Pathological Tremor Detection From Video

Xilin Li

Delft University of Technology

Pathological Tremor Detection From Video

Master's Thesis in Embedded Systems

Computer Vision group
Faculty of Electrical Engineering, Mathematics, and Computer Science
Delft University of Technology

Xilin Li

30th August 2017

Author

Xilin Li

Title

Pathological Tremor Detection From Video

MSc presentation

August, 2017

Graduation Committee

Prof. Dr. Boudewijn Lelieveldt (chair)	Delft University of Technology
Dr. Jan van Gemert	Delft University of Technology
Dr. Stjepan Picek	Delft University of Technology
Dr. Silvia Pintea	Delft University of Technology

Abstract

A pathological tremor is an involuntary and periodic motion of a body part. The detection and quantification of a pathological tremor are essential for diagnosis and therapy. The goal of this research is to detect the frequency of the pathological tremor. Instead of detecting tremors by using a specific medical device, we propose a new architecture jointly using a state-of-the-art pose estimation method and periodicity detection technology to identify pathological tremors from a video. In our approach, an advanced deep neural network is deployed for human pose estimation. A pixel-wise method for frequency estimation is designed to spatially integrate the spectral information of pixels to refine an estimate.

Compared with conventional methods, our method offers significant convenience for both patients and medical staff. Our approach does not need a specific device. Thus it eliminates the error caused by the additional mass of the sensor [22]. The procedure of the test is simple for non-technical staff so that the method decreases the possible operational error.

Each module is evaluated on a real dataset by a series of experiments. Compared with a classic 1D surrogate signal method, our pixel-wise method has a smaller error and deviation on both synthetic videos and real videos. The architecture is finally evaluated on patient videos and shows a promising result. For 21 periodic videos, 13 of our frequency estimations have an absolute error lower than 1 Hz.

Preface

I would like to thank my thesis supervisors Dr. Jan van Gemert and Dr. Silvia Pintea at the Computer Vision group, Delft University of Technology. Both of them offered valuable guidance and resources for my project. I would also like to thank technicians and medical staff in Leiden University Medical Center, who provided the key test data and the feedback for the project. Their help was essential for the accomplishment of the thesis.

Xilin Li

Delft, the Netherlands
30th August 2017

Contents

Chapter 1

Introduction

In this chapter, Section 1.1 introduces the background and motivations for developing the work. Section 1.2 defines the problem that our research will tackle and Section 1.3 lists related works. Finally, Section 1.4 presents the basic idea behind our solution and our contributions.

1.1 Background

A tremor is defined as an involuntary rhythmic movement of a body part [6]. As a key feature of some diseases like Parkinson, pathological tremors affect millions of patients. Before finding a cure, an accurate and objective quantification of pathological tremors provides valuable information for diagnosis and therapy. Among all measurements for characterizing tremors, one primary parameter is frequency.

To detect the frequency of a pathological tremor, a tremor signal is collected and analyzed. The power spectrum of the signal is informative. Compared to a physiological tremor, a pathological tremor is more regular, less noisy and containing sharp concentrations in the power spectrum. A broad spectrum indicates that many different frequency components contribute to the spectrum, while a sharp peak shows that only one dominant frequency for the tremor exists [4].

Conventionally, sensors are attached to a patient's body skin to collect force, displacement or acceleration data and spectrum analysis is performed to get the frequency of the pathological tremor. Electromyography (EMG) is another widely used technique. It offers rich information for tremor frequency and motor unit synchronization [4]. Recently three-dimensional cameras like the Leap [14] are used to record position and acceleration data. These cameras do not require physical contact and result in more accurate estimations.

Perhaps the biggest disadvantage of the conventional methods is that all of them need a specific medical hardware for data collection. Besides, the complicated procedure increases the possibility of an operational error. In this research, we propose to detect the frequency of the pathological tremor from a video. All one needs is a commodity camera with at least HD resolution. Depending on the patient, the frequency of a pathological tremor varies from 2Hz to 12Hz [1]. According to NTSC and PAL television system, the frame rate of a video file is more than 25. Based on the Nyquist theorem, it is possible to detect tremor frequency directly from a video signal.

1.2 Research Problem

Given a video in which a patient performs specific actions, the task of this research is to identify and quantify the pathological tremors of body joints. As one can imagine, the first step of a naive solution is to locate the target joint in a video. Then we collect the tremor signal and transform the signal to the frequency domain. Finally, the periodicity is detected and the frequency is estimated. Following the idea, the task can be split into two main parts.

- Each body joint should be estimated correctly from the frame.
- A rule should be defined to tell the existence of a pathological tremor. A method should be proposed to quantify the tremor frequency along time.

Given the diversities of body profile and pose shape, detecting joints from an image is extremely hard. And for real videos, varieties of backgrounds add more complexities to the problem. As far as we know, there is no completely reliable algorithm or technology available that can perform the task. With the development of pattern recognition technology, for example, neural networks, the problem now becomes more popular and a solution is possible if the target pose is consistent with the training pose data.

The detection and quantification of a pathological tremor in a video are similar to a periodic motion analysis problem, where the scale of a tremor is tinier and the frequency is higher. A tiny action is hard to capture since the signal can be easily covered by the noise. The problem becomes more complicated when the tiny action is accompanied with a large motion. And some tremors only happen when a patient performs a particular action. For example, a tremor appears when a patient raises his hand.

All in all, it is still an open question to detect and quantify the tremor from a video. No existing model is provided as a solution.

1.3 Related Work

As far as we know, there is no existing model for the whole problem, but there exists well-performing approaches for each individual part.

1.3.1 Pose Estimation

A pictorial structure (PS) model is a classic model used for articulated human pose estimation [11]. PS generates a tree graphical model for the spatial constraints between different body parts. The local prediction for a joint is refined by coherently learning the spatial information from body configuration. To improve the performance, some augmented tree models [9, 21, 24] are proposed to add edges to a tree structure or use multiple trees, such that it additionally captures occlusion, symmetric and long-range co-relations.

Since 2014, a number of approaches based on convolutional neural networks (CNN) for pose estimations are designed and supersede previous work [7, 16, 3, 17, 5, 19, 25]. Most of these approaches regress the image to belief maps and sequentially send the map to a graphical model. A partitioning and labeling formulation of joint candidates is generated by a dedicated CNN in [19]. And the candidates are suppressed and grouped to form the configuration of the human body.

