. They use a cascaded correlation-based classifier for multivariate time-series data in combination with a dynamic-time warping based distance metric to evaluate the difference in motion between a performed gesture and an oracle for the matching gesture. Although the classification accuracy of their approach is very good, it assumes that the input motion adheres to the underlying musical beat, whereas the approach presented here does not rely on such assumptions.

Another class of methods is about using accelerometers for activity recognition. Bao and Intille [BI04] present a system designed for context-aware activity recogni-

tion detecting everyday physical activities from acceleration data. They focus on a semi-naturalistic data collection protocol to train a set of classifiers, and find this is best evaluated by decision tree recognition algorithms. Along the same lines, Maurer et al [MSSD06] study the effectiveness of activity classifiers also within a multi-sensor system. Their analysis of the proposed activity recognition and monitoring system concludes it is able to identify and record a subject's activity in real-time.

While Ravi et al. [RDML05] also study the activity recognition techniques, they present a solution that only uses a single triaxial accelerometer worn within different data collection setups. Within this context, they analyze the quality of known classifiers for recognizing activities with particular emphasis on the importance of selected features and level of difficulty of recognizing specific activities. The system developed by Khan et al. [KLLK10] is capable of recognizing a broad set of human physical activities using only a single triaxial accelerometer. The approach is of higher accuracy than the previous works due to a novel augmented-feature vector. Additionally, they provide a data acquisition protocol using data collected by the subjects at home without the researcher's supervision. Similarly, Wyatt et al. [WPC05] describe techniques for mining simple discriminative models of arbitrary object-based activities and for controlling the precision and accuracy of the resulting classifications. The novelty of their approach lies in the description of how to learn labeled models of physical activity from sensor data without any human intervention per activity, even for the annotation of the data.

Since activity-aware systems have inspired novel user interfaces and new applications, recognizing human activities in smart environments becomes increasingly important. In this spirit, Choudhury et al. [CBC+08] propose an automatic activity recognition system using on-body sensors. Several real-world deployments and user studies show the relevance of using the results to improve the hardware, software design, and activity recognition algorithms to context-aware ubiquitous computing applications. In a similar spirit, [KWM11] introduce their activity recognition technique which uses cell phone accelerometer data trained from users as they performed daily activities to induce a predictive model for activity recognition.

This chapter presents a data-driven method that uses motions from a motion capture database to construct a prior-database. Publicly available datasets, like the Carnegie Mellon University motion capture database [Car04] and the

HDM05 [MRC+07] library, recorded at the Hochschule der Medien in Stuttgart, contain large amounts of motion capture data. In this work, the data of the HDM05 library is used, which contains more than three hours of systematically recorded and well documented motion capture data. Of great benefit in the evaluation of the action recognition method are the manually cut out motion clips that were arranged into 64 different classes and styles. Each such motion class contains 10 to 50 different realizations of the same type of motion covering a broad spectrum of semantically meaningful variations. The resulting motion class database contains 457 motion clips with a total length of roughly 50 minutes of motion data.

3.3 Overview

The workflow of the proposed action recognition method (see Fig. 3.1) can be divided into three distinct processes. First, in an offline step, the motion capture database is created from motion data, where the quality can range from sparse and noisy data, such as that of a sensor setup using a single accelerometer only, to highly accurate optical motion capture data. All such data sets can easily be handled and are manually or automatically annotated by specifying start and end frames as well as a keyword for labeling. This is followed by a preprocessing phase in which a kd-tree is created using a specific feature set allowing fast k-nearest neighborhood searches on the poses stored in the database. This feature set depends on the application which, in turn, is interdependent on the specific type of motion capture system respectively the employed sensor setup.

Since the approach at hand is generic, the input need not be of a specific data type and may even cover cross-modal signals. In the online phase, actions are recognized from any type input motion sequence by feeding new frames of the input motion into the annotation module. This module uses similar poses retrieved from the kd-tree in a neighborhood graph called *Action Graph* to output all recognized actions as soon as they are detected.

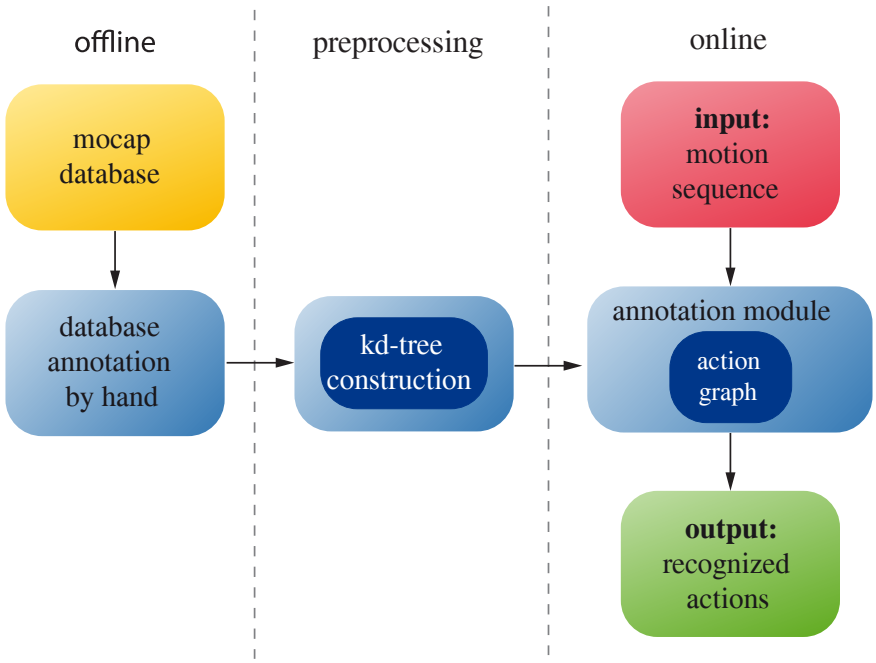


Figure 3.1: Workflow of the proposed method. Starting out with a motion capture database annotated with actions of interest in an offline phase, the method builds up a kd-tree from this data in the preprocessing phase. The online phase then consists of feeding new frames of the input motion into the annotation module, recognizing actions as soon as the actor finishes executing them.

3.4 Action Recognition Methods

3.4.1 Data Preparation

Since the preparation of motion capture data for the presented method is highly dependent on the application and sensors used, no general rules for preparing the data in the database or for processing the poses of the query motion can be given. Various applications presented in the results in Section 3.5 show realistic examples.

3.4.2 Data Annotations

For the training phase of the action recognition methods, as well as for evaluations during the testing phase, accurate annotations are needed. Annotations inform the system at which time in a motion sequence specific actions are performed by the actor. For this reason, in this method, all used datasets were annotated by hand.

Another possibility would be to use automatically annotated mocap data, e.g., by methods presented in [AFO03] or [WPC05]. In this work, the decision was made on a complete, reliable and manual annotation procedure to ensure that the results are not affected by possibly false, automatically computed annotations. For each relevant action, an annotator gives a start frame, an end frame and a keyword that describes the performed motion, ultimately creating a mapping from database frame f to the annotations stored in f.

3.4.3 Action Graph Based Recognition

The presented action recognition method searches for motion segments that are similar to annotated actions in the motion database by taking into account the temporal continuity of the underlying motion. This is in contrast to the SVM-based approach presented in Section 3.4.4 for comparison, which ignores this information and decides whether a frame belongs to an action on a frame-by-frame basis, leading to many possible ambiguities. To avoid these, the *Action Graph* detects if an action ends at the current frame and then searches the input motion's history to find possibly time-warped motion segments spanning the action in its entirety. Looking at the individual pose neighborhoods of the knn-search alone can lead to possibly many different annotations. By using the *Action Graph*, paths representing motion segment matches can be found through the annotated neighborhoods, resolving the ambiguity.

Basically, the method presented in this chapter extends the *Lazy Neighborhood Graph* (LNG) proposed by Krüger et al. [KTWZ10] to find motion segments similar to the currently performed motion. In its original implementation, the LNG first retrieves the k nearest neighbors from a motion database for every pose in the query motion. To bridge the gap from these locally matching poses of the retrieved pose neighborhoods to globally matching similar motions in the database, their method constructs a directed acyclic graph by regarding the retrieved local neighboring poses from the motion database as vertices in the graph. Now, an edge connects a pair of neighbors of temporally adjacent pose neighborhoods, if certain step size conditions are satisfied, similar to *Dynamic Time Warping*. In its simplest form, the step tuple $(step_{pose}, step_{time}) = (1, 1)$ connects pose index p at time t to pose index $p + 1$ at time $t + 1$. By allowing additional step tuples, e.g., $(2, 1)$, $(1, 2)$, the results could

also possibly be time warped. After having made all possible connections, a single source vertex s is connected to all the pose neighbors in the first neighborhood. The problem of finding a motion contained in the database which is most similar to the query motion can now be reduced to solving a single-source shortest paths problem. Starting the search at vertex s, the algorithm only has to check whether there exists a path that terminates at a vertex in the last neighborhood. The entire global matching can be solved in $O(km\log(n))$, where k is the number of retrieved nearest neighbors, m the number of frames contained in the query motion and n the number of frames in the motion capture database.

In contrast to the original implementation of the LNG, the developed action recognition framework tries to find motion segments which start close to the beginning of an annotated action having the currently processed frame close to the terminating frame of an annotation (see Fig. 3.2). This is accomplished by first inspecting the pose neighborhood of the current frame for annotated action ending poses. Now, every annotated action starting pose containing the same annotation as the found ending pose in the pose neighborhood queue of a certain window size w is connected with the single source vertex mentioned above. The parameter w is chosen so that all possible actions from the database (including time warped actions) fit into the window. If an annotated action in the database is similar to the currently performed motion and is contained in the pose neighborhood queue of length w, the single-source shortest paths algorithm is able to find paths from the beginning of an action to the end of the action, containing only the specified annotation. The found actions are possibly time-warped according to the allowed time-steps, making the method very flexible and robust to time variations in motion performance.

In the technical part of the motion detection scenario, a query motion take (e.g., a Kinect recording), is tested against a set of query action classes A, that is, annotated classes that are present in the database. As a result of searching the *Action Graph* for motions associated with these class annotations a set of path candidates $C = \{s_i\}$ is returned, consisting of similar motion segments s_i. The size of this set depends on the employed database, but may range from zero to several thousand retrieved segments. Consequently, the percentage of detected paths s_i from the *Action Graph* which agree with the query set A is computed, thus automatically addressing the fundamental question whether one came across any action contained

in A. The annotations which are represented most strongly are collected, i.e., make for more than 10% of the whole set C, and their respective start and end frames are computed as follows. The frame window which contains the intersection of the annotation ground truth and a minimum of 75% of all according paths retrieved from the *Action Graph* is computed. The start and end of this window marks the start and end frame of the annotation at hand. Note that when the aim is evaluation rather than detection, a slightly different protocol is followed. This will be addressed in Section 3.5.

3.4.4 SVM-Based Recognition

In the evaluation part of this chapter, two different action recognition methods are compared side-by-side, the online-capable *Action Graph*, where the input motion sequence is efficiently compared to annotated motions in the motion capture database, and a *Support Vector Machine* (SVM) approach similar to one that was introduced for motion capture data by Arikan et al. [AFO03]. Here, frame classification based on a Support Vector Machine (SVM) with a standard Radial Basis Function kernel (RBF) was implemented. To this end, the LibSVM implementation [CL11] was used. Optimal SVM parameters C balancing hyperplane minimization and the influence of slack variables as well as the RBF kernel width γ are determined using grid search with cross validation. To reduce the time consumption for training, only 30% of the frames of each training sequence were used and chosen randomly. To take the possible influence of this random selection into account, four runs were conducted, each time using a different training frame selection and the resulting classification accuracies were averaged.

The SVM and the *Action Graph* based methods basically share the same workflow (see Fig. 3.1). Within the SVM-based situation, the preprocessing phase consists of learning SVM parameters on a training set, whereas in the online phase, the SVM classifier checks whether a frame derived from the input query motion belongs to a previously annotated action.

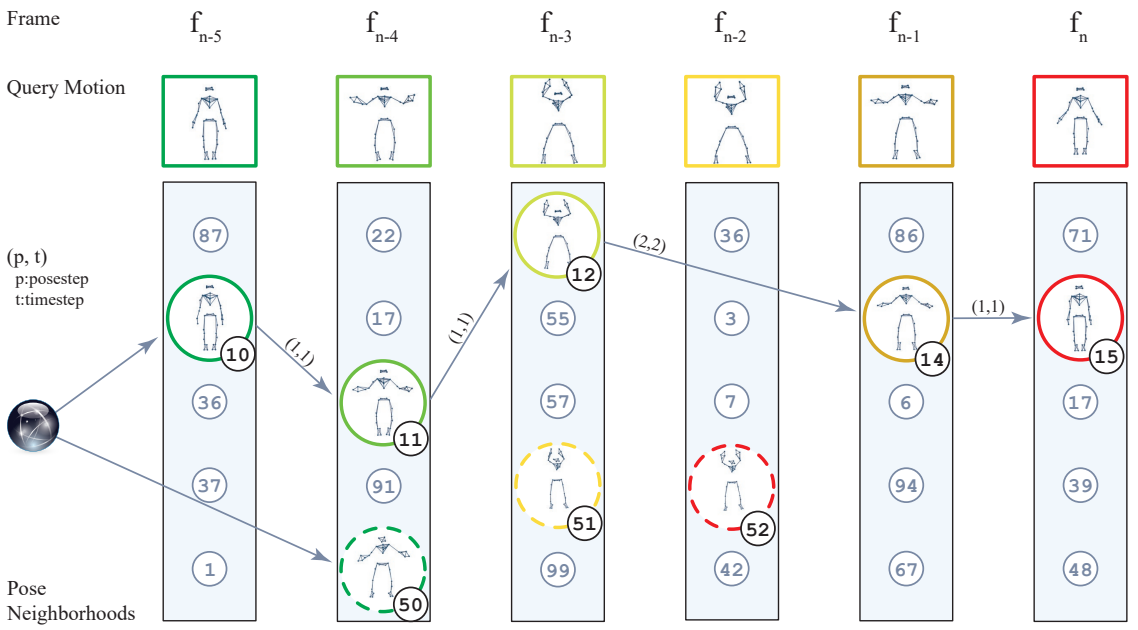


Figure 3.2: Detecting actions in current frame f_n using the Action Graph. *In this example, detecting a 'Jumping Jack' motion is illustrated which is performed in the last six frames (f_{n-5} to f_n) using a window of $w = 6$ frames. The poses of the 'Jumping Jack' are color coded, ranging from green to red, representing the start and end of the action, respectively. First, all poses annotated with starting poses of actions (green) are connected to the single source vertex required for the single-source shortest path algorithm, regarding all past neighborhoods up to window size w. Now, for every neighborhood, poses are connected with edges according to the allowed time and pose steps S ($S = \{(1,1),(2,2)\}$ in this example). After running the single-source shortest path algorithm, the method checks for every candidate path terminating at an action ending pose (red pose in neighborhood f_n) whether the nodes on the path are consistently annotated with the same action, in which case this action is reported as found. Note that every 'JumpingJack' motion contains a 'ClapAboveHead' in its middle, as can be seen in the pose neighborhoods (dashed circles). Consequently, this clap was also detected by the algorithm, but at an earlier stage (frame f_{n-2}).*

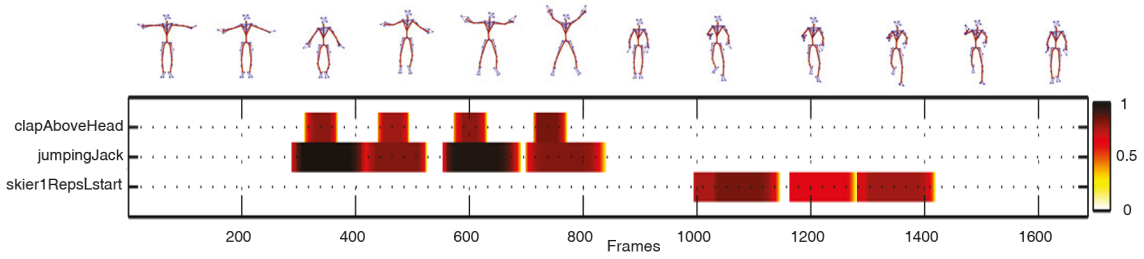


Figure 3.3: Example action recognition run on frames $0 - 1686$ *of motion HDM_bd_03-05_02_120 from the HDM05 motion capture library, consisting of four jumping jack motions followed by three complete and one half skiing motion starting with the left foot. Note that the action recognition method also detects sub-actions like the clap above the head, which is contained in the middle of each jumping jack. The half-executed skiing motion at the end is not detected, because the* Action Graph *is unable to find an annotated end frame in this case.*

3.5 Results

3.5.1 Applications Used for Evaluation

For evaluation purposes, six applications were considered. First, in Section 3.5.4, action recognition tests were performed on the cut dataset taken from the HDM05 motion capture database. Here, the cut sequence database was separated into a training part that contained exactly nine realizations of each motion class, and a testing part that contained at least three realizations. The same motion capture database is then used to test the algorithm on a sparse accelerometer setup, detecting actions using a total of four simulated accelerometers on the wrists and ankles.

In Section 3.5.5, the behavior of the methods was tested with Judo referee signal movements in an online scenario, using query motions coming from an optical mocap system. In this scenario the database contained typical referee signals performed by three different actors with at least three repetitions. This database was captured with a Vicon motion capture system and the motion capture data was stored in the skeleton based .v file format.

The previous scenario was also modified to a cross-modal scenario, so that the query motion was captured with a Microsoft Kinect sensor, obtaining the skeletal data using Microsoft's Kinect SDK. For this reason, the Judo database had to be resampled to the native frequency of the Kinect sensor (30 Hz).

In the next application example (Section 3.5.6), interest points extracted from video data serve as input for the proposed algorithm, demonstrating applicability in a vision-based context.

Lastly, in Section 3.5.7, an attempt is made to extract usable feature set data from laser range scanner recordings.

Some applications which require poses in the database to be comparable need to perform a normalization step on each pose, making them scale- and view-invariant. Along the lines of Krüger et al. [KTWZ10], the root node of the skeleton is transformed such that the skeleton faces forward and is anchored at the global coordinate frame origin. If the skeleton is given in a hierarchical representation (e.g., HDM05 and Judo database skeletons), the root node's position is translated to $(0, 0, 0)^T$ and its orientation is set to the multiplicative identity quaternion, followed by a forward kinematics calculation to update the remaining skeleton nodes. When normalizing skeletons where joint positions are given in absolute world coordinates with no rotational information (e.g., Kinect skeletons), the orientation of the root node is estimated by exploiting rigid connectivity between the pelvis and its neighboring joints, similar to the normalization step used for raw optical marker data in Chapter 2.

To obtain scale-invariance, the bones of any query skeleton are resized to match the skeleton that was used to build up the database.

3.5.2 Description of the Evaluation

Allowing detection of more than one particular action at a time does not make sense for evaluation of the detection method, especially when this is done by means of confusion matrices. The decision criteria presented in Section 3.4.3, which allow for several strongly represented action classes to contribute to the detection results, are clearly not suitable for evaluation purposes. Instead, a choice is made for the single most strongly represented action class found in the *Action Graph* paths. To evaluate the quality of the decision method, the following cases are distinguished:

1. The retrieved motion paths lie completely within the relevant ground truth interval, in which case the method is regarded as properly working.

2. The motion paths lie outside the ground truth interval as a whole, in which

case the method is dismissed as incorrect (this case was rarely observed).

3. The retrieved path set intersects the ground truth interval, in which case further differentiation is necessary: if the intersection includes more than 90% of the total retrieved paths, the method is considered to work well, otherwise this hypothesis is dismissed.

According to the above, matrices similar to confusion matrices are used to visualize the performance of the action recognition algorithm. The columns of the matrix represent instances of the recognized actions while the rows represent the actual actions. Taking into account the cases in which the algorithm fails to detect any action, a column labeled *none* is added. A perfect action recognition would have a confusion matrix with 1 on the main diagonal and 0 for every other element.

3.5.3 Details on the knn Search

Choosing a feature set for the HDM05 cut database was straightforward. The results from Krüger et al. [KTWZ10] indicated that feature set $\mathcal{F}_{\mathrm{E}}^{15}$, which includes the positions of the head, hands and feet, would work very well on this database. The confusion matrices presented in Fig. 3.4 confirm that this assumption holds. Instead of directly including temporal information in the feature set, it is implicitly encoded in the structure of the *Action Graph*: Edges are inserted between successive database indices, according to the allowed step size conditions.

According to [KTWZ10], the steps sizes (1,1), (2,1), (1,2) and (2,2) are used. The first three step sizes allow normal speed steps, half speed steps and double speed steps, respectively, while the last step size allows the algorithm to completely skip a single neighborhood at normal speed. In this configuration, the *Action Graph* easily runs in real-time when searching for 256 nearest neighbors, achieving an average frame rate above 75Hz in a multi-threaded implementation on the regarded motion capture databases. The described results were obtained using a system with an Intel hex core cpu with 3.33 GHz and 24 Gb of memory.

The knn search used in our approach can be replaced by a fixed radius search. This variation does not produce convincing results, due to the following reasons: First, a fixed radius can mean that the method does not find any neighbors. Second, as was determined in tests of this variant, the variability between motions in some

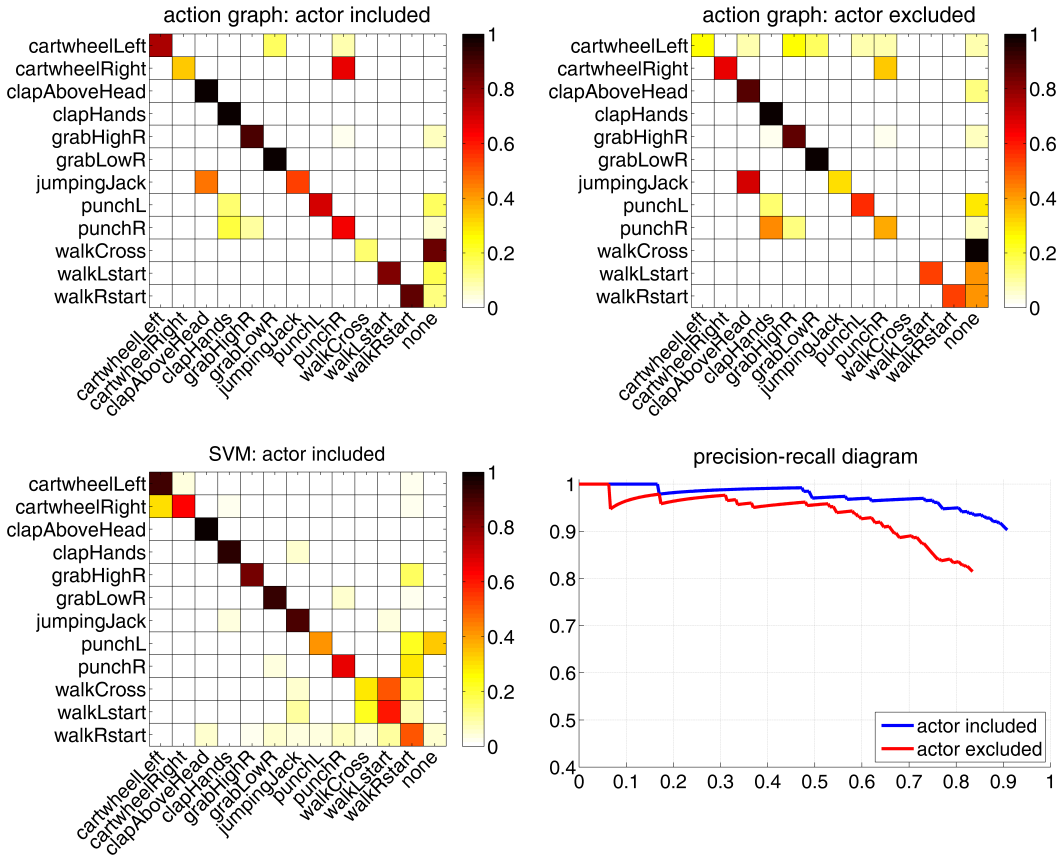


Figure 3.4: Confusion matrices for Action Graph *and SVM-based recognition methods, calculated on the HDM05 cut database using feature set $\mathcal{F}_E^{15}$ and the corresponding precision-recall diagram.*

classes (e.g., cartwheel) is larger than the variability in other classes (e.g., walk two steps). Therefore it is not possible to specify a uniform radius for all regarded motion classes.