The Pose Machine [20] provides a sequential framework to detect articulated human pose in an image. Based on an inference model, the Pose Machine takes advantage of interactive spatial information between different parts in the image for pose prediction. For example, an elbow part is a strong cue for predicting a shoulder part. In addition, a sequential modular model makes it convenient to be implemented with any feature extractor and predictor [25]. With the advantages of the Pose Machine, the Convolutional Pose Machine (CPM) [25] substitutes the feature computation module with a CNN. A sequential CNN architecture increases the receptive area with the depth of the network, which allows the network co-relate different body part predictions and thus improves the accuracy.

1.3.2 Frequency Estimation

There is also extensive research in the detection and quantification of periodic motions from videos. One of the most popular methods is to convert a video signal to one-dimensional surrogate signal and do analysis either in the frequency or time domain. For example, the boundary contour of the target object is computed in [12]. And the frequency is estimated by peak detection in the frequency domain using 1D statistic signal of the contours. Mutual information (MI) signal between the reference frame and other frames is computed in [26] and the frequency is estimated by analyzing the peaks of the MI time series. Compared with the frequency-domain approaches, a time-domain approach is not limited to the resolution that DFT can determine and it is able to deal with a periodic impulsive signal.

Another method widely used is self-similarity matrix (SSM) introduced by [8], while the similarity can be defined in different ways. [8] not only describes a theorem for period detection, but also proposes a robust method based on the 2D power spectrum of SSM to quantify the period. Compared with other methods, SSM approach is able to detect the periodicity when a video contains significant non-Gaussian noise or the period is not constant.

1.4 Proposed Solution

We propose a new architecture (see Figure 1.1) using a combination of a state-of-the-art pose estimation method and a periodicity detection technology to solve the problem. We name it Tremor Frequency Detector (TFD). The structure is designed as a pipeline consisting of three modules, a pose estimation module, a frequency estimation module and a visualization module (see Figure 1.1).

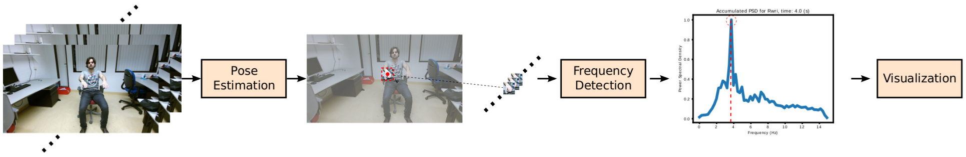


Figure 1.1: Architecture of TFD. A yellow box indicates a functional module. The red point labels the target joint and red dash box denotes the region of interest. Cropped frames form a time series proceeding to the next module. We make the frequency-domain analysis in the subsequent module. Finally, the results are visualized.

We utilize a deep neural network from [25] as our first module, since the model offers informative belief maps and achieves a competitive performance on our dataset. In the second module, we apply a frequency-domain approach like the method in [8]. However, instead of using a surrogate signal like similarity, we propose a pixel-wise method that spatially integrates the spectral information of pixels and estimates the frequency on the composite spectrum. The spatial information is generated by the pose estimation module. We design the approach based on the idea that the frequency estimation should focus on the signal around the target object and ignore the noise from the background.

We evaluate our work on real patient videos provided by Leiden University Medical Center (LUMC). The frequency estimation module is evaluated on both synthetic videos and real videos. And we compare our work with one of the similarity methods from [8], since it also achieves a competitive performance on our dataset.

The main contributions of this work include:

- designing a new architecture for detecting and quantifying human pathological tremors from videos,
- proposing a spatial-temporal method for estimating the frequency of the periodic motion from videos,
- creating a series of experiments to evaluate each module and the whole architecture.

This research provides the first complete solution for the problem described in Section 1.2. Compared with conventional methods, the approach increases the convenience and simplifies the operations. It offers patients a comfortable test that can be easily done anywhere with a common device. Besides, our approach also eliminates the error caused by the additional mass of the sensor [22] and achieves a promising result on our dataset. Furthermore, if extended, our approach could be used for real-time multi-patient monitoring, which saves the cost for medical devices and labors.

Chapter 2

Method

This chapter describes the methodology of our work. Section 2.1 explains the structure of the whole design. Then we explain the principals behind each module. Section 2.2 describes how body pose is estimated by using a deep neural network. It starts from the original model and then introduces the CNN-based model. Section 2.3 presents how we generate a time series and detect the frequency.

2.1 Architecture

We design the architecture of TFD as the following pipeline (Figure 2.1). A video file is taken as the input of the pipeline. The pose estimation module predicts the locations of different body parts and generates belief maps telling the confidence of the predictions. Based on the predictions, the second module crops the frame and creates a time series consisting of cropped joint boxes. Then it detects the existence of a tremor and computes its frequency. Finally, for the convenience of diagnosis, the results are analyzed and visualized in a web page. The visualization module is implemented by using Bootstrap framework and D3.js library, which will not be covered in this chapter. A visualization example of final results is shown in Figure B.1.

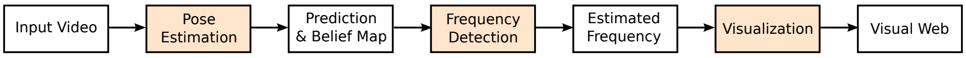


Figure 2.1: Structure of TFD. A white box represents the input or output of a module and a yellow box indicates a functional module.

2.2 Pose Estimation

The goal of this module to estimate the location of human body parts in a video. We make use of the Convolutional Pose Machine from [25] since it offers informative belief maps and shows competitive performance on several state-of-the-art human pose datasets like LEEDS [15] and MPII[2].

2.2.1 Pose Machines

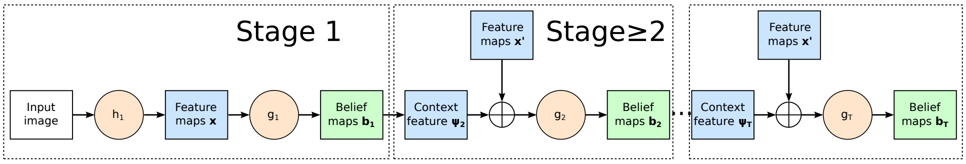

Stage 1
Input image
h_1
Feature maps $\mathbf{x}$
g_1
Belief maps $\mathbf{b_1}$
Stage≥2
Feature maps $\mathbf{x'}$
Context feature ψ_2
g_2
Belief maps $\mathbf{b_2}$
Feature maps $\mathbf{x'}$
Context feature ψ_T
g_T
Belief maps $\mathbf{b_T}$

Figure 2.2: Structure of the pose machine [25]. The leftmost inset shows the first stage operating only on the feature maps. The subsequent inset presents the second stage operating on both the feature maps and the belief maps from the previous stage. The third inset shows a repeated structure for following stages.

The pose machine is a sequential model composed of a series of multi-class predictors [20]. The structure is shown in Figure 2.2, which consists of T stages. In stage $t \in \{1...T\}$, the predictor g_t outputs belief maps $\mathbf{b_t}$ for M joints and passes them to the next stage.

In the first stage, the feature extractor h_1 generates feature maps $\mathbf{x}$ from the input image and proceeds to the predictor g_1 to produce belief maps $\mathbf{b_1}$. To capture the spatial information between different body parts, the subsequent stage maps input belief maps $\mathbf{b_{t-1}}$ to contextual feature maps by function $\psi_t(\cdot)$, where $t \geq 2$. Based on the feature maps $\mathbf{x}'$ and the additional contextual information ψ_t, the classifier g_t refines the spatial information and makes new belief maps $\mathbf{b_t}$. For the feature maps $\mathbf{x}'$, they are not necessarily the same as the initial maps $\mathbf{x}$.

2.2.2 Convolutional Pose Machines

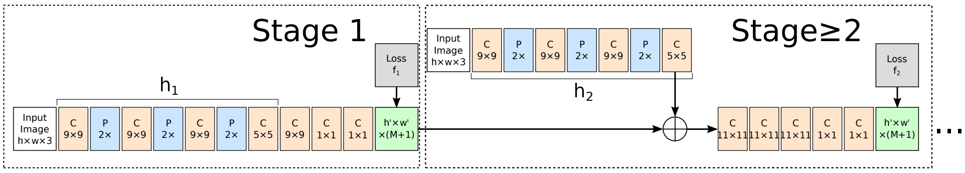


Figure 2.3: Structure of the CPM [25]. We show the convolutional structure of the CPM. The left inset shows that the first stage extracts features from the image evidence and regresses to belief maps. The right inset shows the repeated subsequent stages operating on the image evidence and belief maps.

The Convolutional Pose Machine [25] (see Figure 2.3) integrates convolutional neural network (CNN) to the structure of the pose machine. CNN equips the CPM a better capability of extracting features from both raw and contextual information. Furthermore, the differentiable property of CNN makes it possible to be trained end-to-end.

The first stage of the CPM consists of a feature extractor h_1 composed of four convolutional layers and a classifier composed of one convolutional layer and two 1×1 convolutional layers in fully convolutional structure. The stage outputs $M + 1$ belief maps representing the possibility that a joint exists at the location. L2 loss function f_1 supervises the learning progress to prevent 'gradient vanishing' problem. Subsequent stages are in the same structure as stage two. Based on the feature maps extracted by another sub-network and the additional belief maps from the last stage, the classifier of the next stage refines the information and generates new $M + 1$ belief maps.

In the whole structure, we feed the image forward the network and take the locations with the maximum confidence in the belief maps of the last stage $\mathbf{b_T}$ as joint predictions. The predictions and belief maps then proceed to the next module.

2.3 Frequency Estimation

The goal of this module is to detect and quantify the periodicity of the joint tremor in a video. We create spatial-temporal series and analyze the series in the frequency domain. The whole procedure is summarized as follows.

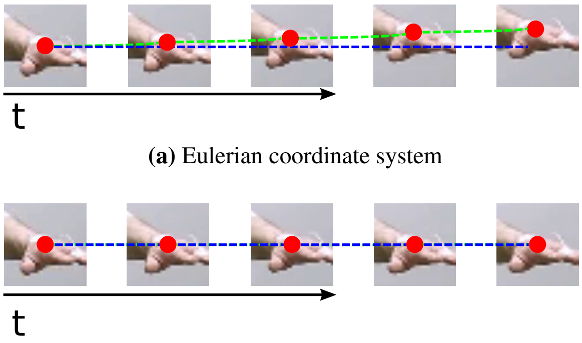
- Crop the frames based on the pose estimations and form an intensity series for each pixel in the cropped box.
- Preprocess the signal spatially and temporally.
- Perform the frequency-domain analysis.
- Integrate the frequency-domain signal spatially to one-dimensional signal.
- Determine the periodicity and frequency of the signal.

2.3.1 Cropping and Tracking

While the whole image includes too much redundant information and analyzing the entire image is computationally expensive, we crop the joint part based on the prediction and track its movement. Two coordinate systems can be used for cropping [27].

The Eulerian coordinate system (see Figure 2.4a) observes the target object from a fixed location. In a video, object movement is observed in a specific box whose position is fixed. In practice, for a sequence of frames, the target object is cropped based on the prediction from one reference frame.

The Lagrangian coordinate system (see Figure 2.4b) follows the movement of an object as it moves along time and space. For a sequence of frames, the target object is cropped based on the prediction for each frame. While the Eulerian method focuses on the target's mobility, the Lagrangian method pays more attentions to the object itself.



(a) Eulerian coordinate system

(b) Lagrangian coordinate system

Figure 2.4: Cropped frame series for two coordinate systems. The red point labels the center of the target object. The blue line connects the center of the frame and the green line connects the center of the target. While the Lagrangian coordinate locks the target, the Eulerian coordinate observes the movement of the target.

2.3.2 Preprocessing

Before analysis, each cropped frame is blurred by Gaussian function to reduce the noise. The two-dimensional Gaussian function is presented as Equation 2.1, where x and y are the coordinates and σ is the standard deviation of the Gaussian distribution. The function generates a circular surface satisfying Gaussian distribution, which is used to produce a convolution matrix. A new pixel value is computed as the weighted average of neighbour pixel values.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.1}$$

Then the signal is filtered by 4^{th}-order Butterworth band-pass filter to eliminate high- or low-frequency signal that is not of interest. Finally, the offset is extracted from the signal to remove direct-current part.