3.5.4 Action Recognition Tests on HDM05 Motion Classes

Action recognition tests were conducted on the HDM05 cut library, which contains manually cut out motion clips that were arranged into several different classes and styles, having multiple realizations of the same motion. These motions were divided into a training set, containing 142 motions, and a test set, containing 273 motions. The confusion matrices for the two action recognition methods on this dataset using

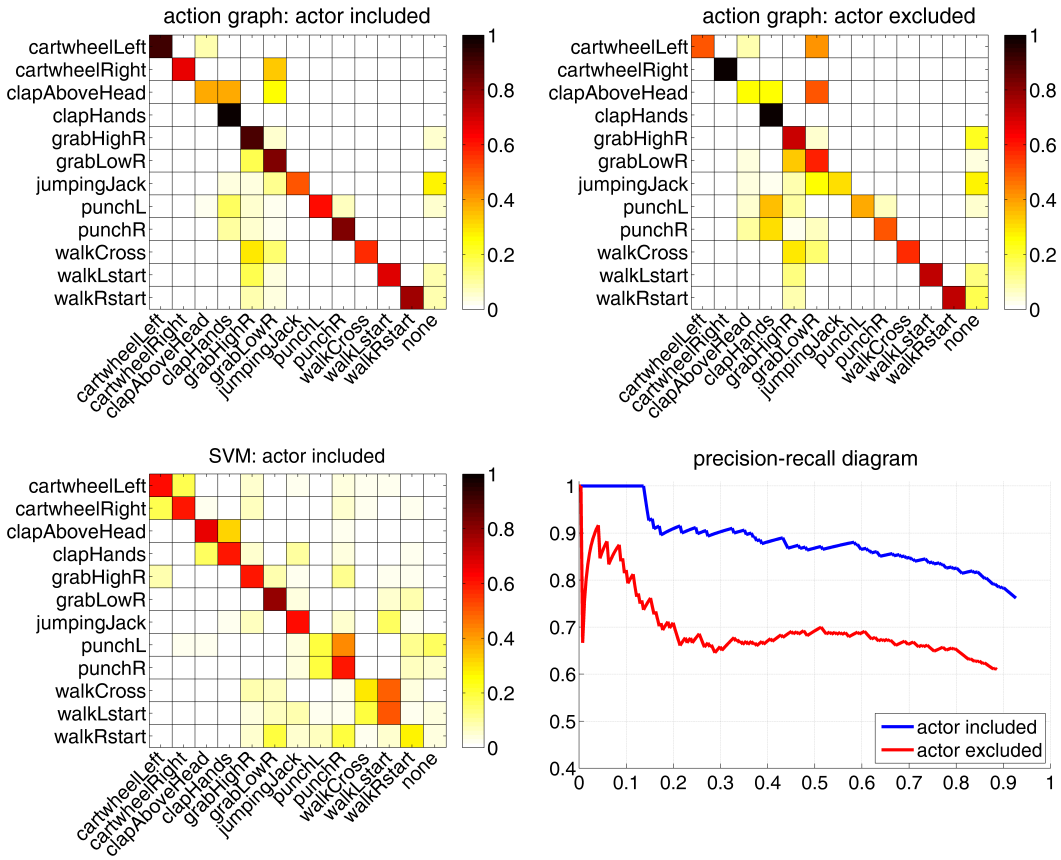


Figure 3.5: Confusion matrices for Action Graph *and SVM-based recognition methods, calculated on the HDM05 cut database using data obtained from accelerometers attached to the wrists and ankles. Also, the corresponding precision-recall diagram is shown.*

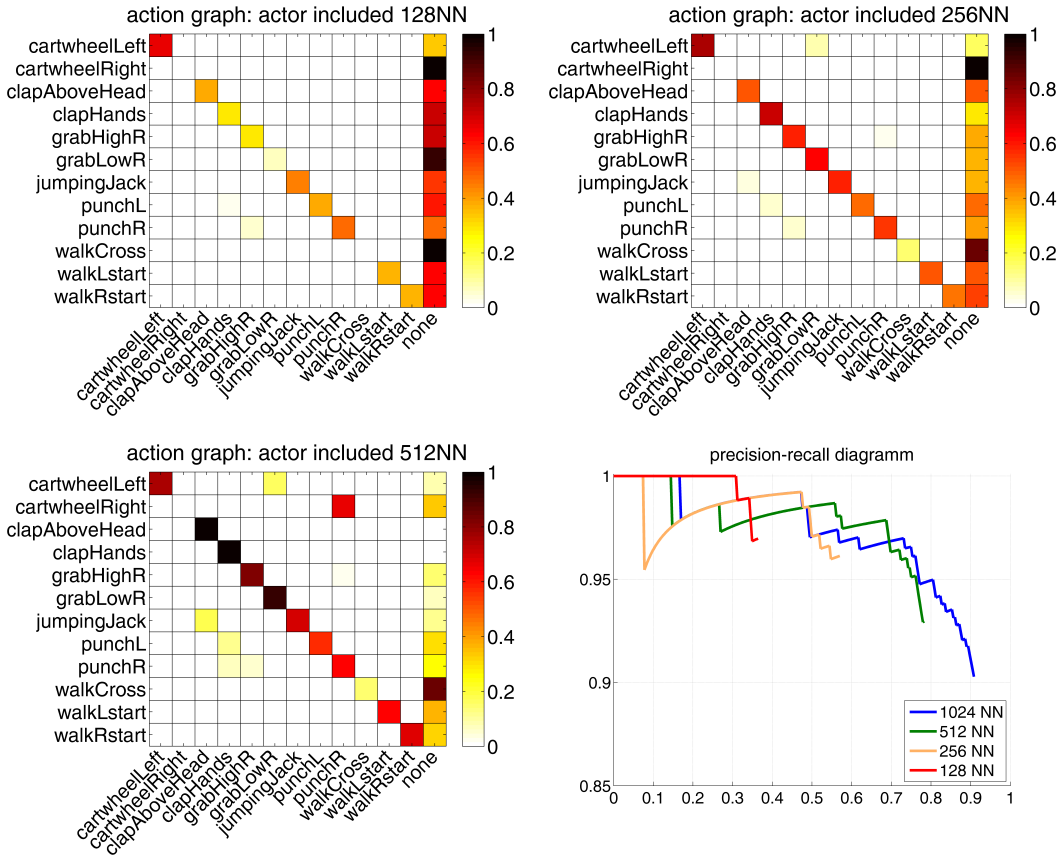


Figure 3.6: Confusion matrices for different values of the parameter k (128, 256, 512) using feature set $\mathcal{F}_E^{15}$ on the HDM05 database and the corresponding precision-recall diagram.

$k = 1024$ in the k-nearest neighborhood search can be seen in Fig. 3.4. Examining the confusion matrices shown in Fig. 3.4, the SVM-based approach shows a good performance for a pose-based approach, having a clearly visible diagonal with a few outliers, primarily confusing walking motions. The *Action Graph* shows a crisp diagonal, with only two major outliers, namely recognizing a sideways punch instead of a cartwheel starting with the right hand and recognizing a clap above the head instead of a jumping jack. However, in both cases the correct actions are sub-actions of the recognized action. Also, when visually comparing the cartwheel instances with the sideways punches, the starting phases of the cartwheels show huge similarities with the sideways punches, leading to false recognitions. This indicates that the method is broadly suitable. Inspecting the accuracy plot for the HDM05 library in Fig. 3.6, the recognition method detects 90% of actions correctly and its accuracy peaks at approximately 90% using $k = 1024$.

3.5.5 Action Recognition Tests on Judo Referee Signals

In a cross-modal scenario, skeletons extracted from a Microsoft Kinect device were used to query for similar motions in the optical motion capture database containing the Judo referee signal motions. Skeletal data obtained from this sensor contains positional data only and is of much lower quality than the optical mocap data, meaning the positional noise is much more noticable and the accuracy of the system is not on par with optical systems. Since the Microsoft Kinect delivers skeletal motion data at 30 Hz, whereas the optical motion capture system has a frame rate of 119.88 Hz, the Vicon data is downsampled to the lower rate to obtain temporal comparability.

In order to improve the probability of finding paths through the pose neighborhoods using the *Action Graph*, additional tests were run with an increased number of allowed steps (see Fig. 3.8). Interestingly, feature set $\mathcal{F}_{\mathrm{E}}^{15}$ gains in accuracy when allowing 8 step tuples and 2^{10} neighbors.

3.5.6 Action Recognition Tests on Video Data

To show that the action recognition concept easily applies to other sorts of data, this section presents an example of action recognition from video data. In order to keep

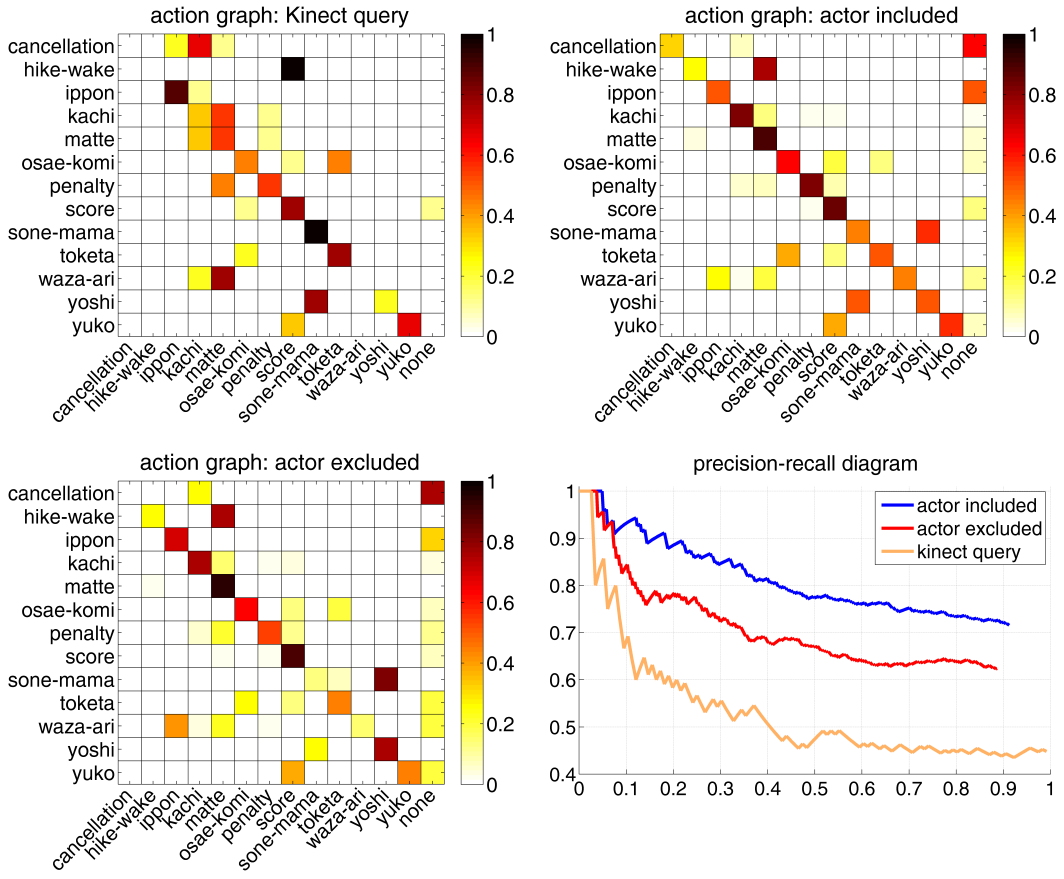


Figure 3.7: Confusion matrices for the Action Graph *based recognition method, calculated on Judo referee motions using feature set $\mathcal{F}_E^{15}$ for Kinect and V-Files (actor in- and excluded) and the corresponding precision-recall diagram.*

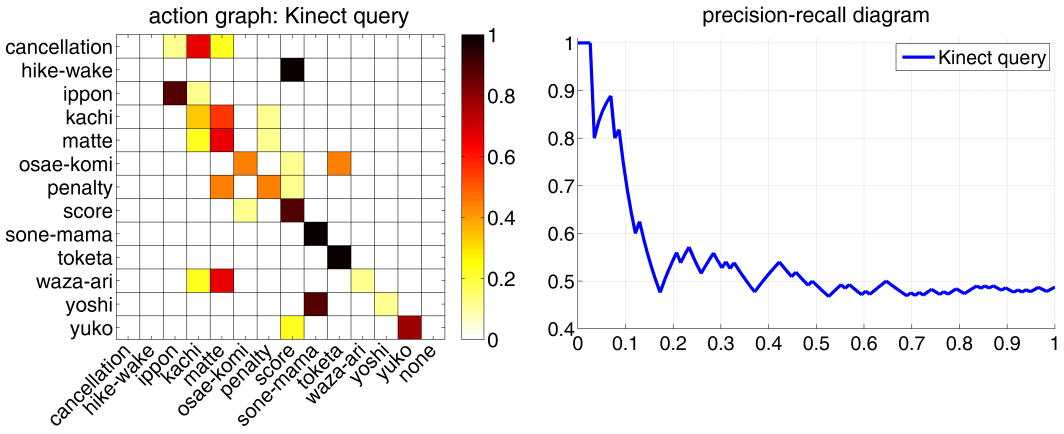


Figure 3.8: Confusion matrix and precision-recall diagram for Kinect queries in the Judo scenario where larger step sizes were used.

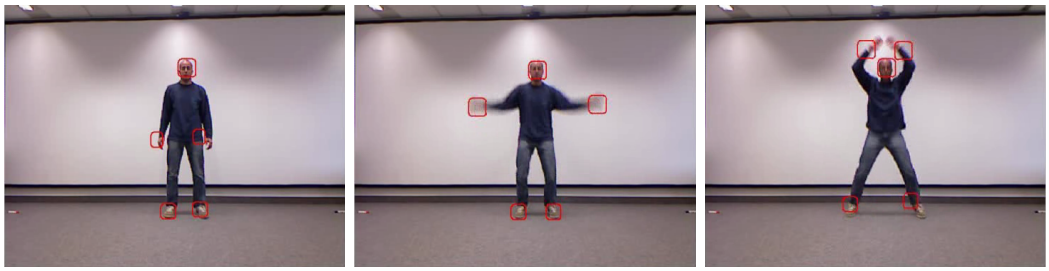


Figure 3.9: Screenshots of the video used for action recognition in Section 3.5.6 with the extracted features highlighted in red.

emphasis on the action recognition method, a simple setup is used to demonstrate the concept. To this end, the positions of hands, feet and the head are annotated in the first frame of video data and standard feature detection methods (MSCR and SURF) are used to track the relevant features used in the algorithm (see the jumping jack example in Fig. 3.9). Based on these, a ten-dimensional feature set $\mathcal{F}^{10}_{\text{E-2d}}$ is obtained consisting of five two-dimensional positions. Camera parameters are derived by incorporating knowledge about the scene and actors. Since the motion database in this example consists of three-dimensional positional data and the feature extraction from video yields two-dimensional interest points, parallel projections of all poses contained in the database were performed. To handle different viewing directions, projections were calculated from varying viewing angles in 20 degree steps. All resulting two-dimensional features were used to construct a kd-tree for knn search.

The back-projection of kd-tree indices results in database indices in the motion's original space, enabling the use of the *Action Graph* to detect the performed actions. Since the tracked features are very noisy in this case, the *Action Graph* does not return paths in all relevant cases. To alleviate this, step size conditions were adapted for this scenario to allow for steps (1,3),(3,1),(4,1) and (1,4) in addition to those previously mentioned. The results of the action recognition tests on video data can be seen in the accompanying video, contained in the supplemental material.

3.5.7 Action Recognition on Laser Range Scanner Data

Laser range scanners (LRS), like the Velodyne LRS shown in Fig. 3.10, output distance information to points in a scene. The applications of depth data range from providing information about surroundings to collision and object avoidance on autonomous vehicles in civilian [TMD+07] and military robotics applications. The devices can also be used for surveillance or mapping of areas. The most recent versions are light enough to even be mounted on small unmanned aerial vehicles (UAV).

By use of a rotating head with an array of laser emitting diodes, a laser range scanner delivers a 360 degree view of the scene around it. The distance to an object is measured by determining the time it takes for the laser light to travel to the target and back to the device. A typical indoor scene captured with a LRS can be seen in Fig. 3.11. Information about motions performed by actors in a scene captured by a LRS could be of great value in many applications. One example would be self-driving cars that could benefit from predicting pedestrian movement by analyzing the LRS data stream. Another example from a military context could be a robot following a soldier, anticipating movement and following commands by understanding full body gestures. All of these additional features come at virtually no extra cost, because the scanners are usually already mounted on the vehicles, making it a very cost effective and useful add-on.

In this section, a known and well working method for extracting feature set $\mathcal{F}_{\mathrm{E}}^{15}$ developed for depth image data [PGKT10, BMB+11] is applied to LRS data with the ultimate goal to detect the performed actions using the *Action Graph*. While depth data delivered by a Kinect or time of flight (TOF) camera usually has high frame rates as well as spatially dense data with comparably little noise, the LRS produces

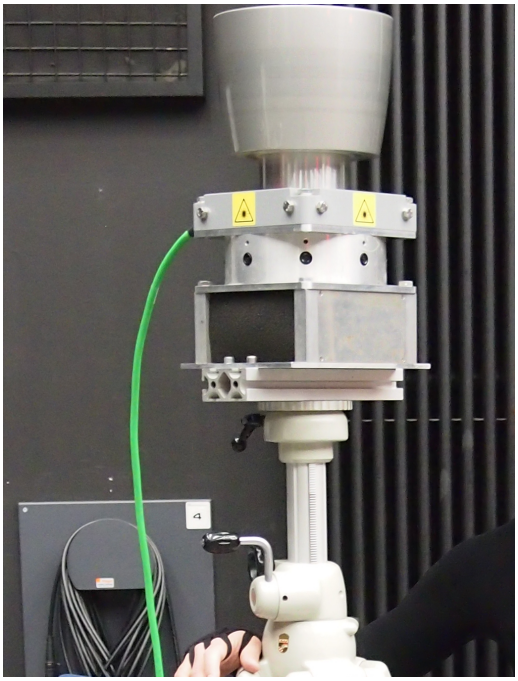


Figure 3.10: Rotating Velodyne laser range scanner.

highly noisy data at low frame rates. The advantages of the LRS are a 360 Degree field-of-view, a much greater range and applicability in outdoor environments.

Assume that for a LRS recording with $N \in \mathbb{N}$ frames the isolated actor point cloud P_f is given for every frame f. LRS poses now have to be normalized and the corresponding features of $\mathcal{F}_{\mathrm{E}}^{15}$ extracted to make them comparable to poses stored in the mocap database. Along the lines of Plagemann et al. [PGKT10], the centroids of all P_f are calculated. Then, the points belonging to the torso are estimated by checking whether they are inside a sphere with radius 0.15 m and the corresponding centroid as center point. The direction that the actor is facing can be estimated by projecting the normal of a least-squares plane fit to the torso points onto the ground plane. Similar to Baak et al. [BMB+11], a graph structure in combination with Dijkstra's Algorithm [Cor09] is used to compute the first 5 geodesic extrema, which often correspond to the end effector positions (see Figs. 3.12 and 3.13).

Unfortunately, frame overlap (double limb) artifacts originating from dynamic actor movement result in ambiguous geodesic extrema estimations (see Fig. 3.13). Due to low spatial resolution and laser diode dead zones, the distance threshold for building the local vertex neighborhoods used to build up the graph structure as in

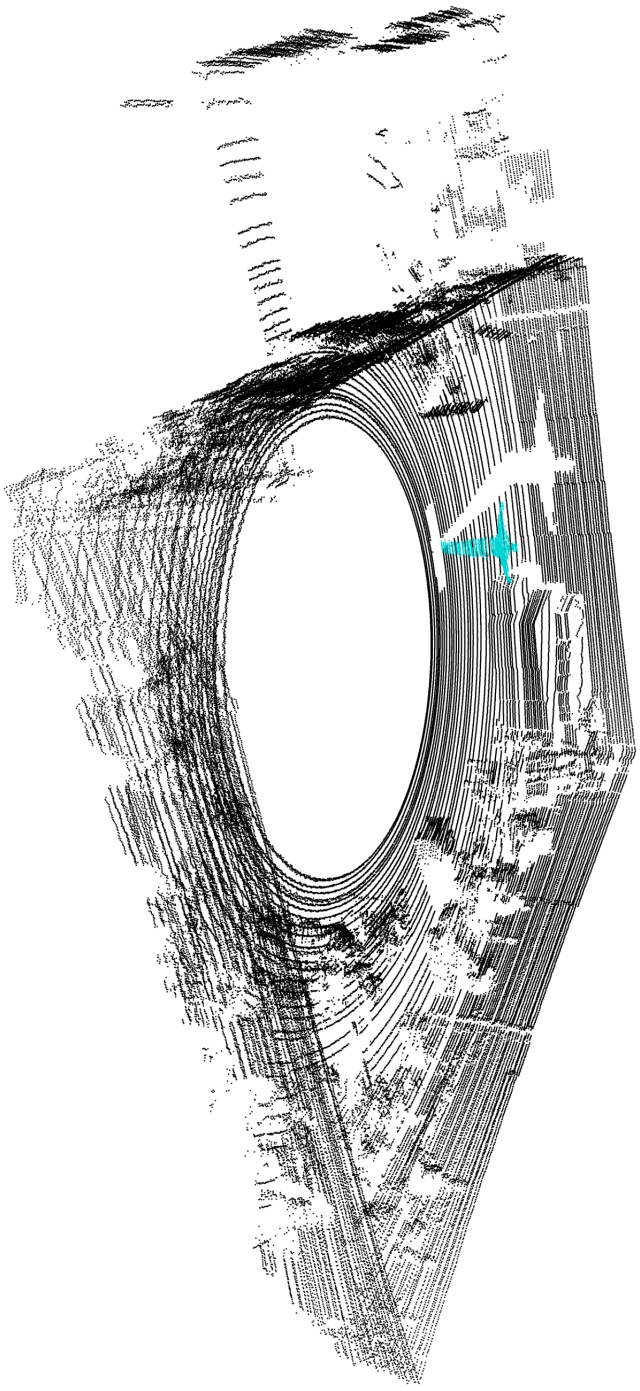


Figure 3.11: Typical frame (point cloud) of an indoor LRS recording, The actor point cloud is highlighted.

[BMB+11] needs to be a relatively large value. This in turn results in falsely merged connected components, especially in the leg area, when limbs are too near to each other (see Fig. 3.12). In addition, estimation of the viewing direction, which is used to rotationally normalize the point cloud prior to feature extraction, is also afflicted with significant errors. The frame-wise error between the estimated and ground truth viewing direction during the course of a jumping jack motion is shown in Fig. 3.14. Note that during this experiment, the actor was intentionally positioned as near as possible to the LRS. Placing the actor further away results in lower spatial resolution, deteriorating results even more. As apprehended, an action recognition test run using the extracted features resulted in no database motions similar to the performed one.

In summary, feature set $\mathcal{F}_{\mathrm{E}}^{15}$ extracted from LRS recordings using normalized point clouds in combination with the *Action Graph* does not allow action recognition, even when large step sizes are used to accommodate for the highly noisy data.

3.5.8 Comprehensive Analysis of the Results

As demonstrated by the confusion matrices in Fig. 3.4 and Fig. 3.5, the results of the above-mentioned tests show that the proposed detection method works very well on optical motion capture data and still well for accelerometer data. However, the Judo results lag far behind this good score both for motion capture data as well as data from Kinect recordings (see Fig. 3.7). In both cases this is partly due to the fact that the recorded referee motion repertoire turns out to be a challenge for the method in itself: For one, most of the gestures typical for Judo referees are fairly static and do not display the continual movement a sensible action recognition method is based on. Additionally, referee gestures with different meanings often differ only marginally, especially for the noisy Kinect data and its poorly aligned skeletons. This causes conditions to deteriorate.

Fig. 3.6 illustrates the transition of the resulting precision respectively recall for increasing choices of the number k of nearest neighbors in the action recognition test. As can be seen, the results for $k = 512$ already display satisfactorily high precision. Achieving this is obviously easy if as little recall is required. A more reliable framework forces the recall to be higher by employing a parameter $k = 1024$ although this effects in some loss of precision.

3.6 Conclusion and Future Work

This chapter examined methods to automatically detect human full body motions using motion capture data obtained from various sensor setups. This includes working with high quality optical motion capture data, skeletons output by the Microsoft Kinect within a cross-modal setup as well as sparse and noisy data obtained from accelerometers. Moreover, the method extends to features extracted from video data. In particular, the presented data-driven motion based detector was found to be superior to *Support Vector Machines* in terms of their performance.

The approach at hand is parameterized by the employed feature sets, hence will work with other capabilities. It will therefore be a matter of future work to use and to evaluate the recently proposed more robust feature sets [OCK+14] within the framework. There are certain areas which turn out to serve as fertile grounds for future work: From one point of view, the application of the fixed radius search method has revealed there is a striking amount of variation in the respective retrieved pose neighborhoods of certain queries. In particular, the gaps occurring within these neighborhoods seem noteworthy. Analyzing neighborhood variation phenomena should provide interesting new insight.

Although the presented method is already real-time capable in many scenarios, it allows for modifications to increase this capability to scenarios with many allowed time steps and large pose neighborhoods: It is clearly not necessary to create a complete graph structure for every single frame throughout the process. Working out a more efficient solution which avoids discarding previously acquired information in the spirit of Tautges et al. [TZK+11] will contribute significantly to greater efficiency. Moreover, since all significant processes involved in the method are easy to parallelize, they come with even more advantages when executed on highly parallel units. In particular, implementing the proposed techniques on a GPU seems a logical step which shall be taken in the future.

Another line of future research is the exploration of other consumer electronic devices—such as contact sensors, simple 1-or-2 axes accelerometers, altimeters, etc.—and their combinations. Although not all of these might be suitable for the current approach, many of them present promising perspectives. Especially smartphones come with an increasing variety of sensors and hence have become popular

objects of study. Combining the information from different sensors at different body locations combined with Bayesian a priori knowledge on the temporal evolution of human motions taken from databases—as the approach can be summarized—might be beneficial in this more general context as well.

Figure 3.12: Frame of a LRS recording (jumping jack) with the actor point cloud isolated. Each of the connected components of the neighboring graph is rendered in a different color. Edges connecting neigbhboring connected components are drawn as black lines while the geodesic extrema are rendered as black dots. Also, the calculated viewing direction is projected onto the ground plane. Clearly, the frame exhibits limb merging (legs), movement artifacts (arm) as well as dead zones, e.g., between waist and chest, not covered by the LRS. This leads to the situation that two of the five geodesic extrema are falsely located (knee, wrong arm).

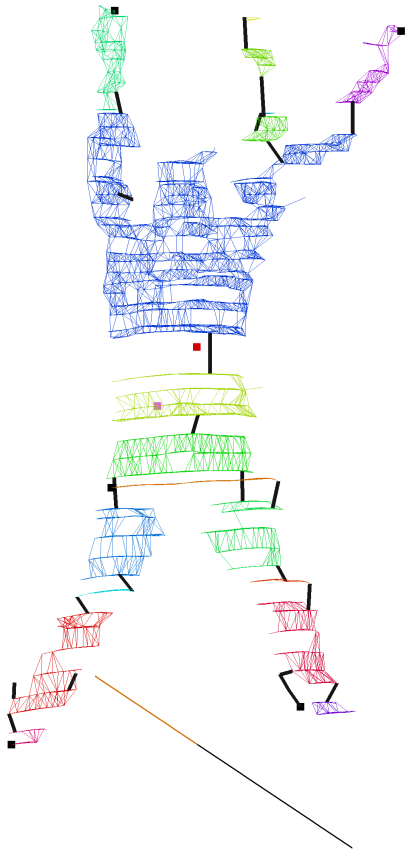


Figure 3.13: Another frame of the same LRS recording already presented in Fig. 3.12. Again, double limb movement artifacts are visible (arm). Although the orientation of the actor to the camera has not physically changed when compared to Fig. 3.12, the derived viewing direction differs significantly. Again, two of the five geodesic extrema are falsely located (waist, wrong arm).