2.3.3 Time-Frequency Analysis

Given a discrete time series $x(n)$, Short-time Fourier Transform (STFT) [18] is applied to transform a signal from the time domain to the frequency domain. STFT decomposes a window of signal to a series of sinusoidal components, as the window slides over the time. The discrete-time STFT function is shown in Equation 2.2, where $x(n)$ is the signal to be transformed, $w(n)$ is the window function and m is the window shift.

$$X(m, f) = \sum_{n=-\infty}^{\infty} x(n)w(n-m)e^{-j2\pi fn} \tag{2.2}$$

Different from the similarity method proposed in [8], we apply STFT to each pixel intensity series and compute the power spectral density (PSD), thus called

’pixel-wise’ method. An adjustable window, Tukey window is chosen as our analysis window, because window size is important for the Eulerian approach and it is necessary that we pick a window that is flexible for different window sizes. The function of the Tukey window is presented in Equation 2.3, where $w(n)$ is the window function, N is the window size and α is the shape parameter indicating the proportion of the window inside the cosine tapered region [13].

$$w(n) = \begin{cases} \frac{1}{2}[1 + cos(\pi(\frac{2n}{\alpha(N-1)} - 1))] & \text{if } 0 \leq n < \frac{\alpha(N-1)}{2} \\ 1 & \text{if } \frac{\alpha(N-1)}{2} \leq n \leq (N-1)(1-\frac{\alpha}{2}) \\ \frac{1}{2}[1 + cos(\pi(\frac{2n}{\alpha(N-1)} - \frac{2}{\alpha} + 1))] & \text{if } 0 \leq n < \frac{\alpha(N-1)}{2} \end{cases} \tag{2.3}$$

2.3.4 Signal Integration

From the frequency-domain analysis, we get PSD for each pixel intensity series. We utilize the spatial information learned from the first module to integrate the frequency-domain information over the image. In detail, we use power spectrum **P** and belief map b in combination to generate a new PSD for the joint.

$P_{x,y}$ is the PSD generated by the time series of the pixel at the location (x, y) in the cropped box B. Since we only care the target in the frame, we normalize the PSD and calculate the accumulated PSD with belief $b_{x,y}$ as the weight as Equation 2.4.

$$P = \frac{\sum_{x,y \in B} P_{x,y} \cdot b_{x,y}}{\sum_{x,y \in B} b_{x,y}} \tag{2.4}$$

An advantage of this method is that it decreases the effect of the background signal and focuses on the region of interest.

2.3.5 Periodicity Detection and Quantification

To determine whether a sequence is periodic, we detect the frequency f_i with the maximum power from the power spectrum $P(f)$. A time series is estimated as periodic if f_i is dominant in the spectrum, and f_i is the final estimate. As in [8], we define a frequency f_i is dominant if

$$P(f_i) > \mu_P + K\sigma_P \tag{2.5}$$

where μ_P is the mean of the power spectrum P, K is a constant factor and σ_P is the standard deviation of P. K of 3 is recommended by the paper.

Chapter 3

Evaluation

This chapter describes how we evaluate our work. In Section 3.1, we describe the dataset used for the tests. Section 3.3 and Section 3.4 show the evaluation methods for the pose estimation module and the frequency detection module. The final test in Section 3.5 presents the general performance of the work.

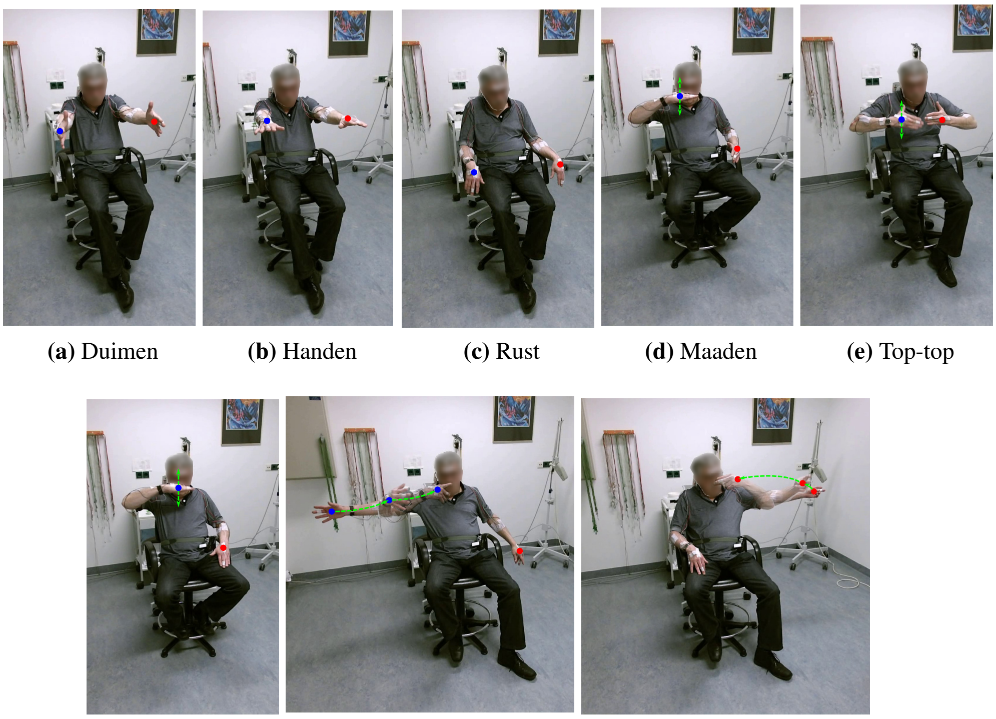
3.1 Dataset

35 videos for different patients offered by LUMC are utilized to test our approach. The frame size is 1920×1280 and the frame rate is 30 frames per second. Depending on the patient, either the left or right wrist is selected to detect the tremor frequency. An accelerometer is adhered to patient's wrist, as the red point in Figure 3.1. The collected acceleration data is analyzed as the benchmark (see Section 3.4 for further details). Two of these videos will be used to compare two proposed cropping approaches and two frequency detection approaches separately.

For each video, a patient performed 8 different actions as shown in Figure 3.1. Different actions are essential for detecting different kinds of tremors for diagnosis. According to [4], mainly two kinds of tremors are classified. One is a rest tremor, which happens when muscles are inactive. Another kind is an action tremor, which occurs when muscles are activated, for example, postural or action-specific tremors. Each video has been separated into independent parts based on different actions. Only one of these actions will be employed for testing the overall performance.

3.2 Experimental Set-up

We utilize a model pre-trained on 'MPII' dataset from [25] for pose estimation. The parameters for evaluating the frequency estimation module are listed in Table 3.1 and Table 3.2, for controlled experiments and real experiments separately. The frame color of the synthetic video is in gray scale.



(a) Duimen **(b)** Handen **(c)** Rust **(d)** Maaden **(e)** Top-top

(f) 100-7 **(g)** Top neus links **(h)** Top neus rechts

Figure 3.1: Example real videos. The patient did different actions in the video, which helps to collect comprehensive tremor data. A red point labels the detecting point of the accelerometer. A blue label and a green line shows the moving joint and its movement trajectory.

3.3 Pose Estimation

3.3.1 Evaluating Pose Estimation Methods

In this section, we describe how we evaluate the first module, the pose estimation module. The results are compared with another state-of-the-art work.

To evaluate the performance of the pose estimation, 15 frames are evenly sampled over time from each video. Some frames include external interference, for example when medical staff help to deploy the experiments, they inevitably cover patient's body. And this exceeds the scope of the test, so before the evaluation we manually filter these frames. Besides, the interference is also eliminated when segmenting the videos. Finally, the prediction results are compared with manual annotations.

To make a comparison, we introduce another work in the field and do the same test. The DeepCut CNN is an advanced network made by [19], which first generates a set of body-part hypothesis and then partitions and labels the joints. It is shown that the DeepCut has an outstanding performance on several different human pose datasets.

Property	**Value**	**Property**	**Value**
Frame Size	32×32 px	Window Size	121
Ball Size	8 px	Overlap	60
Ball Color	130	Tukey α	0.25
Background Color	220	Filter	2-14 Hz
Vibration Magnitude	8 px	Blur Kernel	5×5 px
Vibration Frequency	0-14 Hz	FPS	30 frame/sec
Gaussian noise σ	0-2.0	Disturbance	0-16 px

Table 3.1: Parameters for controlled experiments

Property	**Value**	**Property**	**Value**
Window Size	121	Overlap	60
Box Size	1 head size	Blur Kernel	5×5 px
Filter	2-14 Hz	Constant K	3
Tukey α	0.25		

Table 3.2: Parameters for real experiments

We use a state-of-the-art metric to describe the performance of the pose estimations, Percentage Correctness Keypoint (PCK) [2]. PCK computes the distance between the prediction and the ground truth and regards it as a match if the distance is lower than a threshold. PCKh is a modified metric that normalizes the error tolerance based on target's head size.

3.3.2 Pose Estimation Results

A PCKh curve over error tolerance is drawn in Figure 3.2. It can be seen that both approaches have an impressive performance on our data, approximately 73% correctness. Notice that each real video has its own filming angle and character, which makes the prediction hard. In addition, the benchmark is annotated by only one person who has some bias in the annotating pattern. If adding some diversities to the annotating pattern, the result is expected to be better.

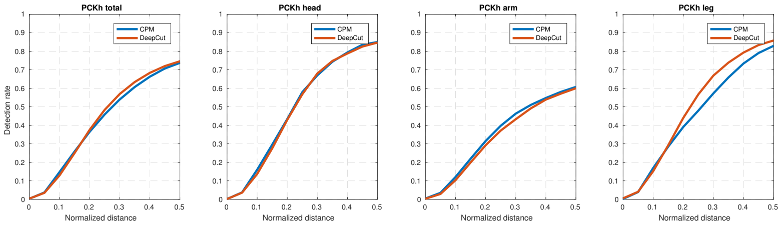
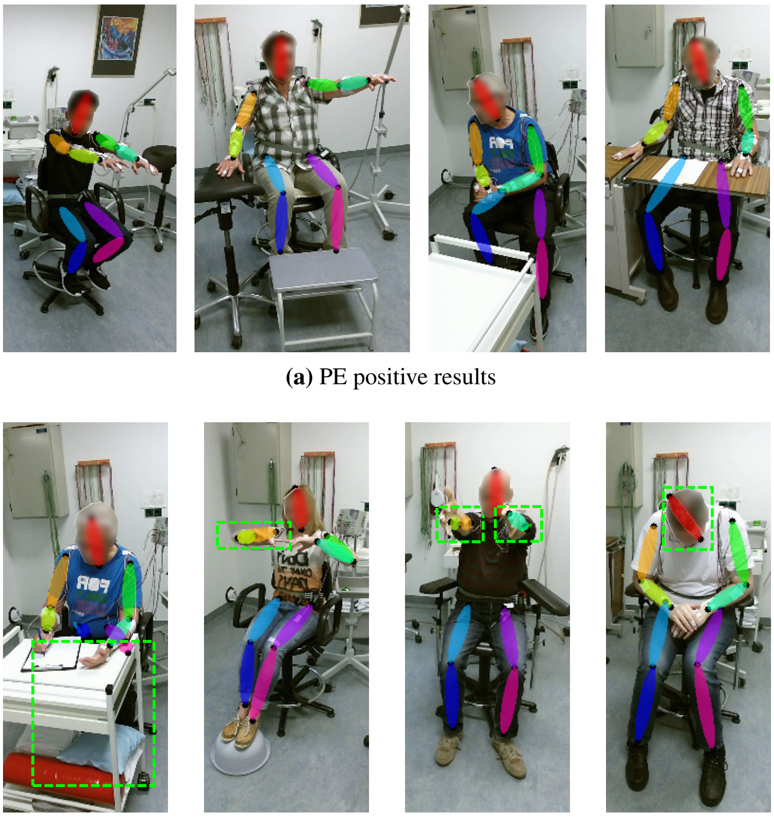


Figure 3.2: Quantitative results of pose estimation on customized dataset. Both approaches have an overall correctness of 73%. The CPM performs better than the DeepCut on upper-body estimations.