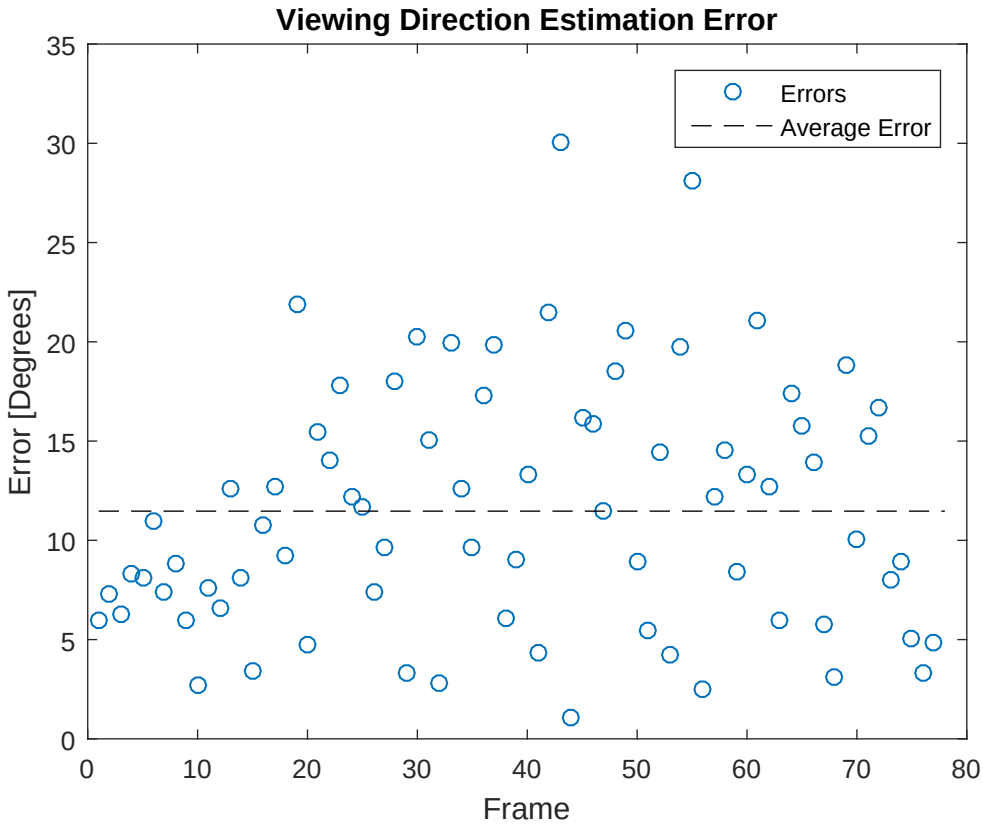


Figure 3.14: Frame-wise viewing direction estimation error in degrees when compared to the viewing direction ground truth, which is straight towards the sensor. The average error for this recording is $11.5° \pm 6.2°$

4

Pulse Detection from Video Data and its Application to Epileptic Seizure Detection and Classification

4.1 Introduction

Epilepsy is a brain disease characterized by an enduring predisposition to generate epileptic seizures affecting more than 50 million people worldwide [Wor, FAA+14]. It is characterized by spontaneously recurring seizures which can present with a large variety of symptoms, depending on the brain area that is affected by abnormally synchronous activity of neurons. For instance, patients may simply have a short and rising sensation of nausea, a tingling in an arm or a déjà-vu sensation. This seizure type is called *aura* or *simple-partial seizure* (SPS). Seizures can also impair the consciousness of patients (i.e., patients do not respond to external stimuli or are not aware of what is happening in their surroundings) and patients display stereotyped automatisms such as smacking, chewing or swallowing (called oroalimentary automatisms) or involuntary repetitive movements with their hands (called manual automatisms). This seizure type is called *complex-partial seizure* (CPS). In the worst case, the patient suffers from a *generalized tonic-clonic seizure* (GTCS) which is characterized by a stiffening of the whole body that develops into repetitive rhythmic jerking of arms and legs. Apart from these symptoms, epileptic seizures frequently lead to alterations of autonomic body functions (i.e., breathing, heart activity, sweating, and others). For instance, 80-90 % of the seizures are associated

with an increase of heart rate [LSL+03].

The time during the seizure is called *ictal* (derived from the latin word "ictus"), the time period before the seizure-onset is labelled *preictal* and the time period after seizure-offset *postictal*, respectively. To control and suppress occurrence of seizures, people with epilepsy take anticonvulsant drugs on a daily basis. The therapeutic effect is commonly evaluated by seizure diaries in which patients, relatives or caretakers report the seizures. A fundamental problem is, however, that self-reported seizure diaries are notoriously incorrect, because patients are not aware of their seizures or they forget them (amongst others because seizure also perturb brain structures that are important for memory). About 50 % of the seizures are incorrectly documented (i.e., not reported by the patients; [HPE07]), challenging the validity of seizure diaries for evaluating the therapeutical efficacy. Therefore, it would be very relevant for both daily clinical practice as well as pharmaceutical drug trials to develop an automatic method to count seizures. Novel mobile health technologies (e.g., smartphones, fitness-trackers) allow measurement of heart rate and body movements, possibly improving the quality of seizure documentation.

A further challenge in the field of epilepsy is that about 18 % of patients who were given the diagnosis of epilepsy do actually not suffer from epilepsy, but other medical conditions [XNM+16]. For instance, syncopes (sudden loss of consciousness along with body jerks) or psychogenic non-epileptic attacks can be mistaken for epileptic seizures. The false diagnosis is often due to the fact that these episodes happen in the absence of a qualified medical doctor and that the proper diagnosis is commonly based on the report of laymen. However, the increasing number of home-videos taken with smartphones may help to make the correct diagnosis. In this context, it is important to note that heart rate was shown to provide some help to support the distinction between these three entities [RPMD12]. Therefore, in addition to the visible symptoms of the episode of unknown nature seen on the video, it would be very useful to have additional information on autonomic features such as heart rate.

In this chapter, it is asked whether information on heart rate can be extracted from video films of epileptic seizures. A proof-of-concept under controlled conditions is provided (i.e., simultaneous information on the heart rate as assessed by established methods), limitations of the method are explored and it is tested on examples

of videos originally taken with smartphones by relatives of epilepsy patients.

4.2 Overview

This chapter is organized as follows:
In Section 4.3, previous work related to video pulse detection methods as well as their application in medical environments is presented.
Following in Section 4.4, a method for determining the heart rate from electrocardiography (ECG) signals is discussed and demonstrated on examples.
The method used for detecting the heart rate from a video of the human face is presented in Section 4.5.
Section 4.6 evaluates the applicability and limitations of the video pulse detection method in the context of epilepsy and contains results of selected usage scenarios.
The chapter closes in Section 4.7 with a conclusion and suggestions for future work.

4.3 Related Work

Photoplethysmography (PPG), i.e., an optically obtained plethysmogram, is a way to optically measure the volumetric changes of an organ and was first introduced by Hertzman and Spealman [HS37] in 1937. In daily clinical routine, PPG measurement devices are usually attached to a finger and deliver heart rate and oxygen saturation derived from the raw PPG signal, which is a reflection of variations in blood volume. This is based on the principle that blood is more light absorbent than surrounding tissue, thus affecting transmission and reflectance of irradiated light.

Scully et al. [SLM+12] demonstrate that no specialized clinical equipment is necessary by showing that a fingertip placed directly on a smartphone camera is enough to record a PPG signal.

Remote measurement of physiological parameters such as heart rate and respiration rate as well as oxygen saturation has seen increased interest over the last years, starting with Wieringa et al. [WMvdS05] and Zheng et al. [ZHCS08]. Verkruysse et al. [VSN08] were the first to analyze remote PPG using videos of faces filmed with a consumer photo camera in video mode in ambient lighting. Since then, the method has been improved by several researchers, incorporating tracking of the

subject's face and increasing its robustness against subject movements and changes in scene illumination [PMP10, PMP11, LCZP14].

Recently, Przybylo et al. [PKJA16] investigated the dependence on lighting conditions and camera performance in videoplethysmography by evaluating the influence of the lighting spectrum, camera frame rate and video compression.

In a clinical context, the video pulse detection method was applied by Tarassenko et al. [TVG+14], who estimated heart rate and respiratory rate from long-term videos of patients undergoing haemodialysis in the Oxford Kidney Unit.

In another line of research, Balakrishnan et al. [BDG13] exploit subtle head oscillations that accompany the cardiac cycle to derive the heart rate.

In *Eulerian video magnification* by Wu et al. [WRS+12] and its follow-up work *Phase-based video motion processing* by Wadhwa et al. [WRDF13], the authors describe methods to both reveal and emphasize subtle color and motion changes in video recordings.

4.4 Heart Rate from Electrocardiography Signals

In Electrocardiography (ECG), the electrical activity of the heart is recorded over time by a set of electrodes that are placed on the skin. The electrical stimulus that causes the heart to contract produces a potential variation on the skin and can be measured by the electrodes. This noninvasive medical procedure produces a graph of voltage versus time which is called an electrocardiogram (also abbreviated as ECG). An example can be seen in Fig. 4.1.

Heart rates are usually given in heart beats per minute [bpm] and can be calculated from the raw ECG signal by measuring how many times the heart goes through a cardiac cycle in 60 seconds. The *QRS complex* consists of three deflections in the ECG waveform and reflects the depolarization of the right and left ventricles and is the most prominent feature of the human ECG. On a near perfect ECG signal as in Fig. 4.1, the easiest and most frequently used manual method employed by physicians is to directly count the number of R-peaks, i.e., the most prominent feature of the QRS complex. Unfortunately, ECG signals often times exhibit all kinds of noise, e.g., power line noise or electrical noise originating from nearby muscle movement, possibly different amplitude levels throughout the recording or even in-

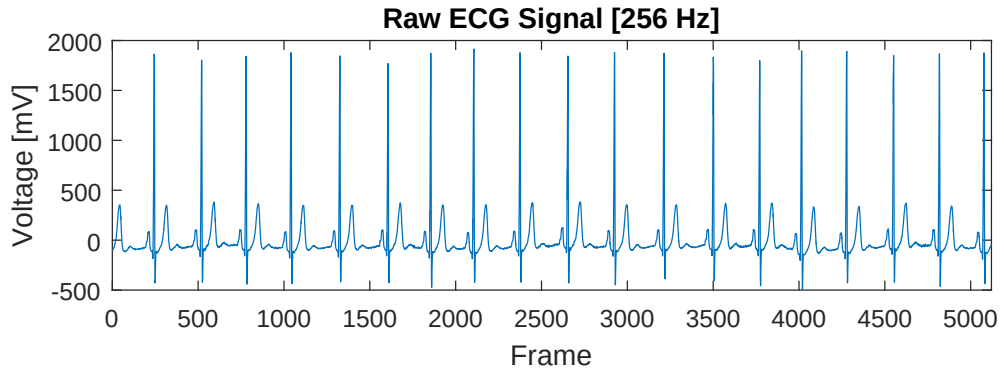


Figure 4.1: Example of an easy to process raw ECG signal (20 s) recorded by a clinical ECG system.

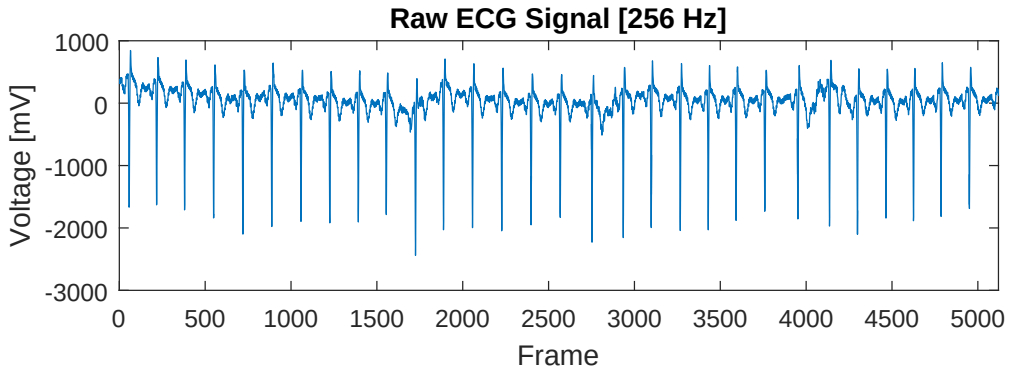


Figure 4.2: Example of a raw ECG signal (20 s) that needs further processing before R-peaks can be detected

verted R-peaks (see Fig. 4.2). This means that in order to automatically detect the R-peaks, some preprocessing has to be done on the original signal.

The performance of ECG analyzers is commonly evaluated using a database of ECG signals annotated by medical experts, e.g., the *MIT-BIH arrhythmia database* [MM01, GAG+00]. Since, in this work, solely the timestamps of the R-peaks are of interest, a simple peak detection algorithm is described that allows quick, accurate and robust R-peak detection.

The first step in preparing the raw signal $\boldsymbol{X} := x_i \in \mathbb{R}^n$ of length n for R-peak detection is centering the data by subtracting the mean:

$$\boldsymbol{X} = x_i \leftarrow x_i - \frac{1}{n}\sum_{k=1}^{n} x_k.$$

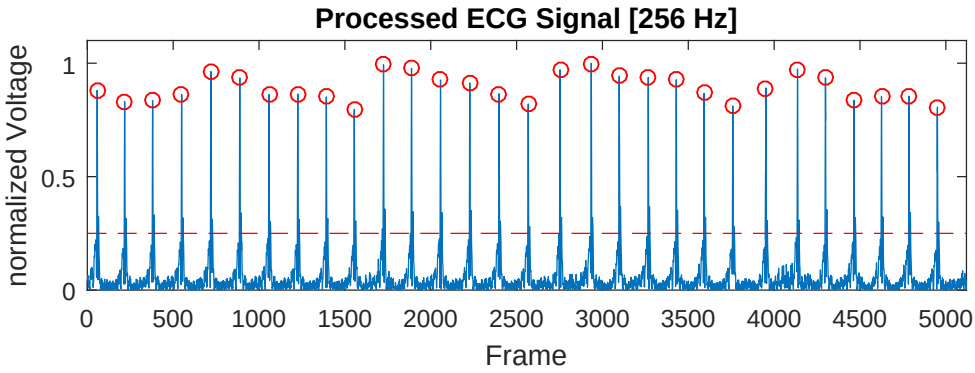


Figure 4.3: Processed ECG signal from Fig. 4.2. Detected peaks (red circles) above 0.25 normalized volts (dashed red line) and a minimum distance of 256/4 frames represent the R-peaks.

Then, the signal is detrended to remove any linear trends. This is done by computing the least-squares fit of a straight line to the data and subtracting the resulting function from the data. After detrending, the absolute value of the signal is calculated to deal with the possibility of inverted R-peaks:

$$\boldsymbol{X} = x_i \leftarrow |x_i|.$$

Finally, the signal is normalized to within the interval $[0, 1]$ to allow a constant peak detection threshold regardless of signal amplitude. The result of processing the signal from Fig 4.2 can be seen in Fig. 4.3. R-peak detection can now be accomplished by searching for all peaks in the signal that have a minimum height of 0.25 normalized volts and a minimum peak distance of the signal's sampling rate divided by 4, resulting in inter-R-peak distances (RR intervals) no shorter than one-fourth of a second, i.e., 240 bpm.

Let $\boldsymbol{RR} = r_j \in \mathbb{R}^m$ be the m RR intervals calculated from signal $\boldsymbol{X}$, represented in milliseconds. Now, with the average RR interval duration

$$R = \overline{\boldsymbol{RR}},$$

the heart rate HR in beats per minute is calculated by

$$HR = 60\,000\,\text{ms}/R.$$

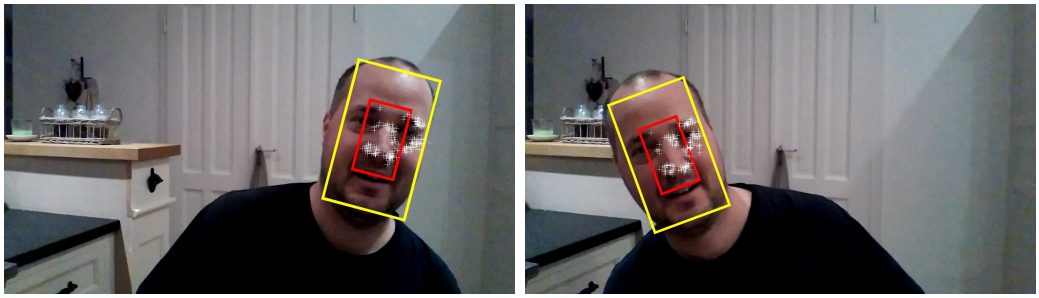


Figure 4.4: Tracking example showing how a bounding box initialized in the first frame is tracked in two frames of the video sequence. The screenshots show the tracking points (white), the transformed bounding box (yellow) and the shrunk bounding box used by the video pulse detection method (red).

4.5 Pulse Detection from Video Data

In this work, the heart rate of a human subject is determined by analyzing the subtle changes in color of a portion of the facial skin, as recorded by a video camera.

Assume that the input video consists of n frames f_1 to f_n. To keep track of the face, even during body movements or camera shake, the subject's face or a portion thereof is annotated in f_1 for further tracking throughout the rest of the video. This can be done manually by specifying a bounding box around the region of interest (ROI) or automatically by means of face detection algorithms [VJ01, VJ04]. This ROI serves as the starting point for further tracking using the KLT algorithm [LK81, TK91], that tracks a set of feature points from one frame to the next and calculates the required transformation (see Fig. 4.4).

The initially tracked region usually contains parts of the background and hair of the subject to allow robust tracking of the face. In order for the pulse detection algorithm to work correctly, these areas as well as the changes in color resulting from movements in the facial region e.g, chewing, talking, blinking, must be removed as much as possible before further processing. Although very sophisticated methods for removing these artefacts using information about face topology exist [LCZP14], a simple and effective way is to shrink the tracked ROI's dimensions by constant scaling factors $s_x, s_y \in (0, 1]$, as presented in [PMP10].

After tracking, assume that for every frame $f_i, i \in [1, n]$, the tracked ROI is

specified by a bounding box represented as tuple

$$ROI_i = (x, y, a, b) \in \mathbb{R}^4,$$

where x and y describe the two-dimensional position of the lower left corner and a and b represent the width and height of the ROI, respectively. The calculation of the shrunk bounding box that is centered in the original bounding box is as follows:

$$ROI_i^{shrunk} = (x + (a - \frac{a \cdot s_x}{2}), y + (b - \frac{b \cdot s_y}{2}), a \cdot s_x, b \cdot s_y).$$

This work uses $s_x = s_y = 0.5$ for all examples, because unlike the controlled environment in [PMP10] where $s_x = 0.6$ and $s_y = 1$ were used, the orientation of the subject's face in the first frame of video is not given as upright in this work. Another solution would be to use a non axis aligned bounding box to initialize the tracking, but the results show that this is unnecessary overhead.

In previous works like [VSN08], the authors found that the green color channel carries the most information with respect to pulse detection applications. Therefore, after the shrunk ROI is calculated over all n frames, the mean green value $\bar{g}_i$ of the shrunk ROI's pixels is calculated for every frame f_i. An example of the resulting signal calculated on a real video recording can be seen in Fig. 4.5. The pulse detection method determines the heart rate by converting a part of the mean green channel signal of a certain window width w into the frequency domain using a fast fourier transform. After the subsignal is selected, any linear trend is removed by detrending. Then, the signal is smoothed using a centered moving average filter with span 3. In order to limit the regarded pulse frequencies to those occuring in the human body, the signal is bandpass filtered with low and high cutoff frequencies of 0.7 Hz and 4 Hz (42-240 bpm), respectively. Finally, the signal is zero padded ([VSN08]) to 120 s to allow a finer frequency discretization. The average heart rate during the time the data of window w was taken from is assumed to be the most powerful frequency of Welch's power spectral density estimation [Wel67] (see Fig. 4.7). The choice of $w = 16$ s for all presented examples is explained in Section 4.6.3.

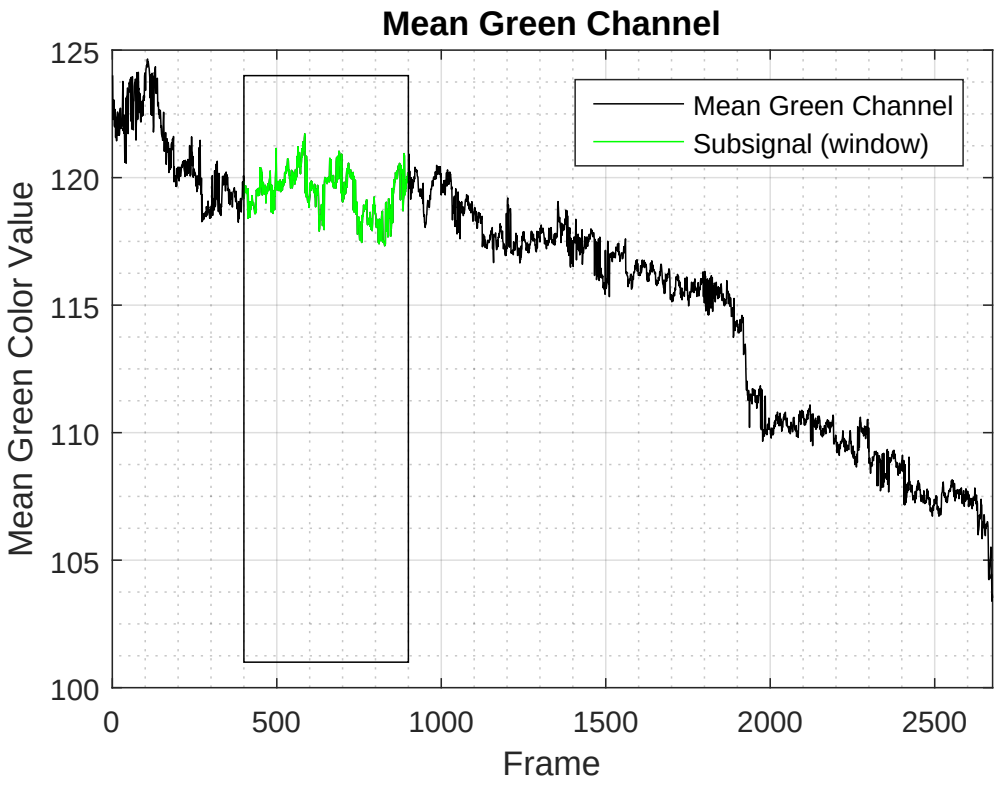


Figure 4.5: Mean green channel signal calculated from the shrunk ROIs.

4.6 Results

4.6.1 Overview

For evaluating the method presented in Section 4.5 and to determine its usefulness in the context of epilepsy, the heart rates derived from video data are compared with average heart rates calculated from simultaneously recorded clinical ECG data (see Section 4.4) using the same window width as the video pulse detection. With the exception of Sections 4.6.6 and 4.6.7, all videos were taken of patients undergoing presurgical assessment at the Department of Epileptology at the University of Bonn.

Beginning with a video pulse detection example in Section 4.6.2, a typical result of the video pulse detection method is shown and discussed.

In Section 4.6.3, the accuracy that the method is capable of is determined by detecting the heart rates of 10 subjects during an interictal phase, i.e., phases with

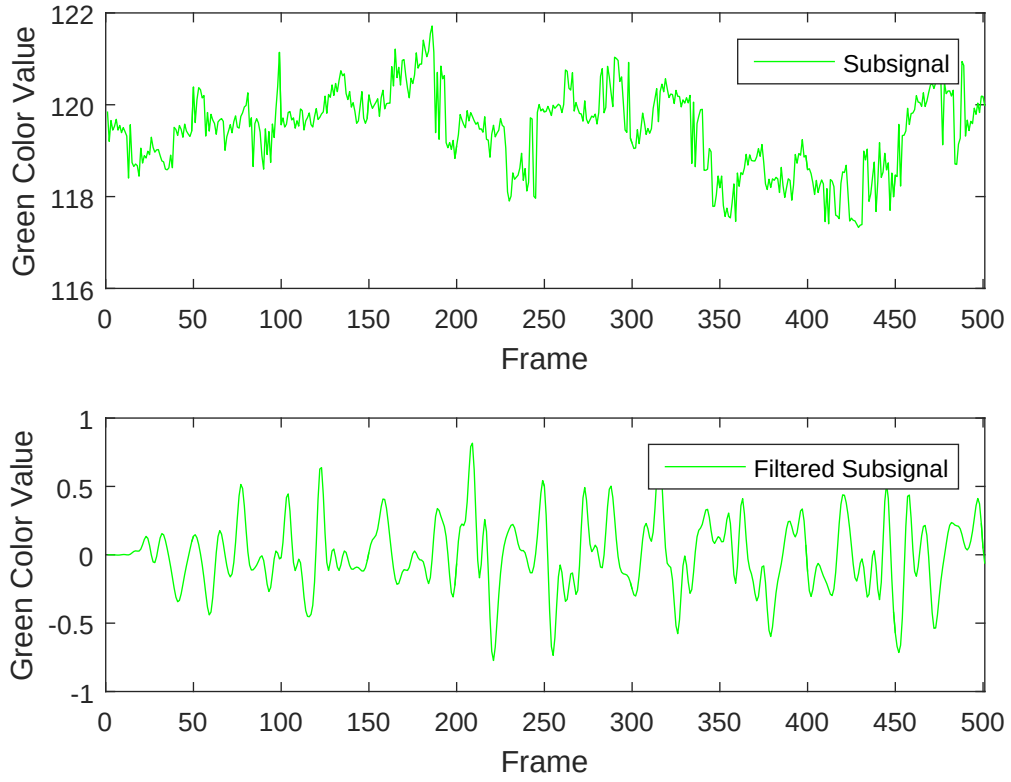


Figure 4.6:
Top: Unfiltered subsignal extracted from window w the signal in Fig. 4.5
Bottom: Detrended, smoothed and bandpass filtered subsignal

no seizure. Usually, these phases feature patients lying in bed reading, sleeping or watching TV, i.e., phases with very little or no movement which could possibly negatively influence the detection results. Also, using the results of this section, a fixed value for the window width w is determined that is used for the rest of the evaluation.

Then, in Section 4.6.4, the method is used to detect the heart rates in 10 ictal scenarios by analyzing and comparing the seizures' immediate preictal and postictal phases.

During nighttime, clinical video/EEG monitoring of patients is done with infrared cameras. The performance and applicability of the video pulse detection method on this type of video is evaluated in Section 4.6.5.

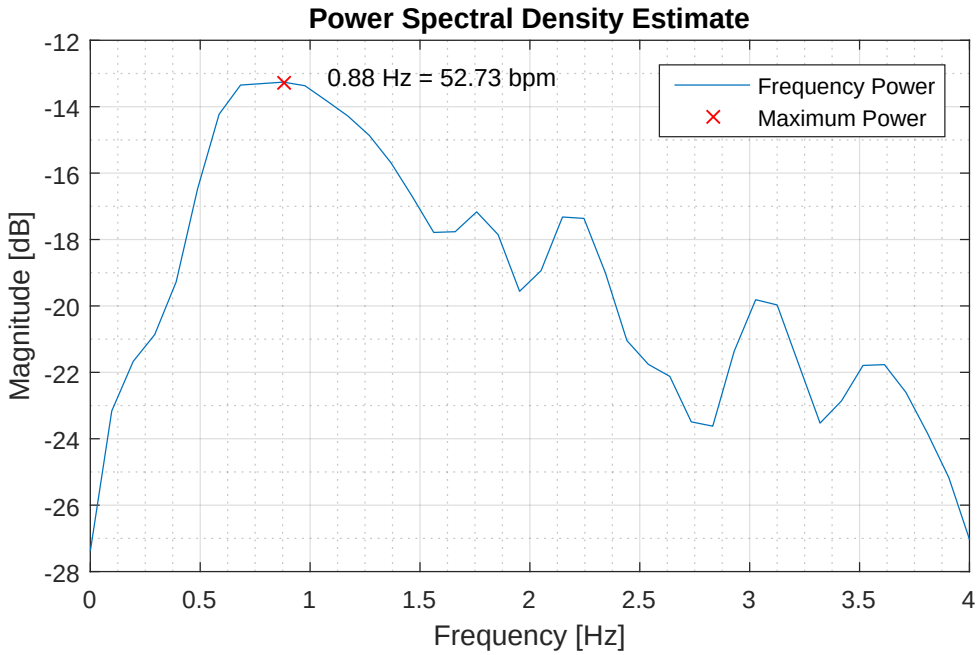


Figure 4.7: Using Welch's method [Wel67] to estimate the power spectral density of the signal shown in Fig. 4.6 (bottom). The frequency with maximum power (0.88 Hz in this example) is assumed to be the average heart rate during window w.

Often times, patients have smartphone video recordings of their seizures, either self recorded or filmed by a family member. An example of the analysis of two recordings is presented in Section 4.6.6.

Finally, the influence of makeup on the detection result are discussed in Section 4.6.7.

4.6.2 Pulse Detection Example

A typical run of the method described in Section 4.5 can be seen in Fig. 4.8. Here, the results of detecting the heart rate of approximately 3.8 minutes of video data during the preictal, ictal and postictal phases of an epileptic seizure are shown and compared to the ECG ground truth heart rate.

Interestingly, all of the data points of the heart rates detected from video lie on discrete plateaus. This is a direct consequence of transforming the signal into the frequency domain using a fast fourier transform. An important property of the FFT is that, with a video sampling rate $f_s = 25\,\text{Hz}$ and a window size $w = 16\,\text{s} =$

400 samples, a maximum frequency resolution of

$$\frac{f_s}{w} = \frac{25\,\text{Hz}}{400\,\text{samples}} = 0.0625\,\text{Hz} = 3.75\,\text{bpm} \tag{4.1}$$

can be obtained [Sha49]. The obtainable frequency resolution is reduced even further by using Welch's method for power spectral density estimation, which reduces noise at the cost of frequency resolution. By padding the signal with zeros (see Section 4.5), a finer frequency discretization is achieved which, in this application, allows better parameter estimation of the underlying signal.

4.6.3 Interictal Video Pulse Detection

In this section, the heart rates of 10 subjects (7 female, age 27.7 ± 13.2 years) are analyzed with the video pulse detection method during interictal phases. In these phases, patients and doctors wait for a seizure to happen, so that it can be recorded by video/EEG for further analysis. In between seizures, patients usually spend their time lying in bed reading books or watching television, which provides a near perfect scenario for video pulse detection.

For each sample, the subject's pulse was determined from 30 to 60 s of video data with window w as a parameter, which was varied from 4 s to 20 s in 1 s steps. Then, the average reconstruction error, i.e., the average distance to the clinical ECG ground truth heart rate, was calculated along with its standard deviation. The results are shown in Fig. 4.9. From this data, a window size of 16 s was chosen for subsequent evaluations, because it provides the shortest window with almost maximum accuracy.

Figure 4.10 shows an example of the video pulse detection during an interictal phase using a window of 16 s. The plots for all other subjects can be found in Appendix A.1.

In Table 4.1, the average video and ground truth heart rates as well as the detection errors of the 10 interictal heart rate detection experiments are shown. Figure 4.11 visualizes the interictal detection results and provides an overview of the achievable detection accuracy, showing a very good average reconstruction error of 0.84 ± 0.62 bpm calculated over all 10 samples.

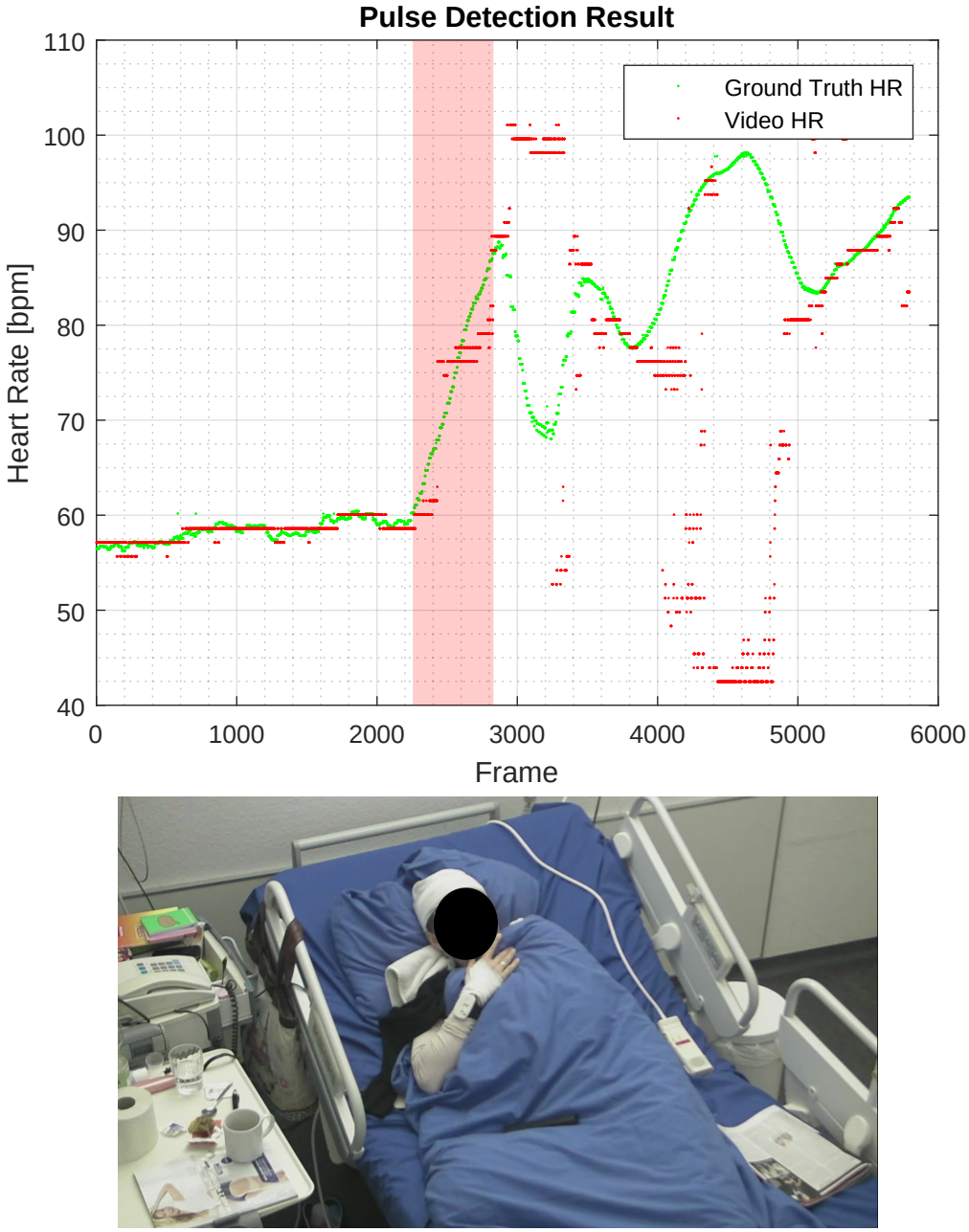


Figure 4.8: Pulse detection results and screenshot of 228 s of video (@25 fps) using a window of 16 s. The ictal phase of the epileptic seizure, annotated from EEG data, is highlighted in red. Although the subject lies in bed and is not involved in any physical activity, the ground truth and video detected heart rates rise abruptly at around frame 2200, indicating the potential beginning of an epileptic seizure. At around frame 2800, possibly as a secondary effect of the epileptic seizure, the subject quickly blinks her eyes for a prolonged time period, fooling the method into detecting a quicker heart beat. At around frame 3800, the patient turns her head to look at the nurse entering the room, which introduces high powers in low frequencies of the power spectral density estimation due to a different illumination of the face. Video pulse detection then returns to normal at around frame 4900, when these low frequencies have left the 16 s window.

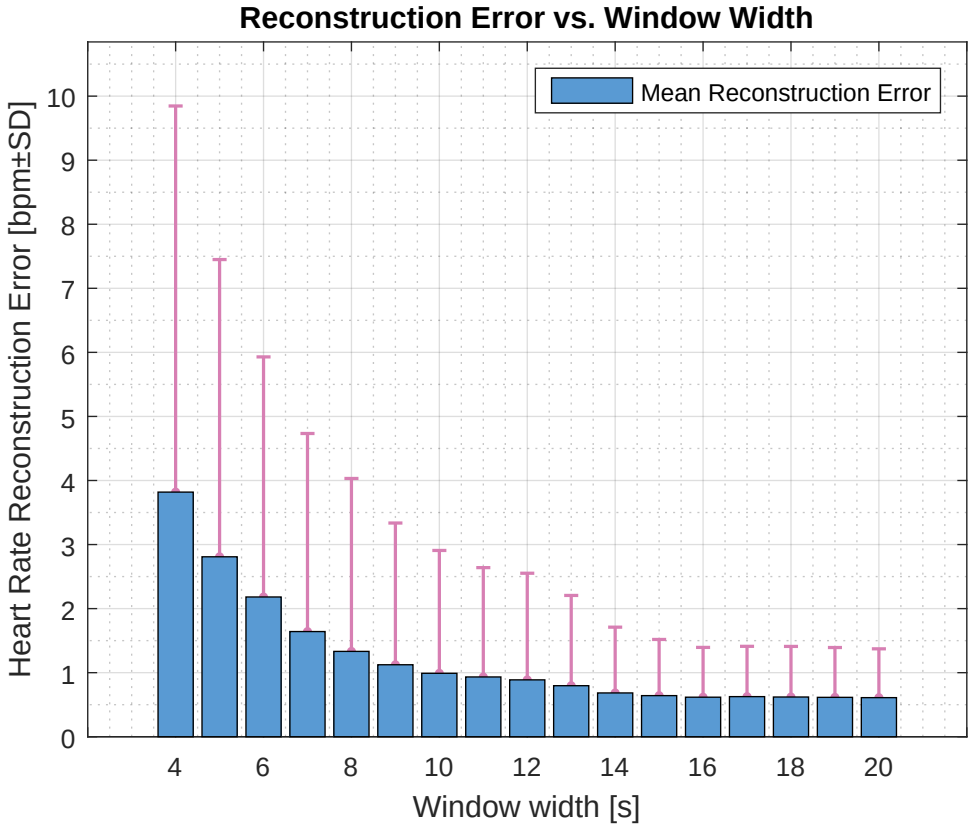


Figure 4.9: Influence of reconstruction error on chosen window width w.

4.6.4 Ictal Video Pulse Detection

In order to determine the suitability of the presented method for epileptic seizure detection, the videos of 10 patients (6 female, age 38 ± 14.62 years) with epileptic seizures were analyzed. The ictal phases of these seizures were annotated using the clinical EEG by an experienced expert in the field of epilepsy. In most cases, excessive movement during the ictal phase results in too much illuminance variation and face tracking problems, for which reason video detection was split into two intervals, using data from the immediate preictal and postictal phases, respectively. Plots of a complex partial seizure as well as a generalized tonic clonic seizure can be seen in Fig. 4.12. The plots for all other subjects can be found in Appendix A.2.

To check if the heart rate rose above a predefined threshold indicating an epileptic seizure, the heart rate detected from the preictal phase HR_{pre} is used as the baseline

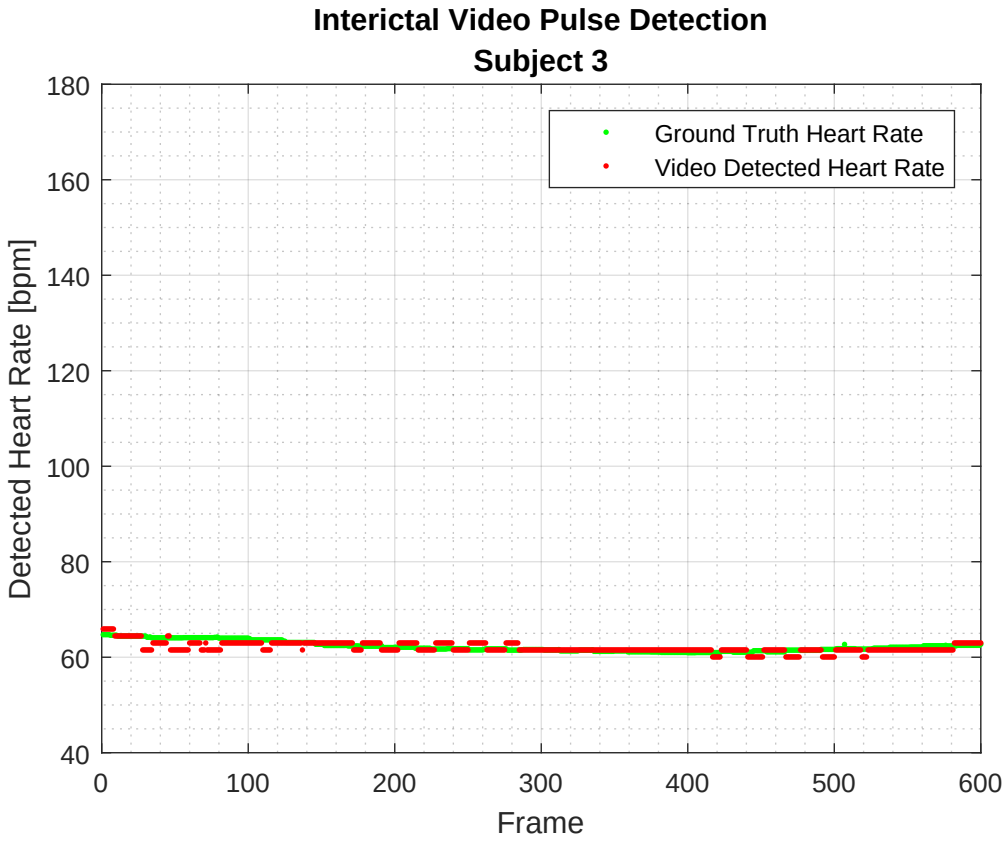


Figure 4.10: Interictal video pulse detection example

heart rate and is combined with the heart rate from the postictal phase HR_{post} to form the heart rate ratio

$$HR_{ratio} = \frac{HR_{post}}{HR_{pre}}. \tag{4.2}$$

The detection results are summarized in Table 4.2 and Fig. 4.13. Clearly, in 90 % of the cases, a rise in heart rate is detectable by the method, indicating the possibility of an epileptic seizure. In the case of the single exception (Subject 4, $HR_{ratio} = 1.03$), it was not possible to find a large enough interval of video frames without patient movement close to seizure offset. In this instance, at the earliest possible postictal video pulse detection time, heart rate had already dropped to preictal baseline again.

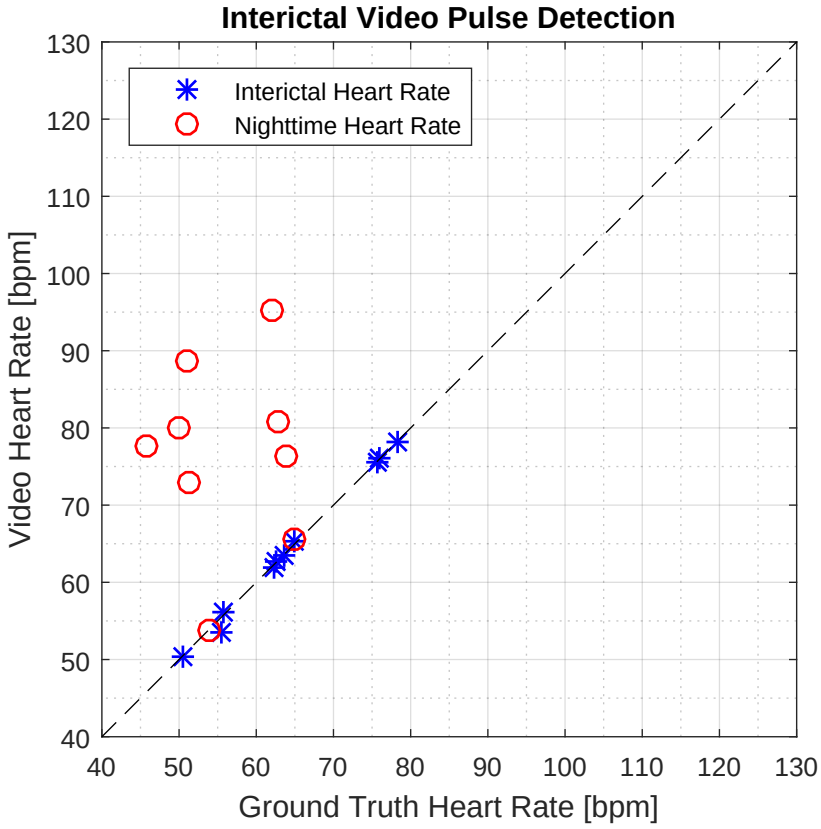


Figure 4.11: Interictal and nighttime video pulse detection results.

4.6.5 Nighttime Video Pulse Detection

Epileptic seizures often happen when patients are asleep. This is why, during clinical video/EEG monitoring, video recording is switched to an infrared camera when the monitoring room is insufficiently illuminated for regular video.

This section evaluates the performance of the video pulse detection method on nighttime recordings of 10 patients (7 female, age 27.7 ± 13.2 years).

The clinic's infrared camera outputs a RGB grayscale image, i.e., $r = g = b$ for each pixel in the frame, so the algorithm can be applied to this data without any changes. The results are summarized in Table 4.3 and Fig. 4.11. A typical pulse reconstruction example is presented in Fig. 4.14. The plots for all other subjects can be found in Appendix A.3. Unfortunately, the results show that infrared light reflection and transmission from the subjects' faces carries little to no information

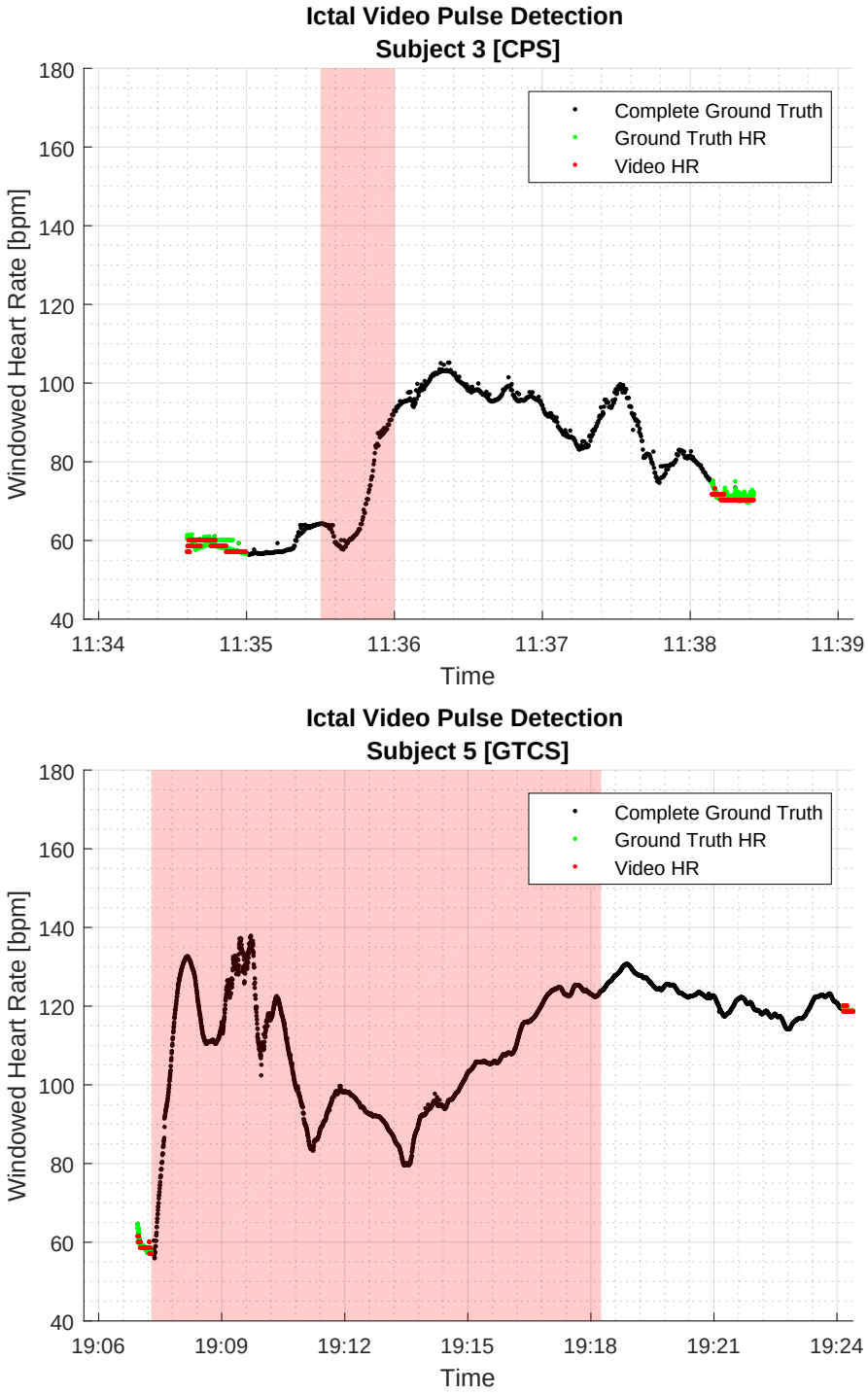


Figure 4.12: Examples of pre- and postictal video pulse detection. The ictal phases are annotated in red from EEG data. In contrast to the CPS example (top), the GTCS example (bottom) shows a higher rise in heart rate at seizure onset, which takes longer to return to the preictal baseline after seizure offset. As expected, the heart rate ratio, i.e. the factor by which the postictal heart rate differs from the preictal baseline, is much higher in the GTCS (2.0) than the CPS (1.22) case, also taking into account the temporal distance between pre- and postictal measurements.

Table 4.1: Interictal video heart rate (HR) and ECG measured ground truth heart rate (GT) along with detection errors using 30 to 60 s *of video for each sample.*

	Detected HR [bpm]		
Subject	Video	GT	Error [$bpm \pm SD$]
1	53.6	55.6	2.58 ± 2.12
2	63.4	63.7	0.83 ± 0.68
3	62.0	62.2	0.71 ± 0.64
4	56.1	55.9	0.58 ± 1.82
5	75.5	75.6	0.49 ± 1.17
6	50.3	50.4	0.66 ± 2.07
7	76.0	75.8	0.64 ± 0.49
8	62.7	62.6	0.60 ± 1.64
9	78.1	78.3	0.74 ± 0.46
10	65.2	65.0	0.53 ± 1.50

to reconstruct the heart rates, with the error averaging at 24.47 ± 15.67 bpm over all 10 samples. The significant noise found in all nighttime videos might also contribute to the high error. Interestingly, the nighttime video pulse detection of Subject 8 and Subject 5 still produced very good (0.42 ± 1.91 bpm) and acceptable (4.08 ± 10.83 bpm) results, respectively. Also, all wrongly detected heart rates lie well above their respective ground truth heart rates.

In another experiment, the videos from the daytime interictal recordings (see Section 4.6.3) were converted to grayscale by eliminating the hue and saturation information while retaining the luminance. Then, the video pulse detection algorithm was run on the modified videos. In this case, only marginal differences to the results calculated on the original RGB videos (see Fig. 4.11) were observed.

4.6.6 Video Pulse Detection from Smartphone Videos

Often times, patients come to see a doctor for help and bring a self recorded video of what they think was an epileptic seizure. These videos are either filmed by the patients themselves or by a family member or friend. As described in Section 4.1, seizures do not always have to be generalized tonic clonic seizures that are visually distinguishable from everyday life due to excessive movement and loss of control.

Table 4.2: Heart rate (HR) detection results of the preictal and postictal phases of 10 seizures, along with ground truth (GT) heart rates calculated from ECG measurements, pre-/postictal HR ratios, lengths of the respective detection intervals as well as the temporal distance between pre- and postictal video pulse detection measurements.

	Detected HR [bpm]										
	Preictal			**Postictal**			**HR Ratio**		**Interval Length [s]**		**Distance**
Type	**Video**	**GT**	**Error**	**Video**	**GT**	**Error**	**Video**	**GT**	**Preictal**	**Postictal**	**Prepost [s]**
GTCS	63.8	63.9	0.54 ± 0.42	73.5	73.2	0.60 ± 0.53	1.15	1.15	60.0	19.7	917.4
CPS	62.6	62.6	0.71 ± 0.59	82.3	82.6	0.60 ± 0.10	1.31	1.32	43.0	22.0	219.0
CPS	58.6	58.5	0.82 ± 0.77	70.7	71.2	0.79 ± 0.71	1.21	1.22	39.9	32.9	173.1
CPS	68.9	69.0	0.69 ± 0.52	70.8	71.3	0.94 ± 0.53	1.03	1.03	29.0	40.0	135.0
GTCS	58.8	58.9	0.74 ± 0.70	119.1	119.1	0.36 ± 0.20	2.03	2.02	36.0	30.0	996.0
SPS	91.6	91.7	0.57 ± 0.35	109.7	110.1	0.34 ± 0.48	1.20	1.20	40.0	18.0	221.4
CPS	58.3	58.4	0.52 ± 0.38	77.7	78.2	1.39 ± 1.03	1.33	1.34	107.0	24.0	41.0
GTCS	81.8	84.6	2.97 ± 4.77	102.5	101.6	1.84 ± 1.45	1.25	1.20	19.1	40.0	268.6
GTCS	65.7	65.6	0.29 ± 0.23	96.7	96.9	1.36 ± 0.42	1.47	1.48	40.0	19.3	610.7
GTCS	56.4	56.7	0.66 ± 0.65	71.8	76.1	4.35 ± 0.07	1.27	1.34	38.0	18.0	250.6

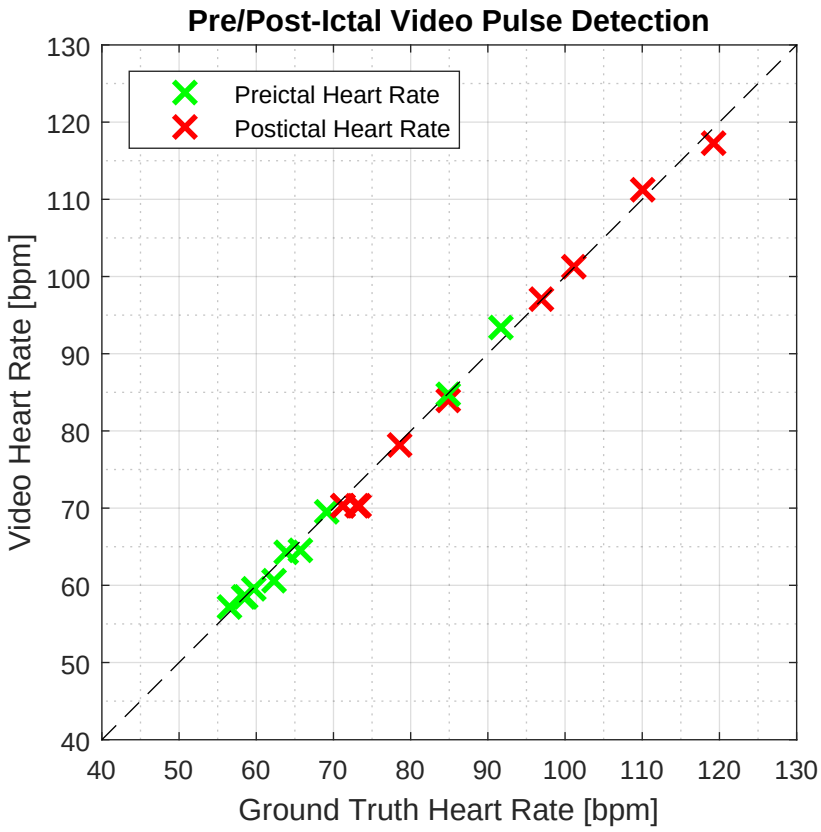


Figure 4.13: Ictal (immediate pre and post) video pulse detection results

In general, epileptic seizures can be hard to detect without ground truth EEG and may only be noticable by subtle differences in behaviour.