The CPM supersedes a little at upper body recognition, including an improvement of 0.7% for the arm and 0.3% for the head, and the DeepCut has a better capability of leg recognition, which has a 2.8% improvement.



(a) PE positive results

(b) PE negative results

Figure 3.3: Visualized results of pose estimations. Figure (a) shows four correct predictions made by the CPM for complex poses. Green boxes in Figure (b) show some typical mistakes caused by self-occlusion and object occlusion.

Some examples of correct predictions are shown in Figure 3.3a, which show that the CPM has an outstanding performance on our data. Even a relatively complicated pose can be accurately predicted, for example, the third example in Figure 3.3a. A slight change of perspective can not affect the final prediction, for example, the first and third example. An image with a small occlusion on the body can also be predicted, for instance, in the fourth example, part of the leg is covered by the desk.

Some typical mistakes are shown in Figure 3.3b. A large area of occlusion is still a problem for the prediction, which is reasonable since the lower part is totally covered and even a human is not able to annotate accurately. Self-occlusion is also a problem. Like the second and third example, the forearm covers the shoulder part. Filming angle is another factor. In the fourth example, the patient's face is not shown completely, for which the head top is not labeled correctly. In the experiments, we also found out that a lying pose was not able to be detected, because the training dataset does not include such pose.

We finally choose the CPM as our first module, because the CPM shows a competitive performance and belief maps offer rich location information of the target for further modules.

3.4 Frequency Estimation

In this section, we evaluate our frequency estimation method. Firstly, independent from the first module, controlled experiments are made to prove that our method is valid on the synthetic data under three different conditions. Then we make tests on the real data with different approaches described in Section 2.3. For both datasets, we compare our method with a classic frequency detection technique.

3.4.1 Controlled Test for Frequency Estimation

Test Methods

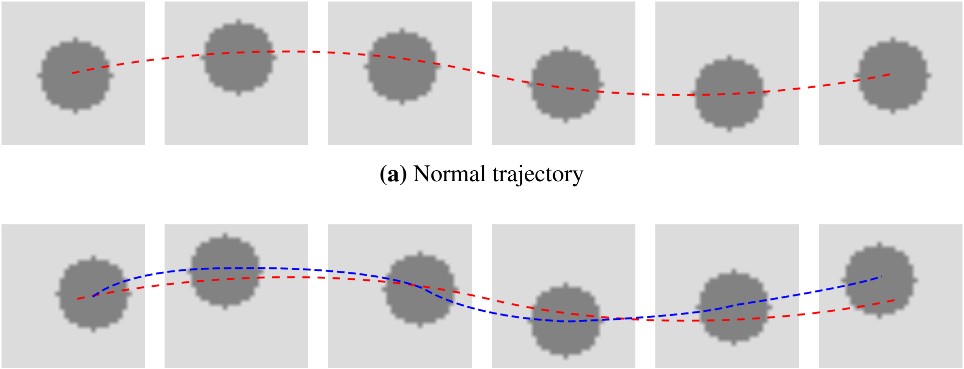
Synthetic videos in gray scale are made for controlled tests. A ball does simple harmonic vibration in the vertical direction as in the Figure 3.4a, the frequency is manually set. The movement trajectory follows a sine curve, which is set to simulate human tremors. In the experiments, we assume that the movement is tracked without error by using the Eulerian coordinate system. The confidence map is simulated by using the Gaussian distribution.

The frame size is referred to the size of a joint box cropped from a real video. The ball size is set to one-third of the whole frame size, which equals to the approximate proportion of a joint in the box. An interval from 2Hz to 14Hz is the range that can be detected in a video. It is also the range of the frequencies to be tested in the experiments.

We compare our pixel-wise frequency detection method with the similarity method proposed by [8], which converts 2D image signal $\mathbf{I}(\mathbf{x}, \mathbf{y})$ to 1D similarity information $\mathbf{S}$. We use the absolute correlation as the similarity metric as in the paper, which can be represented as the following function, for the moment t_1 and t_2:

$$S(t_1, t_2) = \sum_{x,y \in B} |I_{t_1}(x, y) - I_{t_2}(x, y)| \tag{3.1}$$

where $I_{t_i}(x, y)$ is the pixel intensity at position (x, y) at time t_i. It simply calculates the difference between two intensity matrices at different moments. Compared with our pixel-wise method, 1D similarity signal brings convenience to the frequency analysis, while for the pixel-wise method we need to consider how to integrate the information over the image and generate a global estimate. However, a drawback of the similarity method is that it drops the location information in the image.



(a) Normal trajectory

(b) Trajectory with disturbance

Figure 3.4: Movement trajectory of the synthetic video. A gray ball does simple harmonic motion in the frame box. Red line indicates a sine movement trajectory. In the lower figure, a blue line shows a disturbed sine movement trajectory.

Experiment 1: Synthetic Video

A typical power spectral density (PSD) for the normal case is shown in Figure 3.6a. In this case, two obvious peaks are shown in the spectrum. The higher peak is for the main frequency of the movement. Another peak is the harmonic component of the main frequency. In our tests, frequencies of 2Hz to 14Hz are detected correctly without any error by our method and the similarity method.

Experiment 2: Synthetic Video with Gaussian Noise

To further test the performance of the method, 6Hz is chosen as the main frequency of the tremor and Gaussian white noise is added to each frame in the video to simulate the real environment. For the statistic characteristics of the experiment, we run it for 100 times and compute the mean squared error (MSE).

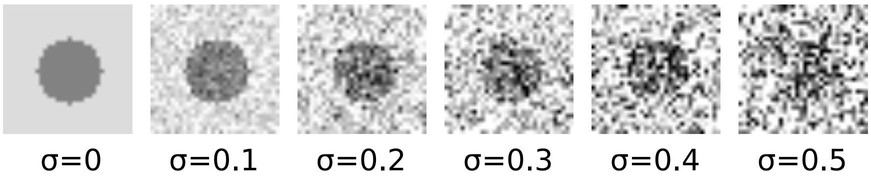


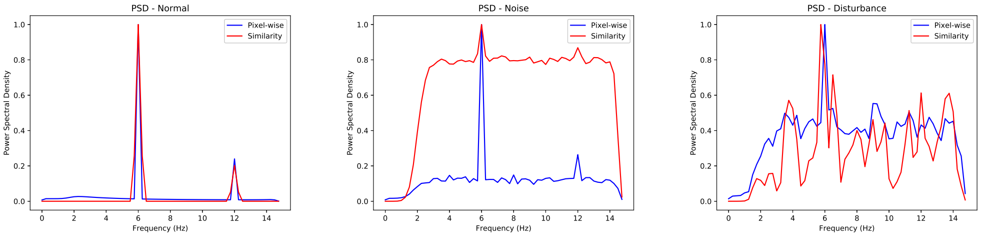
Figure 3.5: Synthetic video frame with Gaussian noise. The target object is ambiguous in a noisy frame.

For the synthetic video with noise, the mean of spectrum raises with the standard deviation σ of the Gaussian noise as shown in Figure 3.6b. The relation between the MSE and σ is shown in Figure 3.7a. The MSE increases with σ. A high deviation means that the pixel intensity is far from the original value as in Figure 3.5, which destroys the original time series and thereby makes it hard to detect the frequency of the signal. As a comparison, the similarity method has a poor performance on this kind of interference, because the noise greatly affects the similarity between two frames, thus affect the analysis of the time series in the frequency domain. However, our method looks at the general PSD for the whole frame and focus on the signal around the target object, which decreases the error from the noise in the background.

Experiment 3: Trajectory with Disturbance

In the third experiment, uniformly distributed disturbance on random direction is made to the motion. For each frame, a random shift in the vertical or horizontal direction is generated in the motion like in Figure 3.4b. The goal is to test the performance under the situation that the pose prediction or the camera shifts irregularly.

For the synthetic video with disturbance, the spectrum raises with the magnitude of the disturbance as Figure 3.6c. The spectrum is more erratic than that of the synthetic video with noise. In Figure 3.7b, the MSE increases with the magnitude of the disturbance. When the disturbance is significant, the spectrum is too noisy to distinguish the tremor frequency. This results from the truth that a large-scale deviation from original track destroys the periodic motion, which confuses the detector. Compared to the similarity method, our method has a more stable PSD and lower MSE, mainly because our method looks at the general weighted PSD, thus weaken the effect of the useless signal from the background.



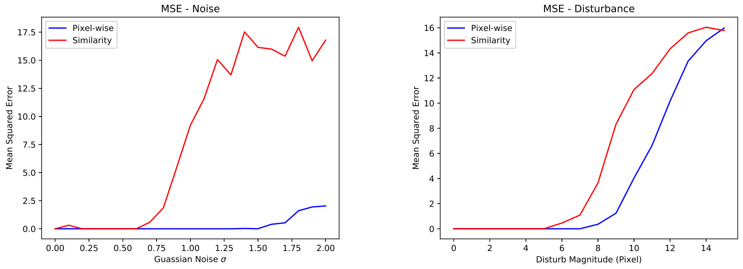
(a) Synthetic video **(b)** Synthetic video with noise **(c)** Synthetic video with disturbance

Figure 3.6: PSD for synthetic videos. We set the vibration frequency to 6Hz. For the normal case, both methods have an obvious peak in the spectrum. For a video with noise or disturbance, the PSD from the similarity method is significantly raised or erratic, but the pixel-wise method has a relatively stable and clean PSD.

3.4.2 Frequency Estimation on Real Data Examples

Frequency Estimation Methods

Synthetic video tests prove that the proposed detection method is valid in controlled environments. In the real test, the whole architecture is implemented and tested on



(a) MSE vs Gaussian Noise σ **(b)** MSE vs Disturbance Magnitude

Figure 3.7: MSE for synthetic videos. The MSE increases with the Gaussian noise deviation and the magnitude of the disturbance. In both tests, the pixel-wise method outperforms the similarity method.

real videos. It is compared with the benchmark computed from acceleration data. In the first experiment, two approaches based on different coordinate systems (see Section 2.3) are compared by using two selected periodic videos. The method with a better performance is taken as a part of the pipeline for further tests. Then our method is compared with the similarity method again.

Frequency Estimation Benchmark

3-axis acceleration data is collected from the sensor adhered on a patient's left or right wrist. The acceleration data is preprocessed and transformed to the frequency metric to be compared with the results from our approach. The process is summarized as follows.

- Filter the components that are lower than 2Hz or higher than 14Hz, by using 4^{th}-order Butterworth band-pass filter. This is to eliminate some physiological frequencies, for example, breathing.
- Extract the offset from the signal to remove the direct-current component.
- Perform STFT to the signal with a Tukey window.
- Compute the average PSD over time and determine the periodicity.
- Take the frequency with the maximum power as the estimation and compute the average of the 3-axis frequencies as the final result.

Experiment 1: Cropping Method Comparison

In the first experiment, two videos, video code 'T008' and 'T011', are selected to test the performance of two cropping methods based on the Eulerian and the Lagrangian coordinate systems. For both videos, we detect the tremor frequency

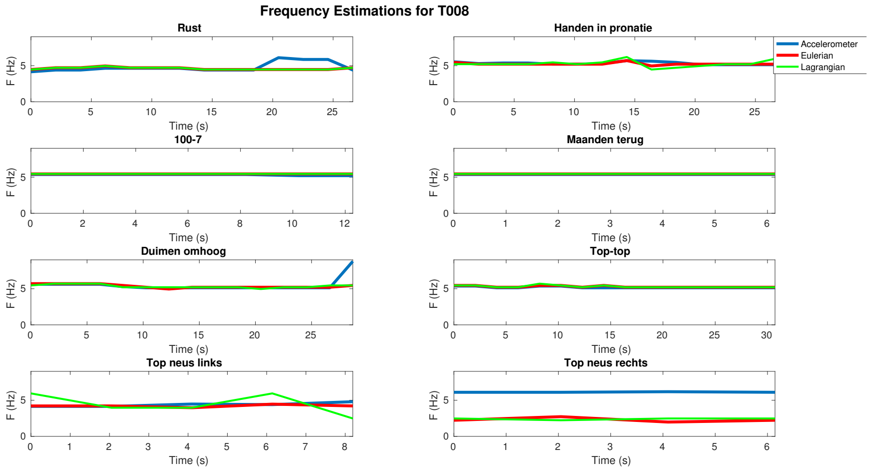


Figure 3.8: Frequency estimations of different cropping methods on 'T008'. Both methods have an impressive accuracy on 7 videos. The Eulerian approach has lower MSE for most videos. An exception is 'Top neus rechts' because of a large motion inside.