In this section, the heart rate is detected from two patients' smartphone videos during an assumed seizure, recorded by a family member. To protect the anonymity of the patients, no screenshots of actual videos are presented in this work, only descriptions are given:

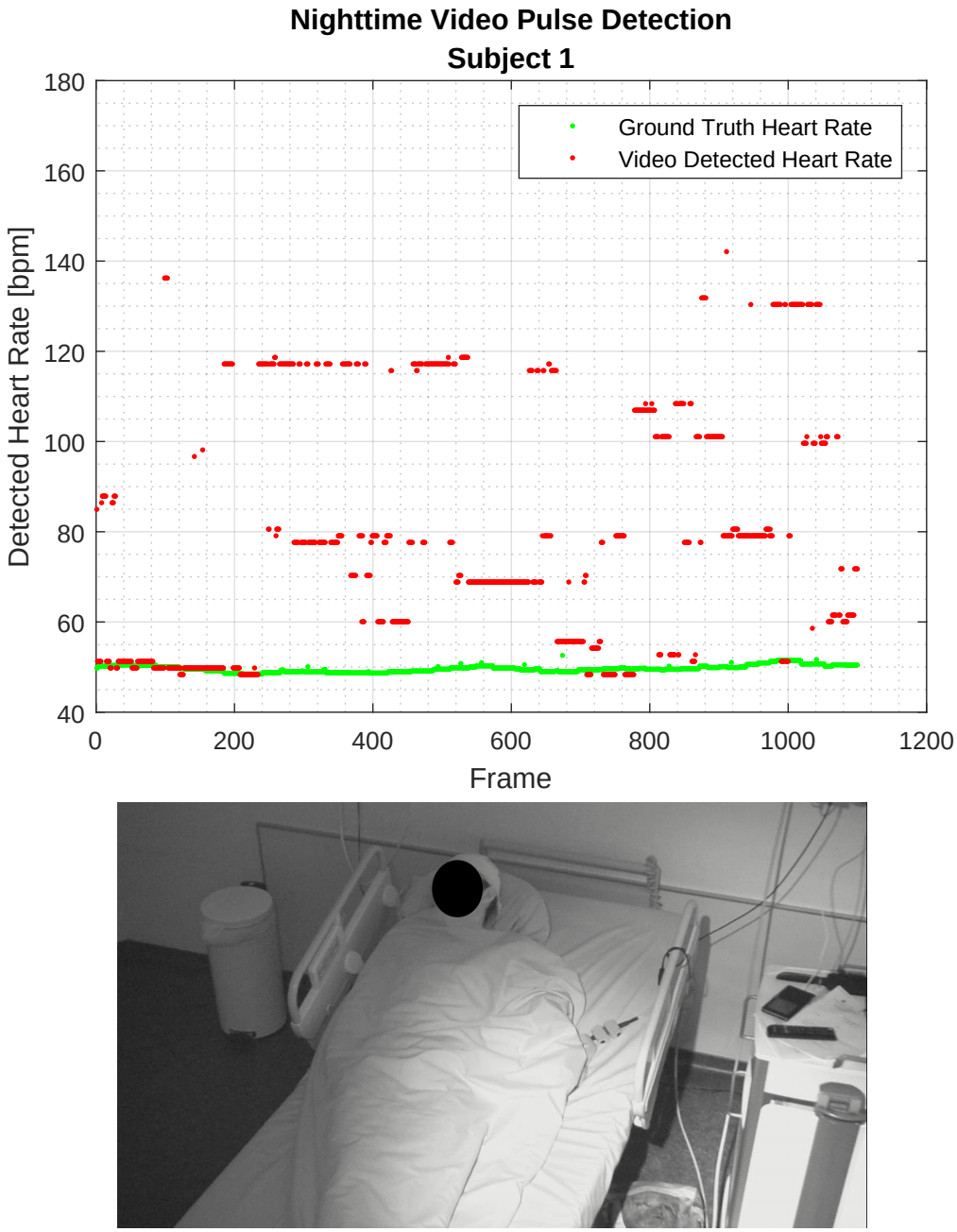


Figure 4.14: Pulse detection results and screenshot of a nighttime (infrared) video (@25 fps) using a window of 16 s.

Table 4.3: Nighttime video heart rate (HR) detection errors using 60 s *of video for each sample.*

	Detected HR [bpm]		
Subject	Video	GT	Error [$bpm \pm SD$]
1	80.1	49.7	30.58 ± 26.54
2	88.7	50.9	37.83 ± 27.01
3	80.9	62.9	18.15 ± 20.43
4	77.7	45.7	31.98 ± 21.57
5	65.7	64.9	4.08 ± 10.83
6	72.8	51.4	21.45 ± 24.44
7	124.1	75.5	51.39 ± 14.20
8	53.9	53.8	0.42 ± 1.91
9	76.4	63.8	15.19 ± 13.20
10	95.2	62.2	33.81 ± 19.92

Smartphone video example 1:
The video is 34 seconds long (@25 fps) and filmed in a vertical orientation. It shows the face of an elderly woman, recorded with the smartphone of a family member. She stares absently into the room and is unable to answer questions addressed to her. Camera movement is existent in all directions and a slight zooming of the video is present as well.

Smartphone video example 2:
The video has a length of 38 seconds (@29.97 fps) and shows a man at around 60 years of age lying on the floor of what seems like a restaurant, recovering from an assumed epileptic seizure. He is able to respond to questions, but unable to get up. Camera movement as well as shake are present. The scene is slightly backlit.

Even though no ground truth heart rate data is available in these self recorded cases, one can still analyze the heart rates in these videos and compare them with typical interictal or postictal phases that were recorded in a controlled, clinical environment.

In the first example, the detection result displayed in Fig. 4.15 shows a steady heart rate of 80 bpm. The lower part of the figure shows the development of Welch's

power spectral density estimation over time. Here, a continuous and well defined crest can be observed. In typical interictal phases with no physical exertion during a hospital stay, a resting pulse rate of 60 bpm was observed for this patient, indicating that the smartphone video possibly showed a seizure due to the increased heart rate.

In the second example, the pulse derived from a part of the assumed postictal smartphone video is shown in Fig. 4.16. Although the most prominent crest of the power spectral density development is not as pronounced as in the first example, the method is still able to reconstruct an almost constant pulse that averages at 78.25 bpm. For this patient, the heart rate calculated from a seizure recorded with the clinical ECG system showed preictal and postictal heart rates of 62 bpm and 87 bpm, respectively. Here, the video detected pulse is close to the clinically recorded postictal heart rate. Estimating from the downward sloping nature observed in frames 0-31 in Fig. 4.16, the patient's heart rate currently seems to be slowly diminishing to normal resting pulse.

In reality, only a few of the patients' smartphone videos exhibit the quality that is needed to produce reliable pulse detection results. In most cases, video pulse detection is impossible due to excessive camera and patient movement as well as bad scene illumination. On the other hand, caretakers could be educated to properly film seizures so that the video is processable by the method.

4.6.7 A Test of the Influence of Makeup on Video Pulse Detection

During video recording, makeup can cover parts of or even the entire facial area. Even when makeup is put on invisibly, i.e., in the same color tone as the underlying skin, it could still prevent the video pulse detection algorithm from working correctly. Unfortunately, in real world applications, the use of makeup cannot be controlled. It is expected that applied makeup makes video pulse detection lose its accuracy or even make it impossible. In this section, the influence of a makeup occlusion layer is tested by making one half of the face up and leaving the other half untouched (see Fig. 4.17). Then, the heart rate is independently determined from both halves of the face using video pulse detection and compared against the simultaneously recorded clinical ECG ground truth. To limit artefacts resulting from eye or mouth movement, the initial ROIs are chosen so that, after shrinking, only the cheek of the

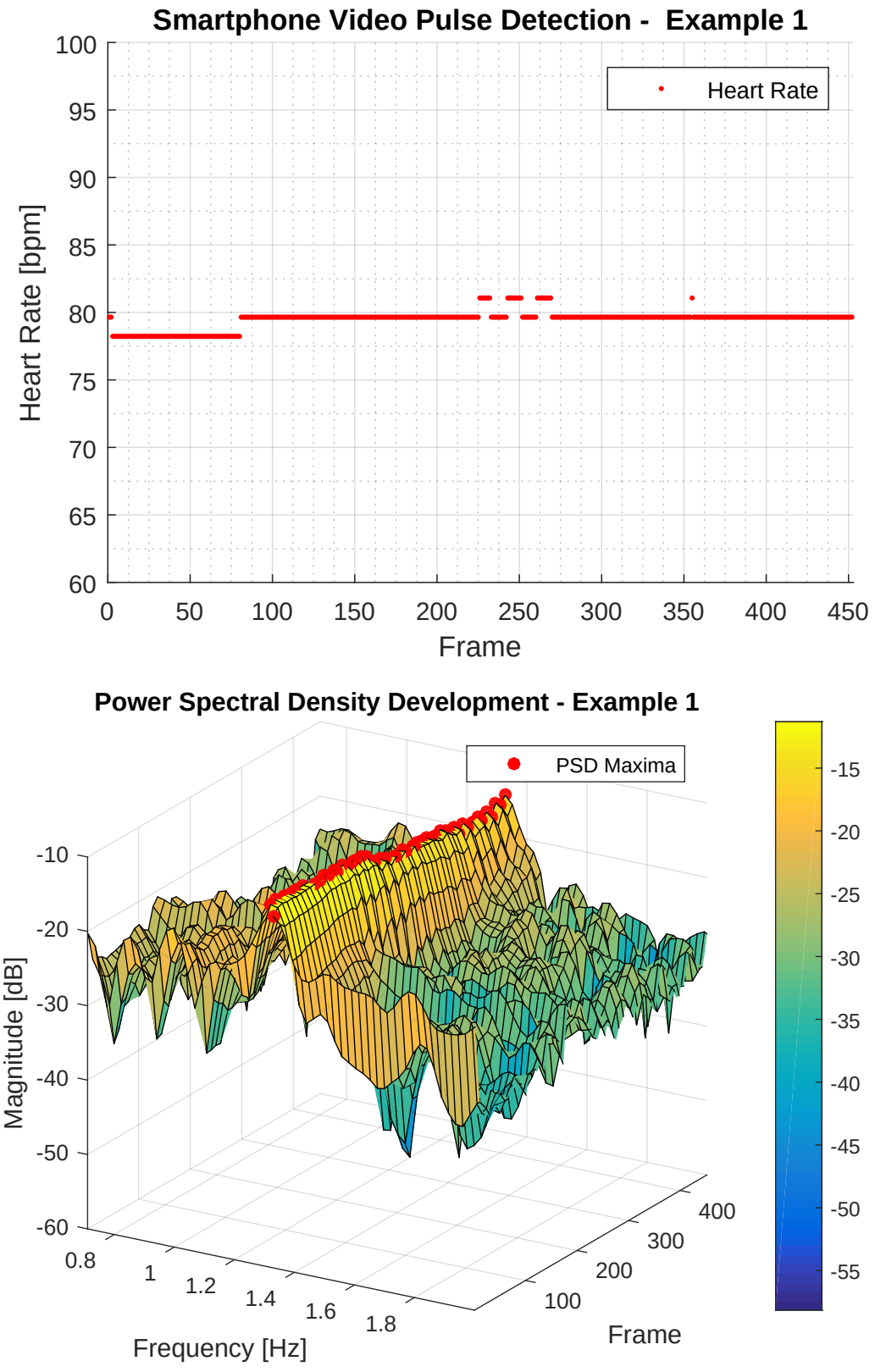


Figure 4.15: Pulse detection result and power spectral density estimation development of a smartphone video (Example 1) showing an assumed epileptic seizure.

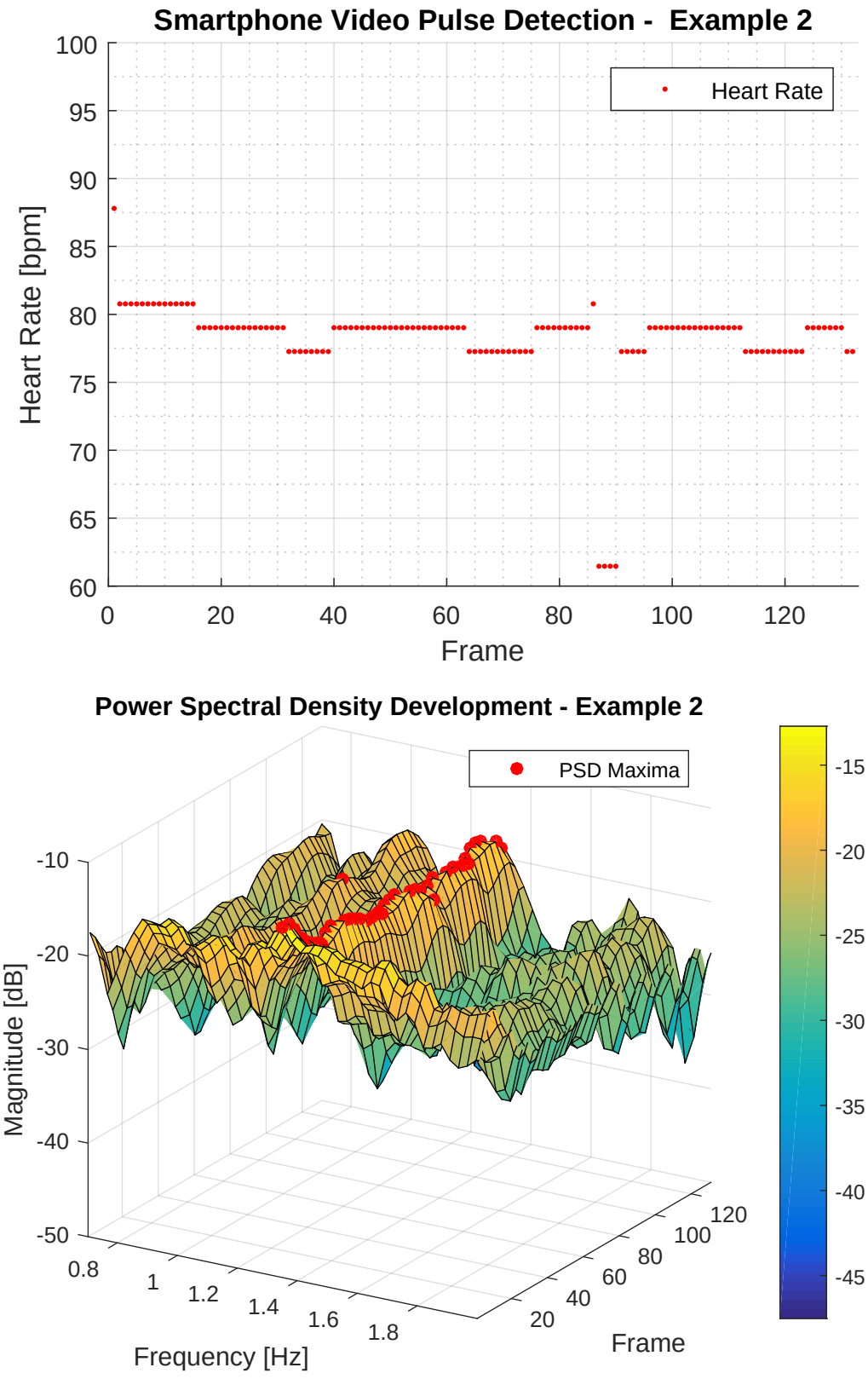


Figure 4.16: Pulse detection result and power spectral density estimation development of a smartphone video (Example 2) showing the postictal phase of an assumed epileptic seizure.

Figure 4.17: Setup of the makeup influence evaluation experiment. The right half of the face is covered with makeup while the other half is left untouched.

respective half face is used as input for video pulse detection. This can be seen in Fig. 4.18.

The results of this evaluation are presented in Fig. 4.19. As expected, video pulse detection from the untouched part of the face shows good results (detection error: 0.99 ± 1.03 bpm), while the detected heart rates from the made up part of the face exhibit a much larger variation (8.64 ± 1.03 bpm). A comparison between the respective developments of the power spectral densities over time (see Fig 4.20) shows a definitive crest along the maxima of the power spectra for the face half without makeup while the made up side reveals an almost flat magnitude plane and thus leaves the algorithm with no distinct prominent peaks to detect. Keep in mind that this test was conducted in a controlled environment using studio lighting equipment. It is to be expected that results are even worse under realistic conditions.

4.6.8 Limitations

The presented results show that the method is well suited for epileptic seizure heart rate analysis in the pre- and postictal phases and, under suitable conditions, pro-

Figure 4.18: ROIs (with tracking markers) used for the makeup influence test.
Left: Without makeup applied
Right: With makeup applied

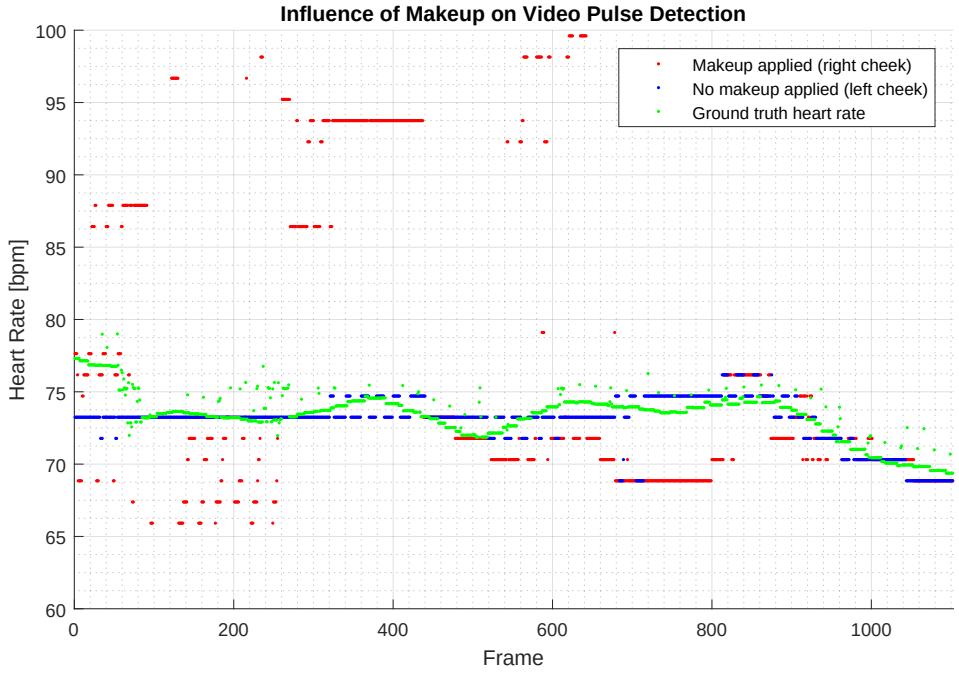


Figure 4.19: Results of the influence on makeup on video pulse detection test.

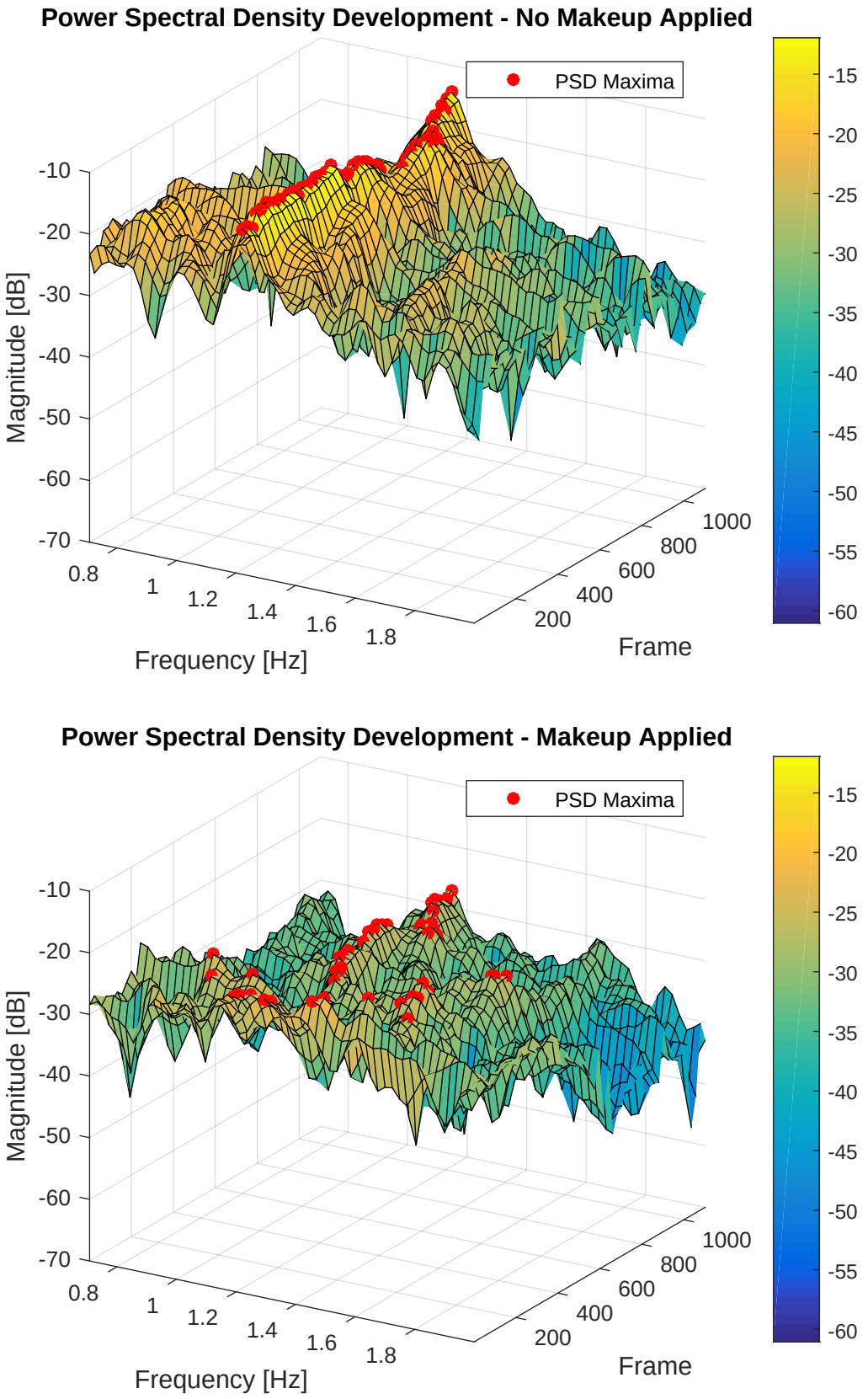


Figure 4.20: Development of the power spectral densities and their maxima over time, calculated from the face half without (top) and with (bottom) makeup applied.

duces results equal to the ECG ground truth. In ictal video pulse detections, it is often problematic to find a large enough, contiguous frame interval without excessive movement or occlusions, making tracking of the face and the subsequent video pulse detection in these phases nearly impossible. Furthermore, nighttime (infrared) videos were shown to be unsuitable for the method, whereas grayscale converted RGB videos showed surprisingly good results. Also, the use of makeup acts like an almost opaque layer between the skin and the visible face, which proved to be a challenge for video pulse detection.

4.7 Conclusion and Future Work

In this chapter, a method for video pulse detection of epileptic seizures was presented and evaluated against ground truth ECG annotated videos of patients during interictal, ictal and nighttime phases. The applicability on smartphone videos as well as the influence of makeup on the detection result were also discussed. The results show that careful application of the method whilst being aware of its limitations makes heart rate ratio based seizure detection and even classification possible.

For future work, a method for estimating a confidence level of the detected heart rates could be developed, so that in the absence of ground truth data, an estimation of result reliability can be given. The continuity and prominence of the power spectral density development crestline could possibly be used in this case. The application of applying the method to thermal camera videos would be another interesting line of research.

5
Conclusion

This thesis presented and summarized methods for the reconstruction of human motion using a multitude of sensors and motion capture setups in a variety of contexts and applications.

Clean, raw motion capture data is the foundation of high quality animation and is used as input for many algorithms in computer animation. When recording with a passive optical mocap system, marker data may get lost due to tracking problems, mislabelings or occlusions, amongst other reasons. The trajectories of single missing markers or short time gaps in trajectories of multiple markers are generally easy to fill by utilizing rigid body properties or spline interpolation. In the situation of missing multiple markers for long-time gaps, a data-driven method was presented in Chapter 2. The algorithm used a mocap database stored in an efficient spatial indexing structure (kd-tree) to search for nearest pose neighbors using normalized data of markers that are continuously valid in the gap. Ultimately, the retrieved examples served as priors for synthesizing the missing marker data. The method was applied to real gaps present in motions originating from publicly available motion databases as well as to artificial gaps created for the purpose of evaluation and was shown to work well over a variety of motion styles, databases and actors.

Unfortunately, many data-driven cleaning methods suffer from an out-of-sample problem, i.e., they would be unable to reconstruct a cartwheel motion from a database comprised of only walking motions. However, since the presented method scales well to large mocap databases likely to contain a clean version of almost every imaginable motion, it represents a useful aid in automated motion capture data refinement. In the rare case that no poses of similar motions can be found in the

database, a general cleaning framework could combine the presented data-driven method with complementary approaches like [THC15, FXZ+14, Fed13, BL16].

Building upon having a motion capture database containing artifact-free motions, a method for analyzing a stream of motion data for similar, previously labeled actions was developed in Chapter 3 by extending the *Lazy Neighborhood Graph* by Krüger et al. [KTWZ10]. The method is not only able to detect exact copies of actions stored in the database, but can handle temporal deviations from the original motion as well. It was shown that the method is very flexible regarding the sensor input data by demonstrating successful action recognition from simple accelerometers, Kinect skeletons, optical motion capture data and video based features.

Even though the chosen featuresets used for retrieving similar poses seemed to work well in the discussed applications, it would be interesting to evaluate the action recognition specific similarity models assessed by Valcik et al. [VSZ15a]. Also, in case the presented action recognition scenarios used Kinect skeletons to query the database for similar poses, an application of an improved Kinect skeleton estimation algorithm like [VSZ15b] to the query skeleton data may improve recognition results.

In Chapters 2 and 3, motion completion and action recognition were applied to traditional, full body motion capture recordings. In contrast, as presented in Chapter 4, the indirect detection of the human heart's pumping motion, i.e., the pulse frequency, in the context of epileptic seizure detection and classification, represents motion reconstruction on a much finer scale. Here, the normally invisible and subtle color changes in a video recording of a subject's face that are caused by blood volume changing over time were analyzed and their dominant frequency extracted using signal processing techniques. The algorithm was first tested under controlled, non-seizure, clinical conditions, comparing the extracted frequencies with ground truth ECG measurements, demonstrating that a high detection accuracy could be achieved. In most cases, when applied in the context of detecting epileptic seizures, where a significant increase in heart rate can usually be observed during the ictal phase, the video pulse detection algorithm was unable to detect a valid pulse signal due to excessive patient movement, camera shake, occlusions or changes in illumination. This was remedied by shifting the video pulse analysis to the immediate pre- and postictal phases, allowing the calculation of the absolute and relative increases in the subject's heart rate. The results of this scenario were again validated

using simultaneously recorded clinical ECG data. Finally, the method has also been shown to work on patient-supplied smart phone videos of seizures recorded outside of the clinical environment. Here, the smartphone video detected heart rates were compared to typical clinical seizure recordings of the same patient and shown to have similar patterns.

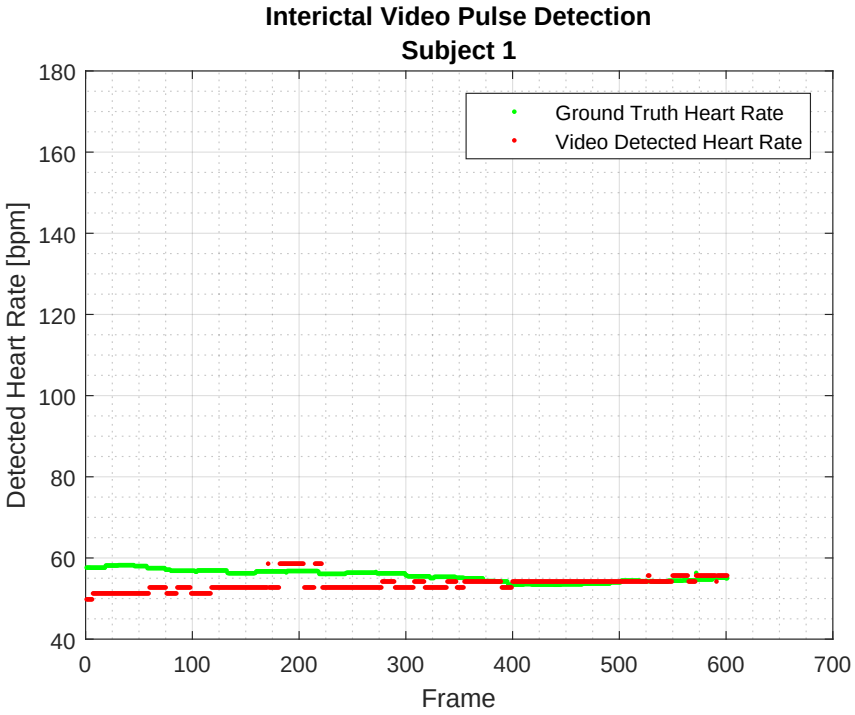
When processing videos of seizures, one is not only limited to analyzing the heart rate. During epileptic seizures, patients often display characteristic motion patterns. This can range from a stiffening of the whole body that develops into repetitive rhythmic jerking of arms and legs observable during a generalized tonic-clonic seizure to more subtle, patient specific involuntary repetitive hand movements. Using these motion patterns as *actions* in the proposed action recognition method (see Section 3.5.6) in combination with video pulse detection would be a step towards an automatic, video based epileptic seizure detection and alerting system.

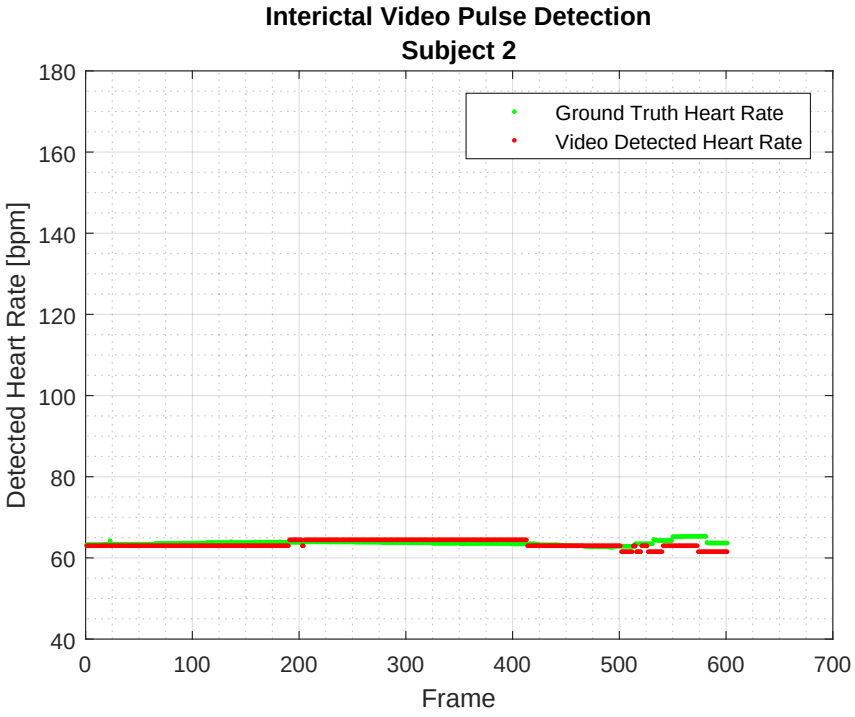
Even outside of the epilepsy context, a system that is able to detect vital signs like heart rate, body temperature and respiratory rates by analyzing RGB and thermal camera video streams would be of great help during emergency situations like car accidents. Here, first-aiders could record smartphone videos of unreachable, trapped passengers to gain valuable first insights about their condition and forward them to the notified rescue team.

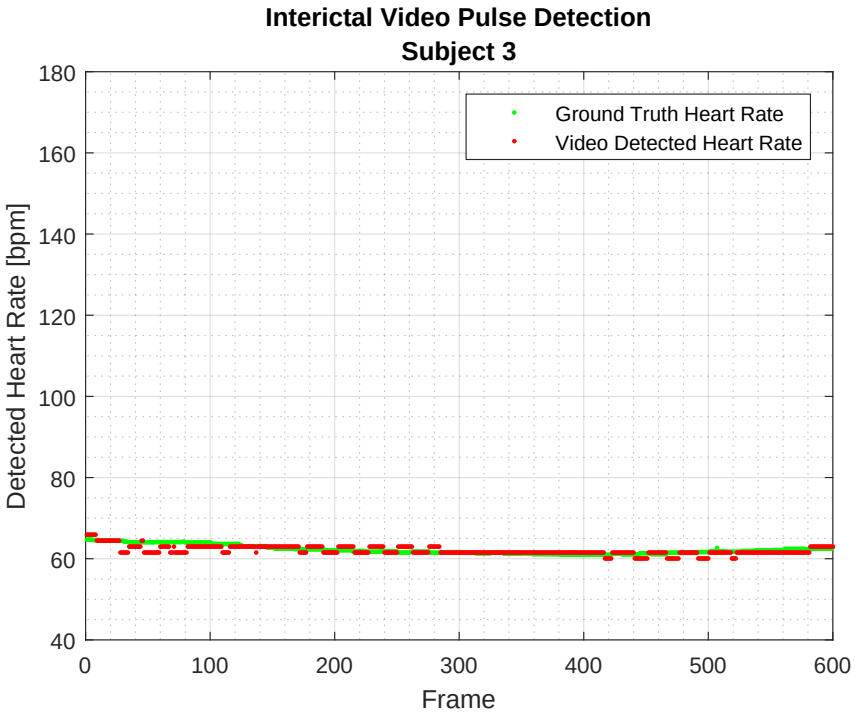
A

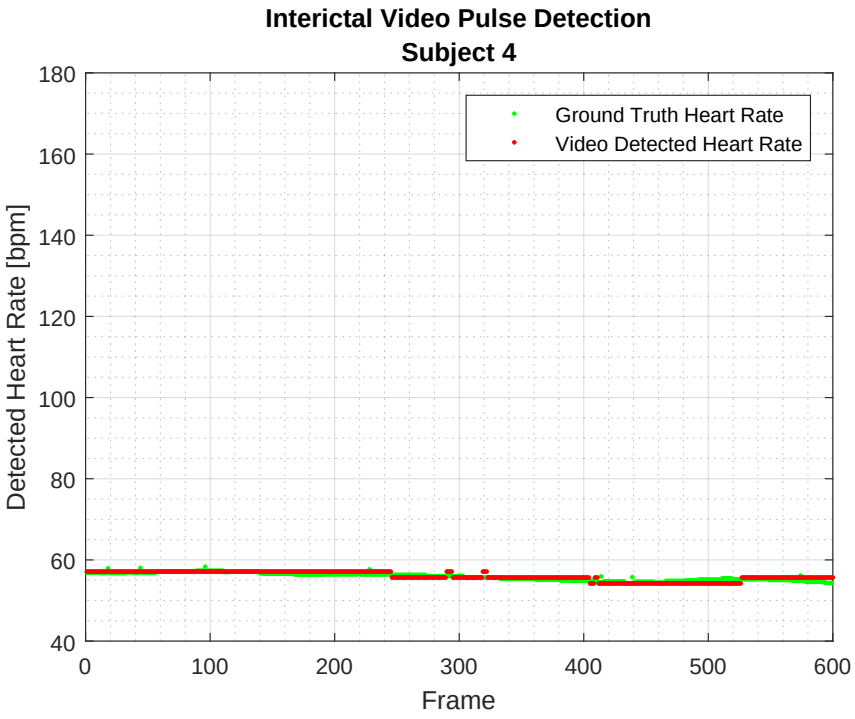
Video Pulse Detection Result Plots

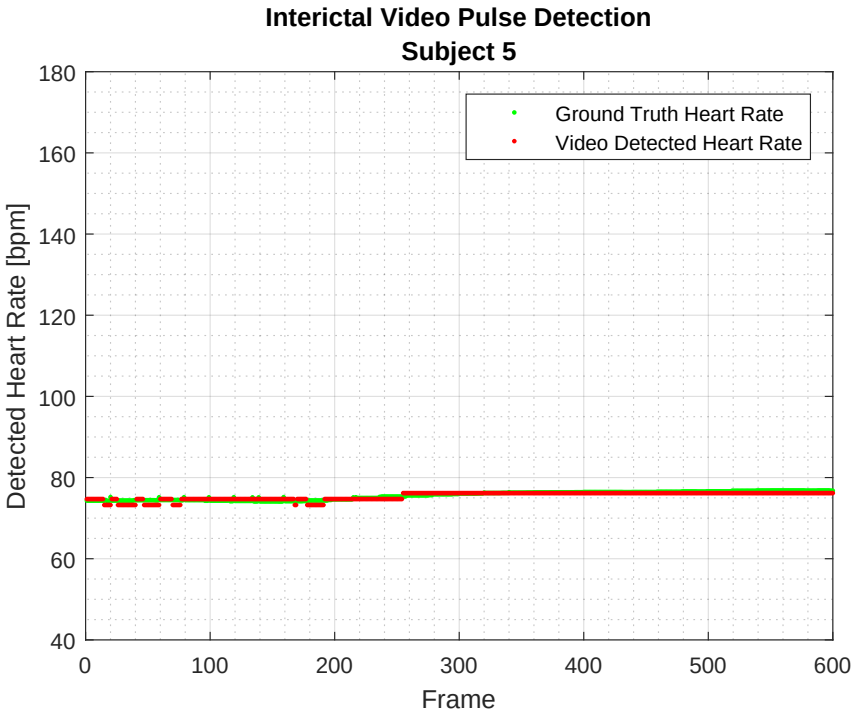
A.1 Interictal Video Pulse Detection Plots

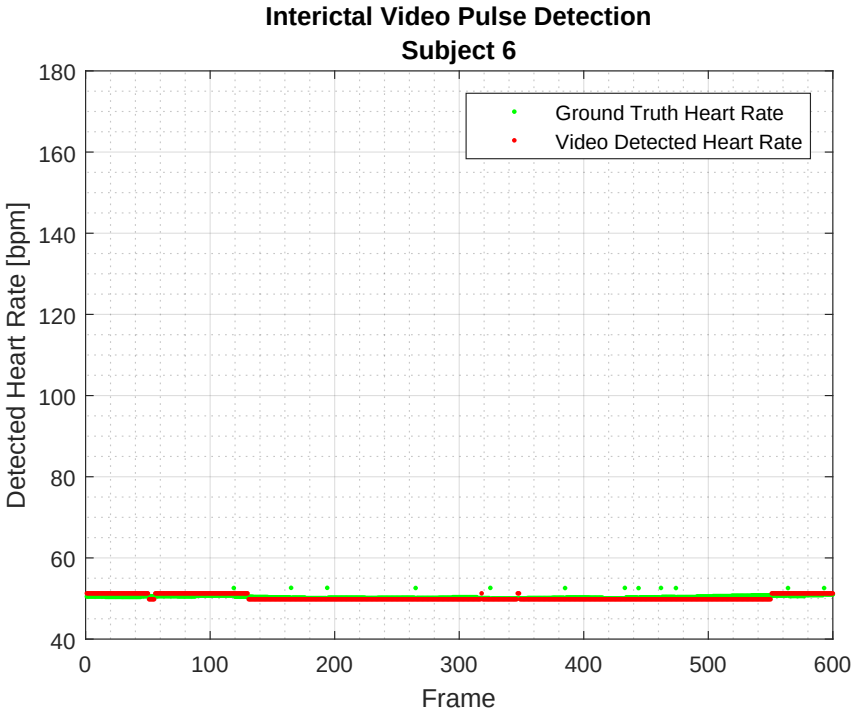


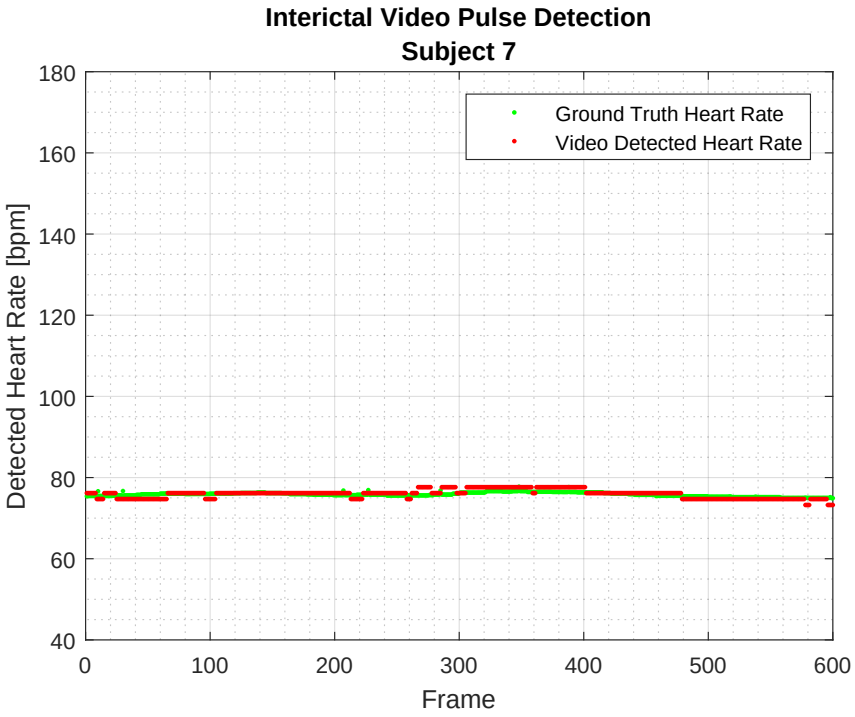

Interictal Video Pulse Detection
Subject 2
Ground Truth Heart Rate
Video Detected Heart Rate
Detected Heart Rate [bpm]
Frame
180
160
140
120
100
80
60
40
0
100
200
300
400
500
600
700

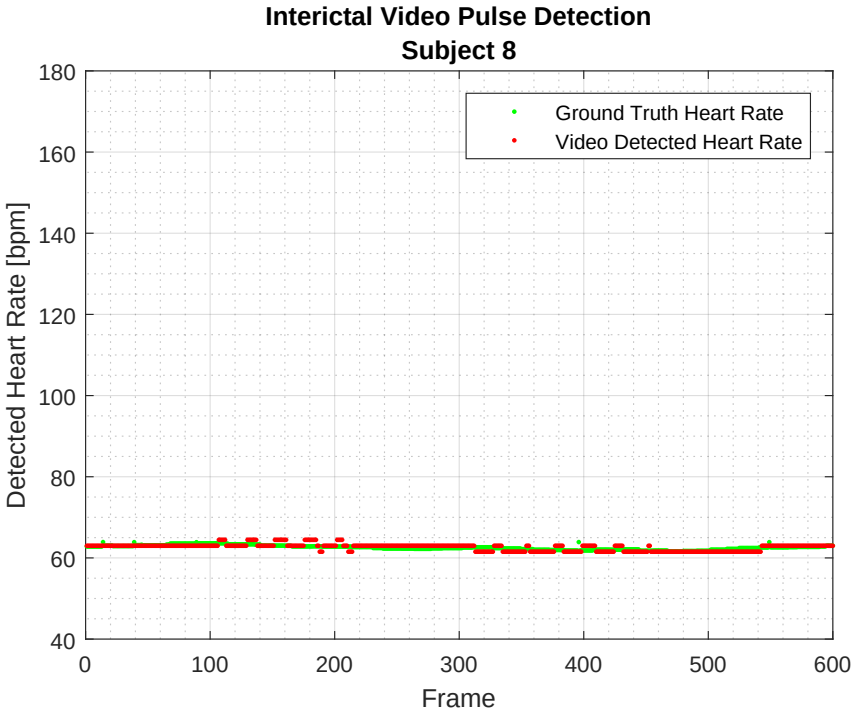

Interictal Video Pulse Detection
Subject 3
Ground Truth Heart Rate
Video Detected Heart Rate
Detected Heart Rate [bpm]
Frame
180
160
140
120
100
80
60
40
0
100
200
300
400
500
600

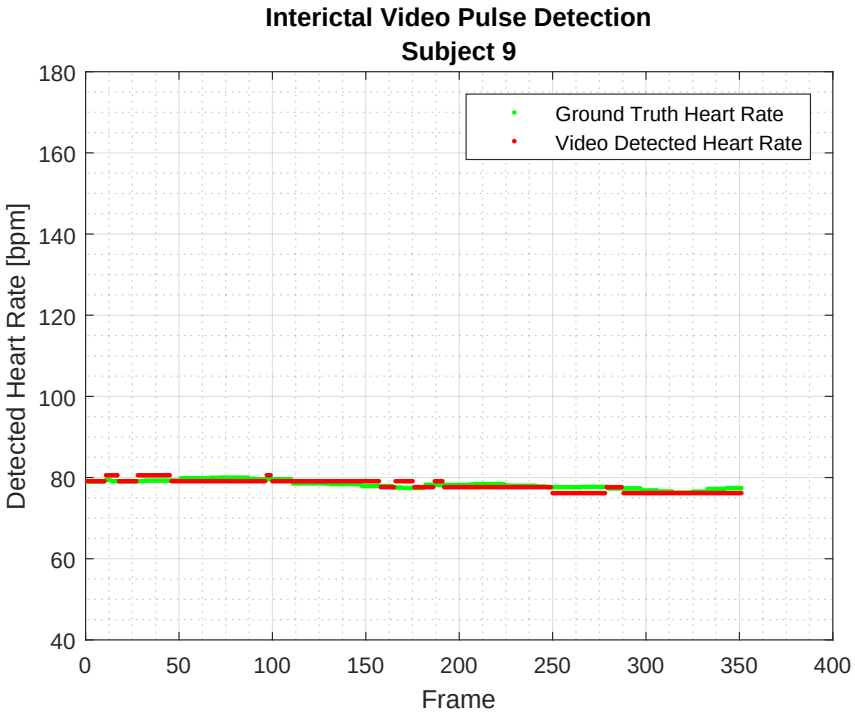

Interictal Video Pulse Detection
Subject 4
Ground Truth Heart Rate
Video Detected Heart Rate
Detected Heart Rate [bpm]
Frame
180
160
140
120
100
80
60
40
0
100
200
300
400
500
600

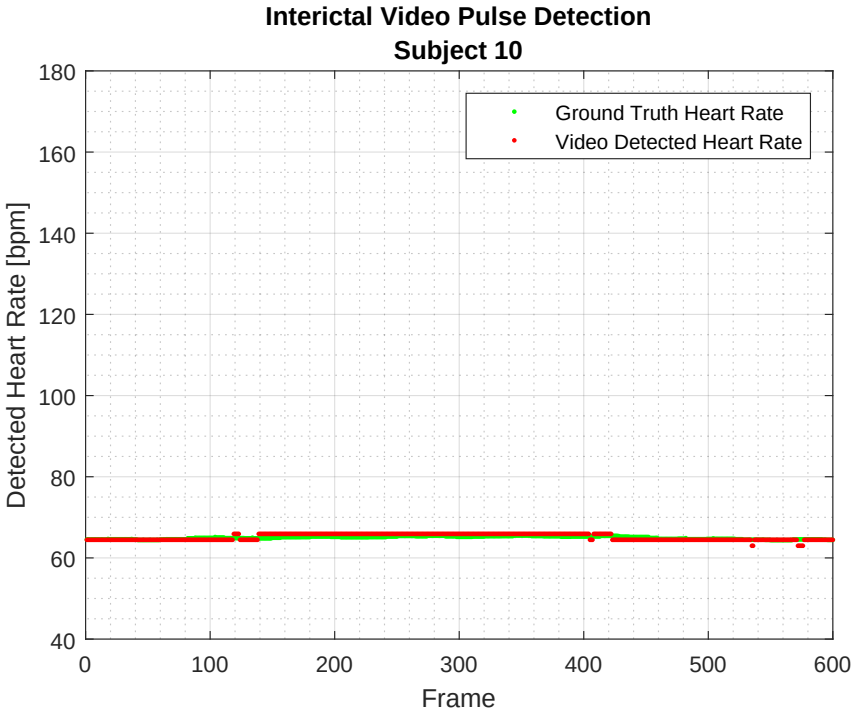

Interictal Video Pulse Detection
Subject 5
Ground Truth Heart Rate
Video Detected Heart Rate
Detected Heart Rate [bpm]
Frame
180
160
140
120
100
80
60
40
0
100
200
300
400
500
600


Interictal Video Pulse Detection
Subject 6
Ground Truth Heart Rate
Video Detected Heart Rate
Detected Heart Rate [bpm]
Frame
180
160
140
120
100
80
60
40
0
100
200
300
400
500
600

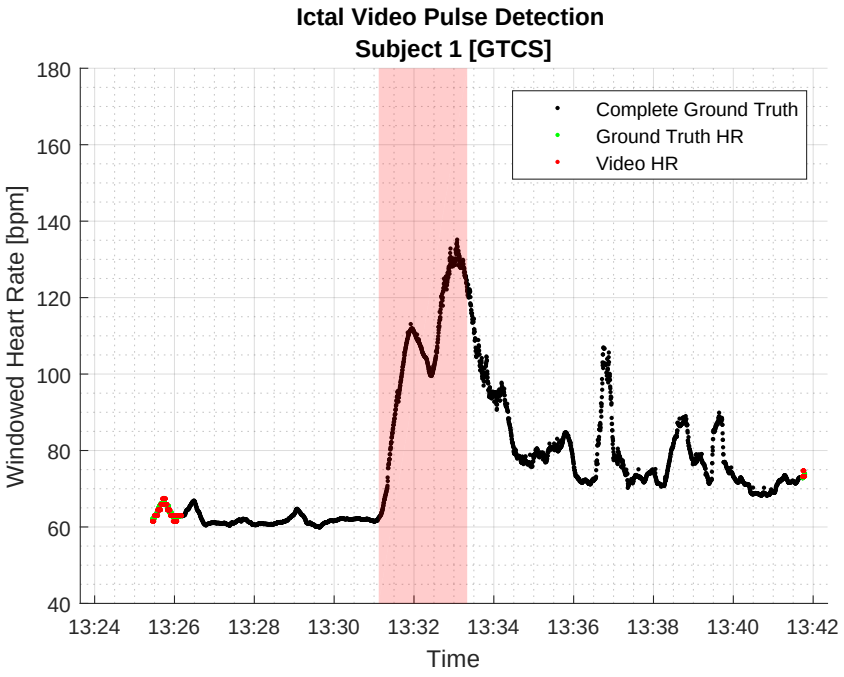

Interictal Video Pulse Detection
Subject 7
Ground Truth Heart Rate
Video Detected Heart Rate
Detected Heart Rate [bpm]
Frame
180
160
140
120
100
80
60
40
0
100
200
300
400
500
600

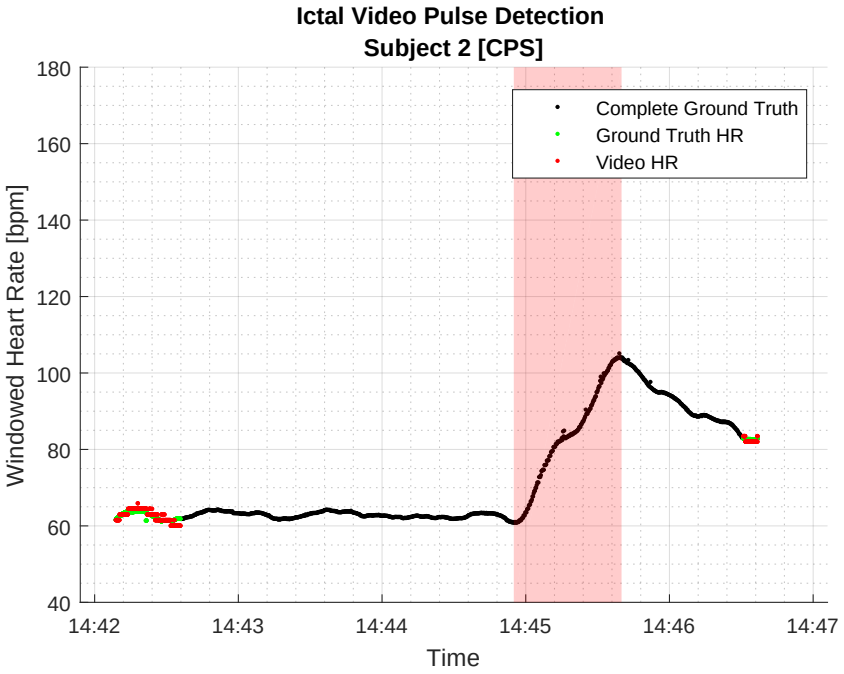

Interictal Video Pulse Detection
Subject 8
Ground Truth Heart Rate
Video Detected Heart Rate
Detected Heart Rate [bpm]
Frame
180
160
140
120
100
80
60
40
0
100
200
300
400
500
600

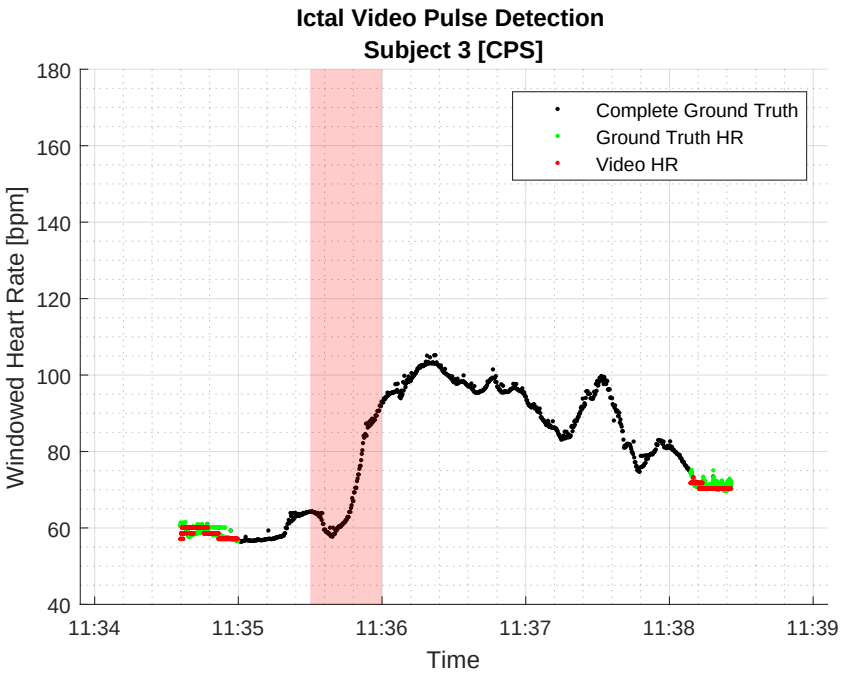

Interictal Video Pulse Detection
Subject 9
Ground Truth Heart Rate
Video Detected Heart Rate
Detected Heart Rate [bpm]
Frame
180
160
140
120
100
80
60
40
0
50
100
150
200
250
300
350
400

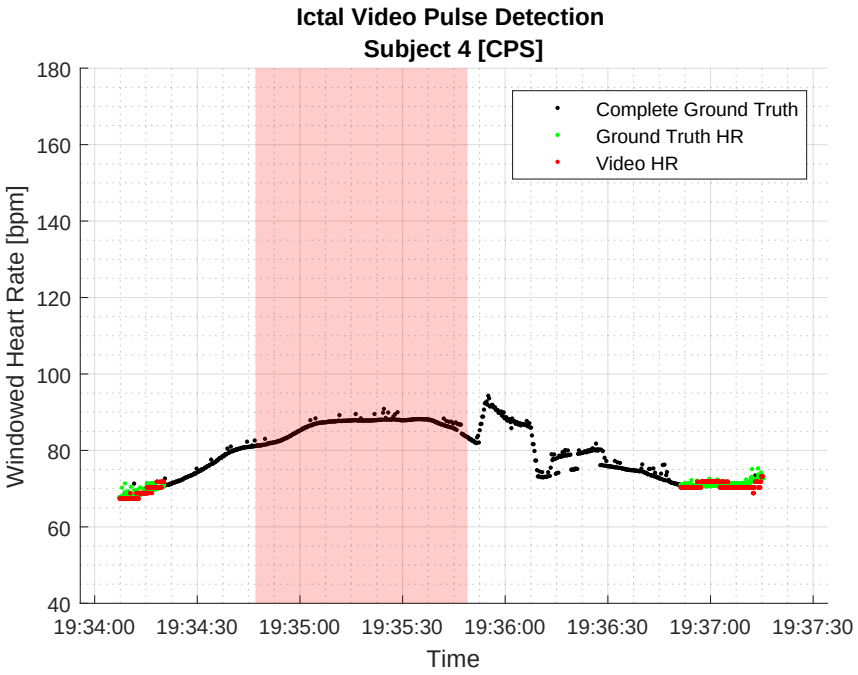

Interictal Video Pulse Detection
Subject 10
Ground Truth Heart Rate
Video Detected Heart Rate
Detected Heart Rate [bpm]
Frame
180
160
140
120
100
80
60
40
0
100
200
300
400
500
600

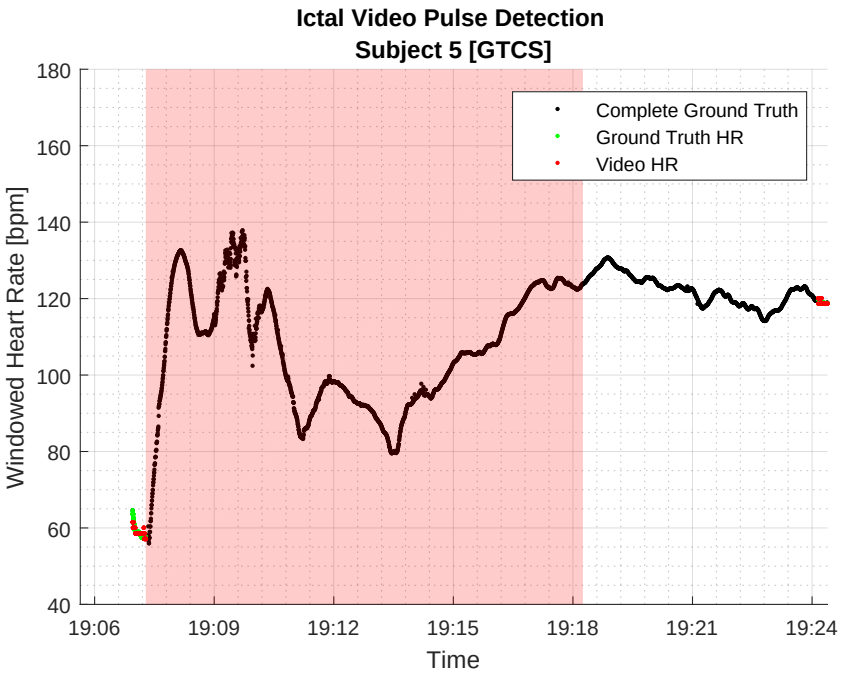
A.2 Ictal Video Pulse Detection Plots

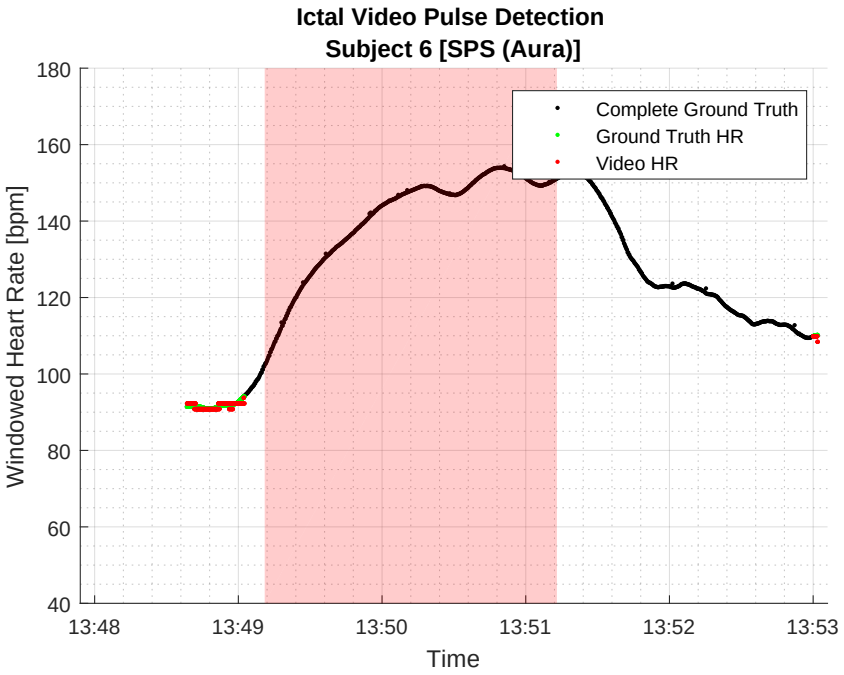


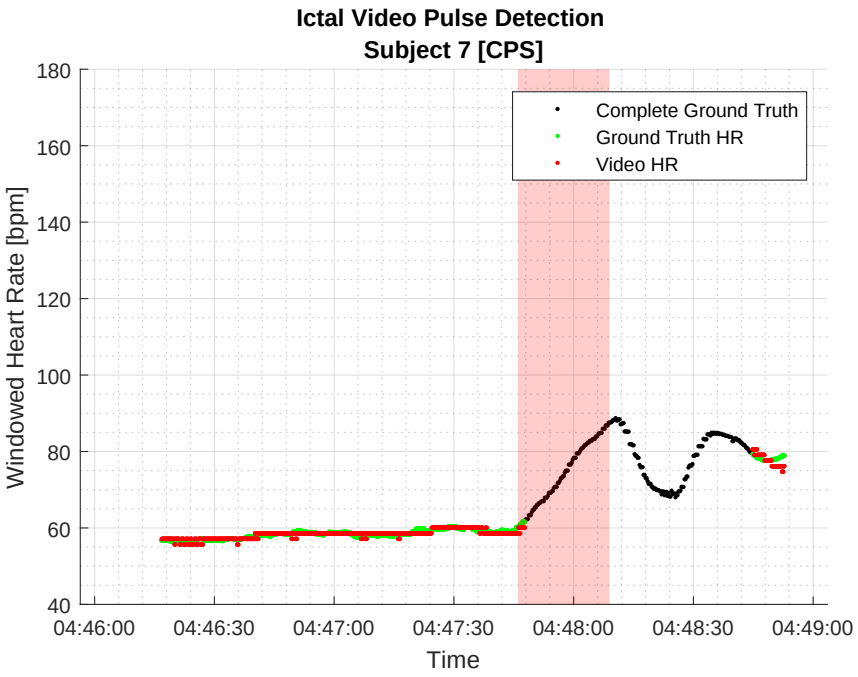


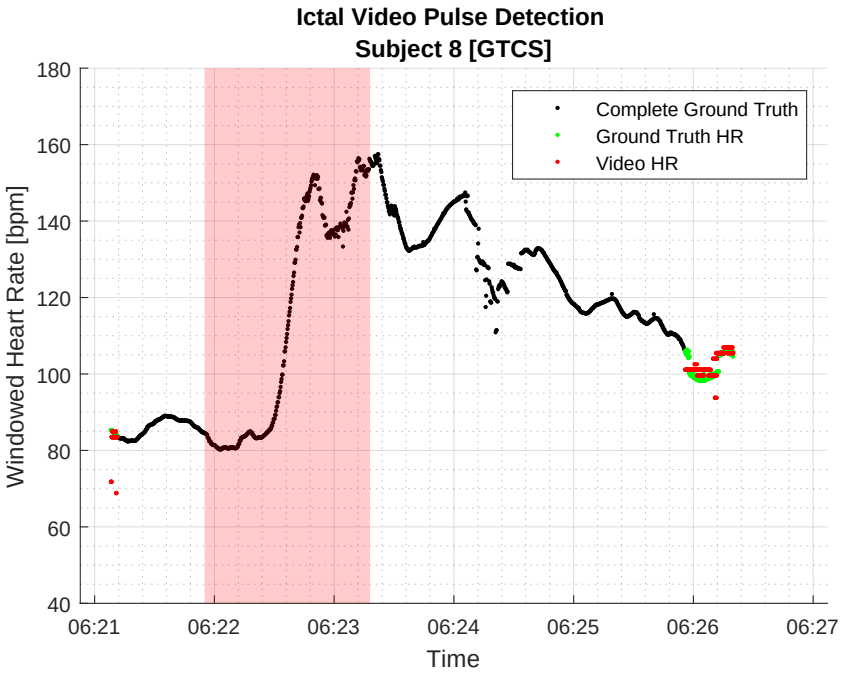

Ictal Video Pulse Detection
Subject 3 [CPS]
Complete Ground Truth
Ground Truth HR
Video HR
Windowed Heart Rate [bpm]
180
160
140
120
100
80
60
40
11:34
11:35
11:36
11:37
11:38
11:39
Time

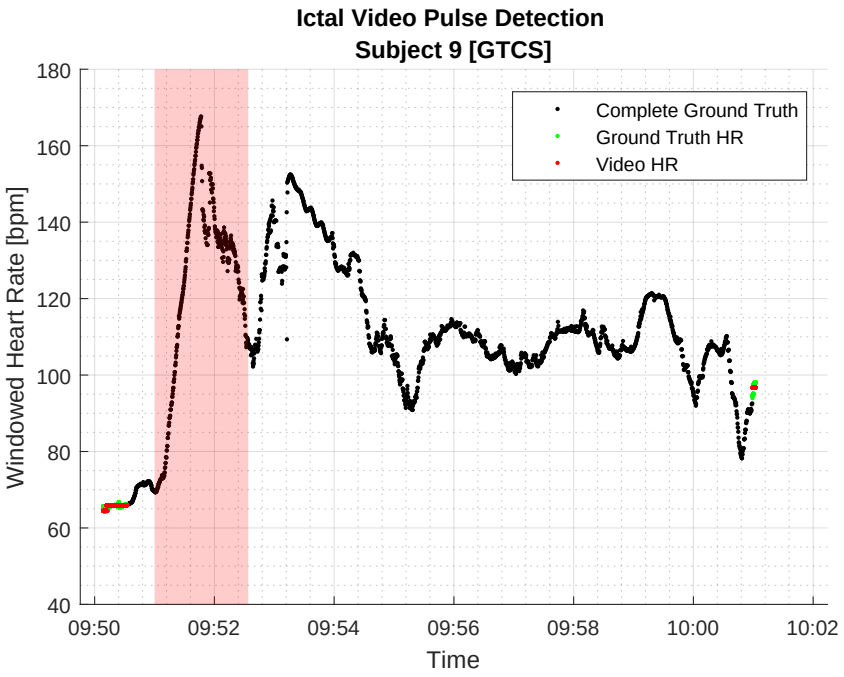

Ictal Video Pulse Detection
Subject 4 [CPS]
Complete Ground Truth
Ground Truth HR
Video HR
Windowed Heart Rate [bpm]
180
160
140
120
100
80
60
40
19:34:00
19:34:30
19:35:00
19:35:30
19:36:00
19:36:30
19:37:00
19:37:30
Time

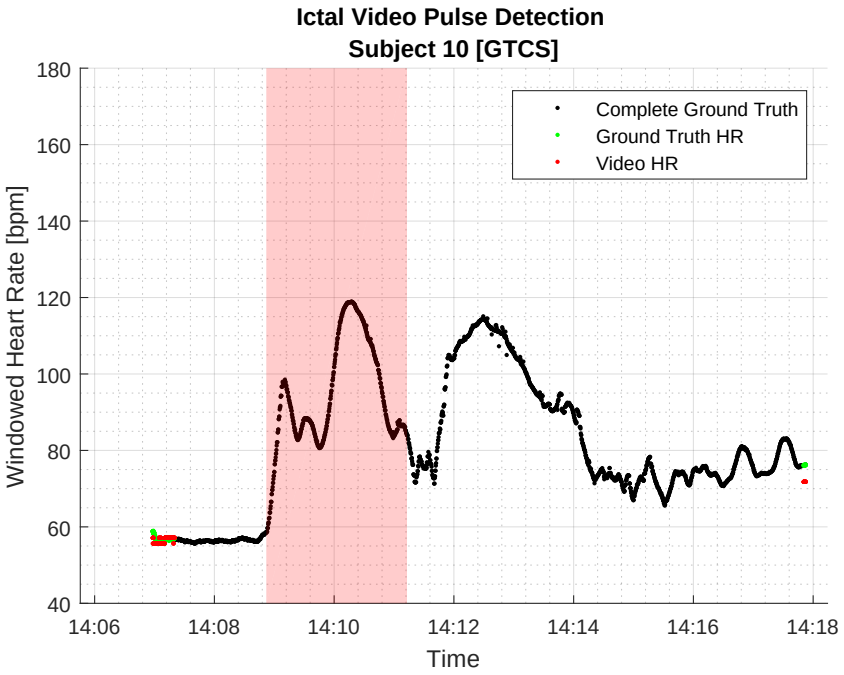

Ictal Video Pulse Detection
Subject 5 [GTCS]
Complete Ground Truth
Ground Truth HR
Video HR
Windowed Heart Rate [bpm]
180
160
140
120
100
80
60
40
19:06
19:09
19:12
19:15
19:18
19:21
19:24
Time


Ictal Video Pulse Detection
Subject 6 [SPS (Aura)]
Complete Ground Truth
Ground Truth HR
Video HR
Windowed Heart Rate [bpm]
180
160
140
120
100
80
60
40
13:48
13:49
13:50
13:51
13:52
13:53
Time

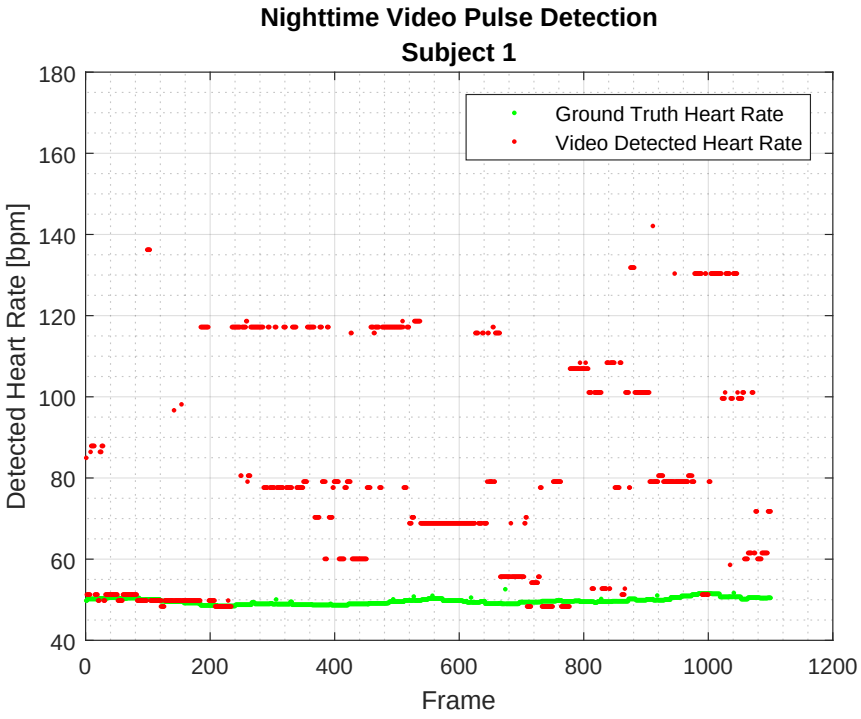

Ictal Video Pulse Detection
Subject 7 [CPS]
Complete Ground Truth
Ground Truth HR
Video HR
Windowed Heart Rate [bpm]
180
160
140
120
100
80
60
40
04:46:00
04:46:30
04:47:00
04:47:30
04:48:00
04:48:30
04:49:00
Time

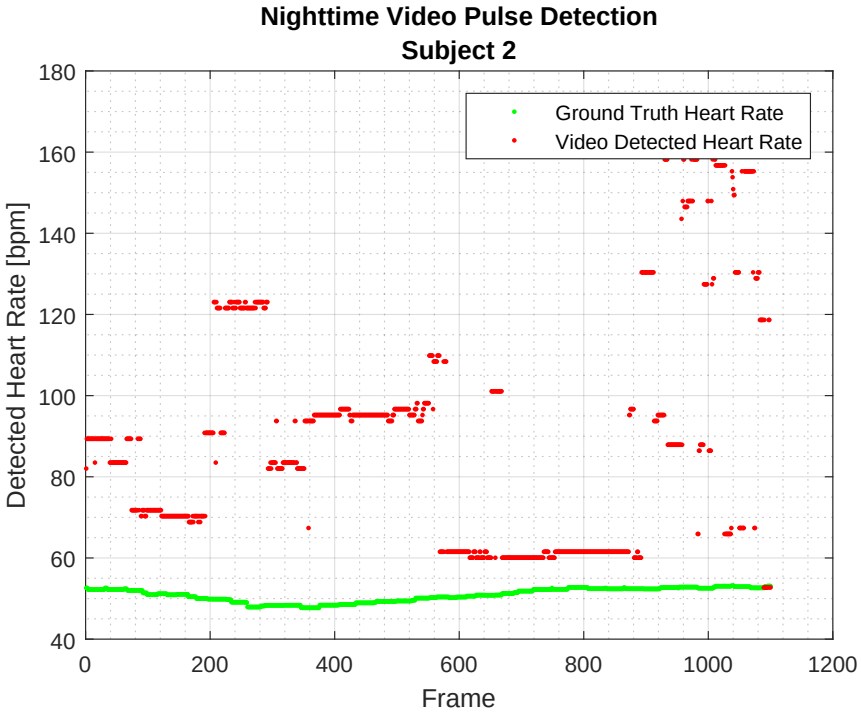

Ictal Video Pulse Detection
Subject 8 [GTCS]
Complete Ground Truth
Ground Truth HR
Video HR
Windowed Heart Rate [bpm]
180
160
140
120
100
80
60
40
06:21
06:22
06:23
06:24
06:25
06:26
06:27
Time

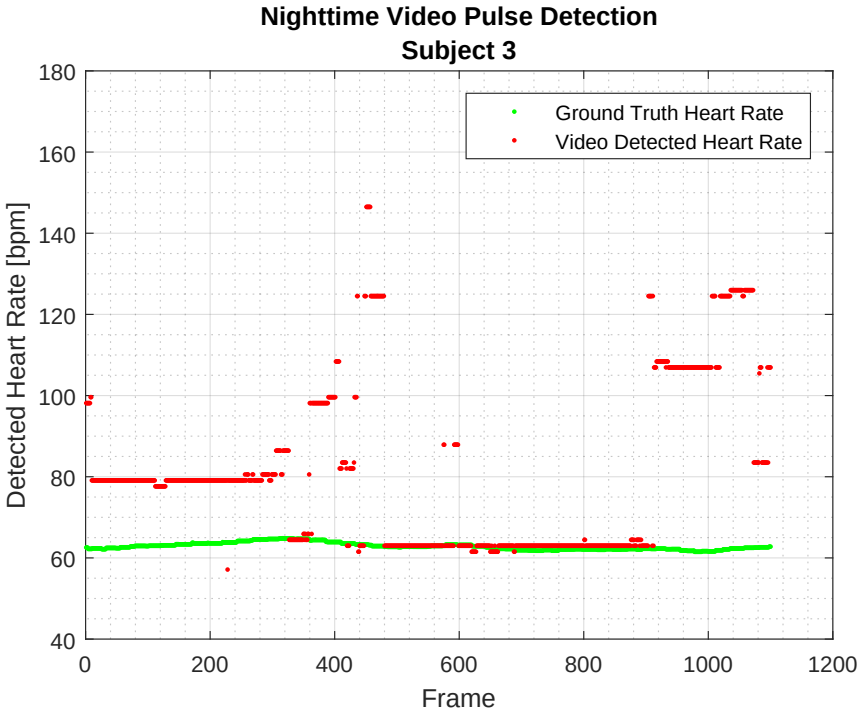

Ictal Video Pulse Detection
Subject 9 [GTCS]
Complete Ground Truth
Ground Truth HR
Video HR
Windowed Heart Rate [bpm]
180
160
140
120
100
80
60
40
09:50
09:52
09:54
09:56
09:58
10:00
10:02
Time

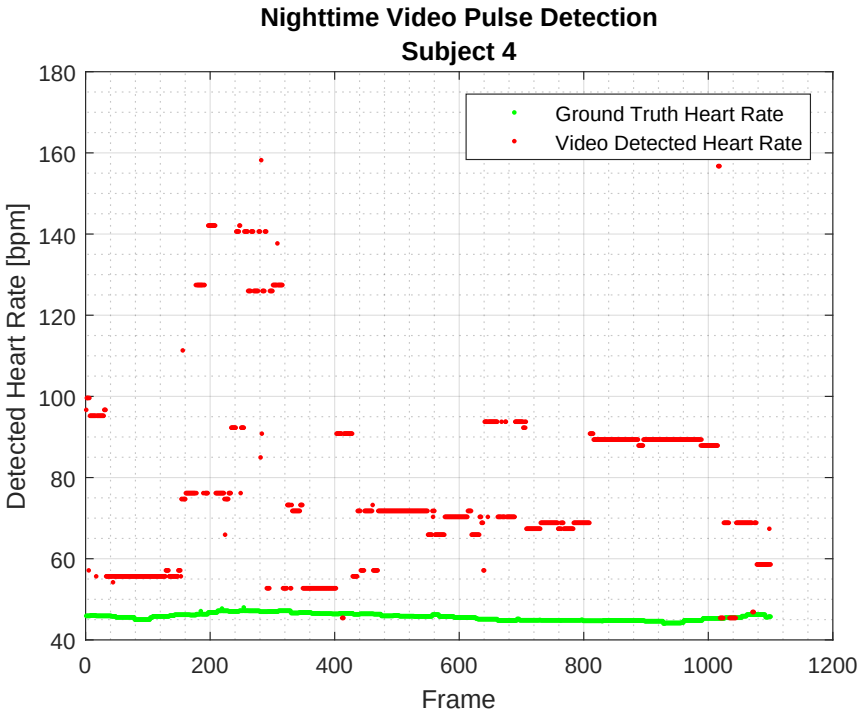

Ictal Video Pulse Detection
Subject 10 [GTCS]
Complete Ground Truth
Ground Truth HR
Video HR
Windowed Heart Rate [bpm]
180
160
140
120
100
80
60
40
14:06
14:08
14:10
14:12
14:14
14:16
14:18
Time

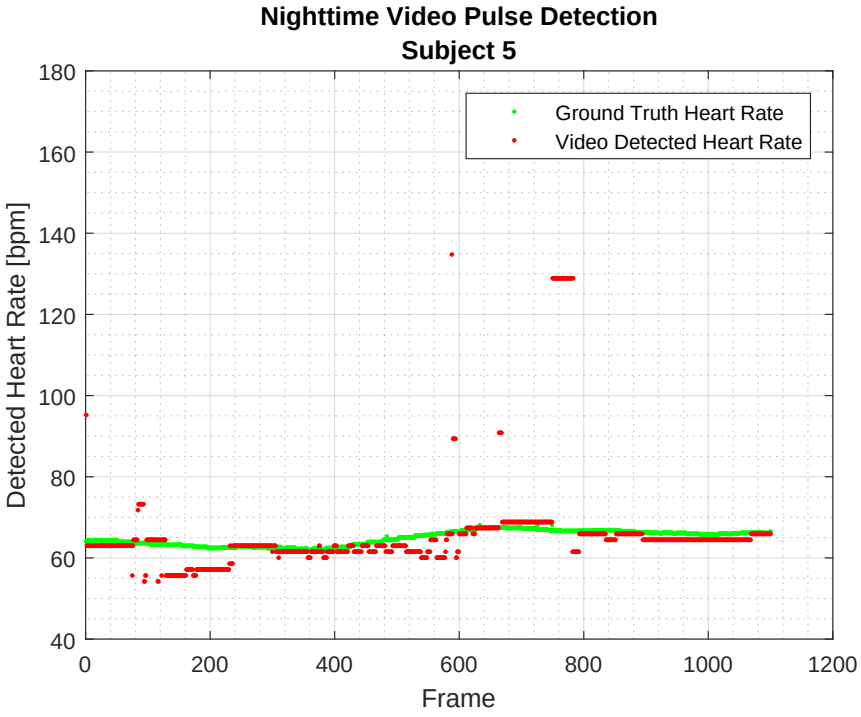
A.3 Nighttime Video Pulse Detection Plots

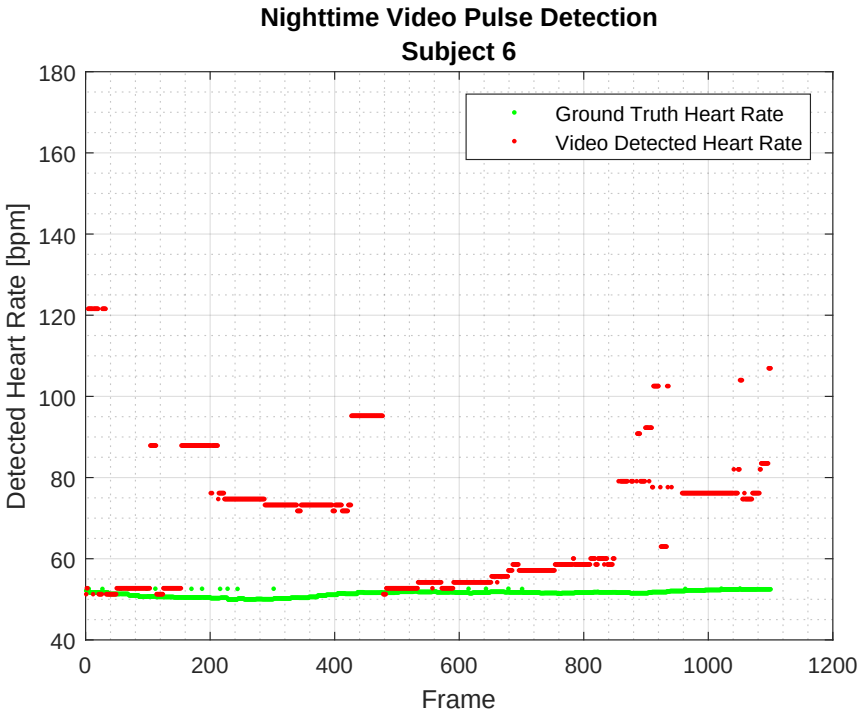


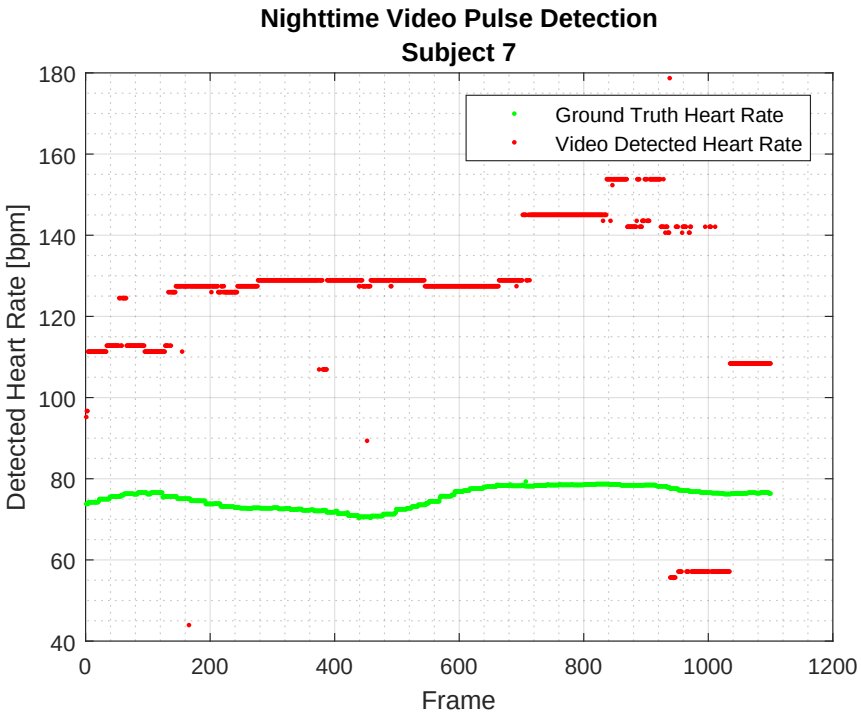


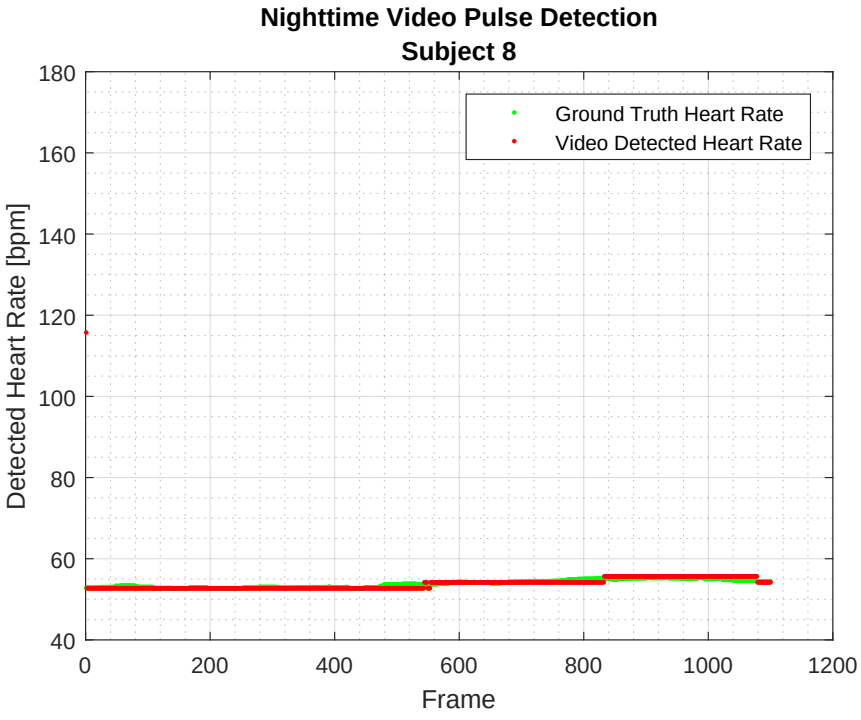

Nighttime Video Pulse Detection
Subject 3
Ground Truth Heart Rate
Video Detected Heart Rate
Detected Heart Rate [bpm]
Frame
180
160
140
120
100
80
60
40
0
200
400
600
800
1000
1200

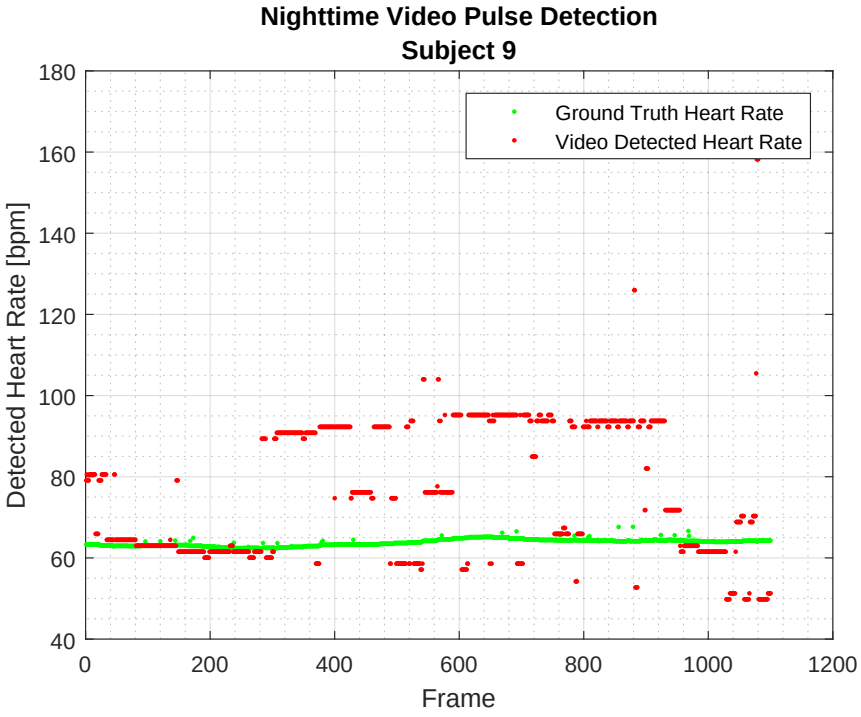

Nighttime Video Pulse Detection
Subject 4
Ground Truth Heart Rate
Video Detected Heart Rate
Detected Heart Rate [bpm]
Frame
180
160
140
120
100
80
60
40
0
200
400
600
800
1000
1200

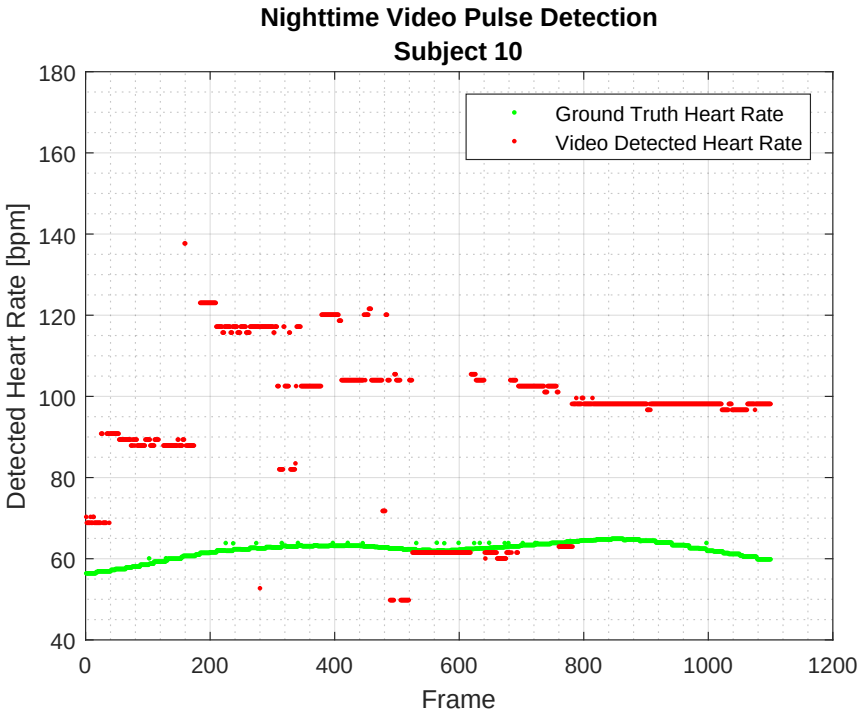

Nighttime Video Pulse Detection
Subject 5
Ground Truth Heart Rate
Video Detected Heart Rate
Detected Heart Rate [bpm]
Frame
180
160
140
120
100
80
60
40
0
200
400
600
800
1000
1200


Nighttime Video Pulse Detection
Subject 6
Ground Truth Heart Rate
Video Detected Heart Rate
Detected Heart Rate [bpm]
Frame
180
160
140
120
100
80
60
40
0
200
400
600
800
1000
1200


Nighttime Video Pulse Detection
Subject 7
Ground Truth Heart Rate
Video Detected Heart Rate
Detected Heart Rate [bpm]
Frame
180
160
140
120
100
80
60
40
0
200
400
600
800
1000
1200


Nighttime Video Pulse Detection
Subject 8
Ground Truth Heart Rate
Video Detected Heart Rate
Detected Heart Rate [bpm]
Frame
180
160
140
120
100
80
60
40
0
200
400
600
800
1000
1200


Nighttime Video Pulse Detection
Subject 9
Ground Truth Heart Rate
Video Detected Heart Rate
Detected Heart Rate [bpm]
Frame
180
160
140
120
100
80
60
40
0
200
400
600
800
1000
1200


Nighttime Video Pulse Detection
Subject 10
Ground Truth Heart Rate
Video Detected Heart Rate
Detected Heart Rate [bpm]
Frame
180
160
140
120
100
80
60
40
0
200
400
600
800
1000
1200

Bibliography

[AFO03] Okan Arikan, David A. Forsyth, and James F. O'Brien. Motion synthesis from annotations. *ACM Trans. Graph.*, 22:402–408, July 2003.

[BDG13] Guha Balakrishnan, Fredo Durand, and John Guttag. Detecting pulse from head motions in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3430–3437, 2013.

[BDSS01] Aaron F. Bobick, James W. Davis, Ieee Computer Society, and Ieee Computer Society. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:257–267, 2001.

[BI04] Ling Bao and Stephen Intille. Activity recognition from user-annotated acceleration data. In Alois Ferscha and Friedemann Mattern, editors, *Pervasive Computing*, volume 3001 of *Lecture Notes in Computer Science*, pages 1–17. Springer Berlin / Heidelberg, 2004.

[BKZW11] Jan Baumann, Björn Krüger, Arno Zinke, and Andreas Weber. Data-driven completion of motion capture data. In *Workshop on Virtual Reality Interaction and Physical Simulation (VRIPHYS)*. Eurographics Association, December 2011.

[BL16] M Burke and J Lasenby. Estimating missing marker positions using low dimensional kalman smoothing. *Journal of biomechanics*, 49(9):1854–1858, 2016.

[BMB+11] Andreas Baak, Meinard Müller, Gaurav Bharaj, Hans-Peter Seidel, and Christian Theobalt. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *IEEE 13th International Conference on Computer Vision (ICCV)*, pages 1092–1099. IEEE, November 2011.

[BSP+04] Jernej Barbič, Alla Safonova, Jia-Yu Pan, Christos Faloutsos, Jessica K. Hodgins, and Nancy S. Pollard. Segmenting motion capture data into

distinct behaviors. In Wolfgang Heidrich and Ravin Balakrishnan, editors, *Proceedings of the Graphics Interface 2004 Conference*, pages 185–194. Canadian Human-Computer Communications Society, 2004.

[BWKW14] Jan Baumann, Raoul Wessel, Björn Krüger, and Andreas Weber. Action graph: A versatile data structure for action recognition. In *GRAPP 2014 - International Conference on Computer Graphics Theory and Applications*. SCITEPRESS, January 2014.

[Car04] Carnegie Mellon University Graphics Lab. CMU Motion Capture Database, 2004. `mocap.cs.cmu.edu`.

[CB95] L. W. Campbell and A. F. Bobick. Recognition of human body motion using phase space constraints. In *Proceedings of the Fifth International Conference on Computer Vision*, ICCV '95, pages 624–, Washington, DC, USA, 1995. IEEE Computer Society.

[CBC+08] Tanzeem Choudhury, Gaetano Borriello, Sunny Consolvo, Dirk Haehnel, Beverly Harrison, Bruce Hemingway, Jeffrey Hightower, Pedja Klasnja, Karl Koscher, Anthony LaMarca, James A. Landay, Louis LeGrand, Jonathan Lester, Ali Rahimi, Adam Rea, and Danny Wyatt. The mobile sensing platform: An embedded system for activity recognition. *Appears in IEEE Pervasive Magazine - Special Issue on Activity-Based Computing*, 7(2):32–41, April 2008.

[CH05] Jinxiang Chai and Jessica K. Hodgins. Performance animation from low-dimensional control signals. *ACM Trans. Graph.*, 24:686–696, July 2005.

[CL11] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[Cor09] Thomas H Cormen. *Introduction to algorithms*. MIT press, 2009.

[dATS06] Edilson de Aguiar, Christian Theobalt, and Hans-Peter Seidel. Automatic learning of articulated skeletons from 3d marker trajectories. In *ISVC (2006)*, 2006.

[DU03] Klaus Dorfmüller-Ulhaas. Robust optical user motion tracking using a kalman filter. Technical report, Universitätsbibliothek der Universität Augsburg, Universitätsstr. 22, 86159 Augsburg, 2003.

[FAA+14] Robert S Fisher, Carlos Acevedo, Alexis Arzimanoglou, Alicia Bogacz, J Helen Cross, Christian E Elger, Jerome Engel, Lars Forsgren, Jacqueline A French, Mike Glynn, et al. Ilae official report: a practical clinical definition of epilepsy. *Epilepsia*, 55(4):475–482, 2014.

[Fed13] Peter Andreas Federolf. A novel approach to solve the "missing marker problem" in marker-based motion analysis that exploits the segment coordination patterns in multi-limb motion data. *PloS one*, 8(10):e78689, 2013.

[FXZ+14] Yinfu Feng, Jun Xiao, Yueting Zhuang, Xiaosong Yang, Jian J Zhang, and Rong Song. Exploiting temporal stability and low-rank structure for motion capture data refinement. *Information Sciences*, 277:777–793, 2014.

[GAG+00] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet. *Circulation*, 101(23):e215–e220, 2000.

[Gle98] Michael Gleicher. Retargetting motion to new characters. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 33–42, New York, NY, USA, 1998. ACM.

[GMHP04] Keith Grochow, Steven L. Martin, Aaron Hertzmann, and Zoran Popović. Style-based inverse kinematics. *ACM Transactions on Graphics*, 23(3):522–531, 2004. SIGGRAPH 2004.

[HFP+00] L. Herda, Pascal Fua, Ralf Plankers, Ronan Boulic, and Daniel Thalmann. Skeleton-based motion capture for robust reconstruction of human motion. In *Computer Animation*, Philadelphia, USA, 2000.

[HPE07] Christian Hoppe, Annkathrin Poepel, and Christian E Elger. Epilepsy: accuracy of patient seizure counts. *Archives of neurology*, 64(11):1595–1599, 2007.

[HS37] Alrick B Hertzman and CR Spealman. Observations on the finger volume pulse recorded photoelectrically. *Am J Physiol*, 119(334):e5, 1937.

[KGP02] Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. *ACM Transactions on Graphics*, 21(3):473–482, 2002. SIGGRAPH 2002.

[KLLK10] A M Khan, Young-Koo Lee Young-Koo Lee, S Y Lee, and Tae-Seong Kim Tae-Seong Kim. A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer. *IEEE transactions on information technology in biomedicine a publication of the IEEE Engineering in Medicine and Biology Society*, 14(5):1166–1172, 2010.

[KTWZ10] Björn Krüger, Jochen Tautges, Andreas Weber, and Arno Zinke. Fast local and global similarity searches in large motion capture databases. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '10, pages 1–10, Madrid, Spain, July 2010. Eurographics Association.

[KWM11] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. Activity recognition using cell phone accelerometers. *SIGKDD Explor. Newsl.*, 12(2):74–82, March 2011.

[LC10] Hui Lou and Jinxiang Chai. Example-based human motion denoising. *IEEE Transactions on Visualization and Computer Graphics*, 16:870–879, 2010.

[LCB06] Beno Le Callennec and Ronan Boulic. Robust kinematic constraint detection for motion data. In *Proceedings of ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, sept 2006.

[LCZP14] Xiaobai Li, Jie Chen, Guoying Zhao, and Matti Pietikainen. Remote heart rate measurement from face videos under realistic situations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4264–4271, 2014.

[LK81] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.

[LM06] G. Liu and L. McMillan. Estimation of missing markers in human motion capture. *The Visual Computer*, 22(9):721–728, 2006.

[LMPF10] Lei Li, James McCann, Nancy Pollard, and Christos Faloutsos. Bolero: a principled technique for including bone length constraints in motion capture occlusion filling. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '10, pages 179–188, Aire-la-Ville, Switzerland, Switzerland, 2010. Eurographics Association.

[LSL+03] Fritz Leutmezer, Christiana Schernthaner, Stefanie Lurger, Klaus Pötzelberger, and Christoph Baumgartner. Electrocardiographic changes at the onset of epileptic seizures. *Epilepsia*, 44(3):348–354, 2003.

[MM01] George B Moody and Roger G Mark. The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):45–50, 2001.

[MRC+07] Meinard Müller, Tido Röder, Michael Clausen, Bernhard Eberhardt, Björn Krüger, and Andreas Weber. Documentation: Mocap Database HDM05. Computer Graphics Technical Report CG-2007-2, Universität

Bonn, june 2007. Data available at `www.mpi-inf.mpg.de/resources/HDM05`.

[MSSD06] U. Maurer, A. Smailagic, D.P. Siewiorek, and M. Deisher. Activity recognition and monitoring using multiple sensors on different body positions. In *Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on*, pages 4 pp. –116, april 2006.

[NNSH11] Naoki Numaguchi, Atsushi Nakazawa, Takaaki Shiratori, and Jessica K. Hodgins. A puppet interface for retrieval of motion capture data. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, August 2011.

[OCK+14] Ferda Ofli, Rizwan Chaudhry, Gregorij Kurillo, René Vidal, and Ruzena Bajcsy. Sequence of the most informative joints (smij): A new representation for human skeletal action recognition. *Journal of Visual Communication and Image Representation*, 25(1):24–38, 2014.

[PGKT10] Christian Plagemann, Varun Ganapathi, Daphne Koller, and Sebastian Thrun. Real-time identification and localization of body parts from depth images. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3108–3113. IEEE, 2010.

[PKJA16] Jaromir Przybyło, Eliasz Kańtoch, Mirosław Jabłoński, and Piotr Augustyniak. Distant measurement of plethysmographic signal in various lighting conditions using configurable frame-rate camera. *Metrology and Measurement Systems*, 23(4):579–592, 2016.

[PMP10] Ming-Zher Poh, Daniel J McDuff, and Rosalind W Picard. Non-contact, automated cardiac pulse measurements using video imaging and blind source separation. *Optics express*, 18(10):10762–10774, 2010.

[PMP11] Ming-Zher Poh, Daniel J McDuff, and Rosalind W Picard. Advancements in noncontact, multiparameter physiological measurements using a webcam. *IEEE Transactions on Biomedical Engineering*, 58(1):7–11, 2011.

[RDML05] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L. Littman. Activity recognition from accelerometer data. In *Proceedings of the 17th conference on Innovative applications of artificial intelligence - Volume 3*, IAAI'05, pages 1541–1546. AAAI Press, 2005.

[RKH11] Michalis Raptis, Darko Kirovski, and Hugues Hoppe. Real-time classification of dance gestures from skeleton animation. In *Symposium on Computer Animation*, pages 147–156, 2011.

[RPMD12] Claus Reinsberger, David L Perez, Melissa M Murphy, and Barbara A Dworetzky. Pre-and postictal, not ictal, heart rate distinguishes complex partial and psychogenic nonepileptic seizures. *Epilepsy & Behavior*, 23(1):68–70, 2012.

[Sha49] Claude Elwood Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949.

[SLC04] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: A local svm approach. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3 - Volume 03*, ICPR '04, pages 32–36, Washington, DC, USA, 2004. IEEE Computer Society.

[SLM+12] Christopher G Scully, Jinseok Lee, Joseph Meyer, Alexander M Gorbach, Domhnull Granquist-Fraser, Yitzhak Mendelson, and Ki H Chon. Physiological parameter monitoring from optical recordings with a mobile phone. *IEEE Transactions on Biomedical Engineering*, 59(2):303–306, 2012.

[THC15] Cheen-Hau Tan, JunHui Hou, and Lap-Pui Chau. Motion capture data recovery using skeleton constrained singular value thresholding. *The Visual Computer*, 31(11):1521–1532, 2015.

[TK91] Carlo Tomasi and Takeo Kanade. *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ. Pittsburgh, 1991.

[TMD+07] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, et al. Stanley: The robot that won the darpa grand challenge. *The 2005 DARPA Grand Challenge*, pages 1–43, 2007.

[TVG+14] L Tarassenko, M Villarroel, A Guazzi, J Jorge, DA Clifton, and C Pugh. Non-contact video-based vital sign monitoring using ambient light and auto-regressive models. *Physiological measurement*, 35(5):807, 2014.

[TZK+11] Jochen Tautges, Arno Zinke, Björn Krüger, Jan Baumann, Andreas Weber, Thomas Helten, Meinard Müller, Hans-Peter Seidel, and Bernd Eberhardt. Motion reconstruction using sparse accelerometer data. *ACM Trans. Graph.*, 30:18:1–18:12, May 2011.

[Vic] Vicon Motion Capture Systems. `http://www.vicon.com/`.

[VJ01] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[VJ04] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[VSN08] Wim Verkruysse, Lars O Svaasand, and J Stuart Nelson. Remote plethysmographic imaging using ambient light. *Optics express*, 16(26):21434–21445, 2008.

[VSZ15a] Jakub Valcik, Jan Sedmidubsky, and Pavel Zezula. Assessing similarity models for human-motion retrieval applications. *Computer Animation and Virtual Worlds*, 2015.

[VSZ15b] Jakub Valcik, Jan Sedmidubsky, and Pavel Zezula. Improving kinect-skeleton estimation. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 575–587. Springer, 2015.

[Wel67] Peter D Welch. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on audio and electroacoustics*, 15(2):70–73, 1967.

[WMvdS05] Fokko P Wieringa, Frits Mastik, and Antonius FW van der Steen. Contactless multiple wavelength photoplethysmographic imaging: a first step toward "spo2 camera" technology. *Annals of biomedical engineering*, 33(8):1034–1041, 2005.

[Wor] World health organization. epilepsy. (2012). `http://www.who.int/mediacentre/factsheets/fs999/en/index.html`. Accessed: 2016-11-08.

[WPC05] Danny Wyatt, Matthai Philipose, and Tanzeem Choudhury. Unsupervised activity recognition using automatically mined common sense. In *Proceedings of the 20th national conference on Artificial intelligence - Volume 1*, AAAI'05, pages 21–27. AAAI Press, 2005.

[WRDF13] Neal Wadhwa, Michael Rubinstein, Frédo Durand, and William T Freeman. Phase-based video motion processing. *ACM Transactions on Graphics (TOG)*, 32(4):80, 2013.

[WRS+12] Hao-Yu Wu, Michael Rubinstein, Eugene Shih, John Guttag, Frédo Durand, and William Freeman. Eulerian video magnification for revealing subtle changes in the world. *ACM Trans. Graph.*, 31(4):65:1–65:8, July 2012.

[XFH11] Jun Xiao, Yinfu Feng, and Wenyuan Hu. Predicting missing markers in human motion capture using l1-sparse representation. *Computer Animation and Virtual Worlds*, 22(2-3):221–228, 2011.

[XNM+16] Ying Xu, Dennis Nguyen, Armin Mohamed, Cheryl Carcel, Qiang Li, Mansur A Kutlubaev, Craig S Anderson, and Maree L Hackett. Frequency of a false positive diagnosis of epilepsy: A systematic review of observational studies. *Seizure*, 41:167–174, 2016.

[XSZF16] Guiyu Xia, Huaijiang Sun, Guoqing Zhang, and Lei Feng. Human motion recovery jointly utilizing statistical and kinematic information. *Information Sciences*, 339:189–205, 2016.

[ZHCS08] Jia Zheng, Sijung Hu, Vassilios Chouliaras, and Ron Summers. Feasibility of imaging photoplethysmography. In *BioMedical Engineering and Informatics, 2008. BMEI 2008. International Conference on*, volume 2, pages 72–75. IEEE, 2008.