of the patient's right wrist. All 8 videos of 'T008' and 7 of 8 videos of 'T011' have been examined as periodic. 'Top neus links' of 'T011' is regarded as non-periodic. The frequency estimations from videos are compared with those from the accelerometer in Figure 3.8 and Figure 3.9. The mean squared error (MSE) for two videos are computed and presented in Figure 3.11a and Figure 3.11b. The standard deviation (STD) results are listed in Table A.1a and Table A.2a.

For video 'T008', the estimations of both cropping methods are accurate for most of the actions, while the Eulerian method has a slight advantage over the Lagrangian method. The only exception is 'Top neus rechts', in which patient waved the target arm. The error is significant since a large motion is hard to track for the Eulerian method. One will lose the target if the target moves out of the cropped box, which means that a good box size and a proper frame sequence length are important for the Eulerian approach.

The Lagrangian method has a better performance for the video 'Top neus rechts' since it always tracks joint's motion. However, the estimation is still far from correct, mainly because the perspective is changing with the movement (see Figure 3.10). And the shape of the joint is changing. However, our method assumes that the shape of the object is consistent. Our approach regards it another object when a different perspective is applied, which leads to a significant error of the frequency estimation. Besides, the background changes with the movement of the joint, which results in a significant disturbance for the pose estimation.

The estimations for the video 'T011' show the advantage of the Eulerian method, which has a smaller MSE in 6 cases. For the STD, the estimations from the Lagrangian method are not stable compared with the Eulerian method, because of the disturbance when tracking the target. When looking into some videos, we also

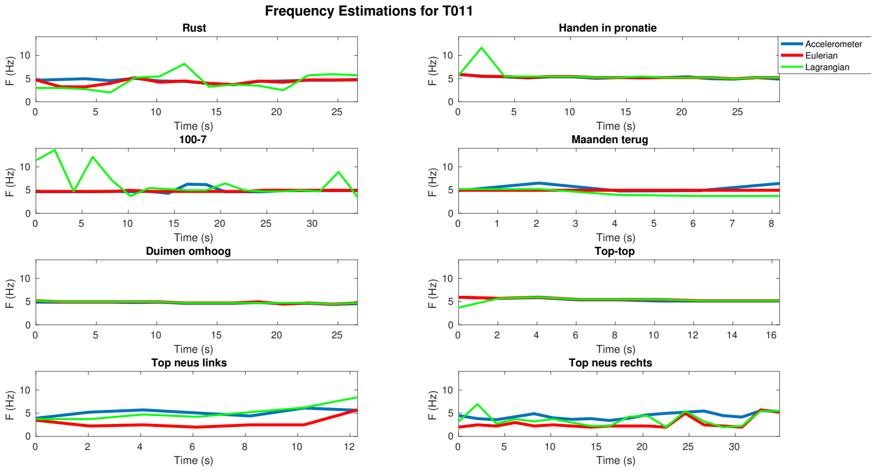


Figure 3.9: Frequency estimations of different cropping methods on 'T011'. Notice that the Eulerian method performs better in most of the videos, only with exceptions 'Top neus rechts' and 'Top neus links'.

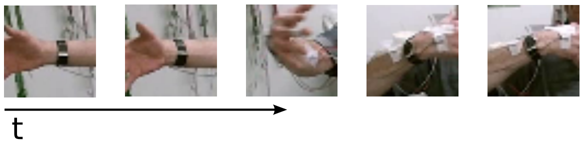


Figure 3.10: 'Top neus rechts' tracking pose example. The interval between two frames is 0.33s. Notice that pose estimations are well performed, but the perspective is changing so that the shape of the target object is changing.

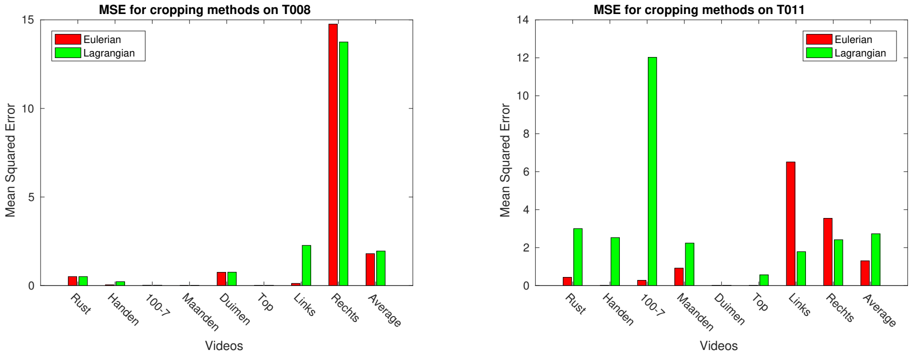
observe that a medical crew instructed or helped the patient to complete the action, which results in a wrong pose prediction and thereby affect the frequency estimation. Nevertheless, as 'T008', for a large motion like 'Top neus rechts', the estimation from tracking method has a better performance.

In most cases, the Eulerian method is better than the Lagrangian method, regarding the accuracy and the stability. For computation complexity, the Lagrangian method consumes more resources since it requires a pose prediction for each frame. Finally, the Eulerian method is taken as a part of the work.

Experiment 2: Parameter Selection

As described in Experiment 1, window size plays an important role in the Eulerian approach. In the previous experiment, we use a general number recommended by LUMC and it shows a good performance for most cases, except for a large-motion video. A basic idea to improve the performance for a large-motion video is decreasing the window size to limit the joint in the cropped box.

The MSE results for an additional test on window size on video 'T008' is shown



(a) 'T008' **(b)** 'T011'

Figure 3.11: MSE of different cropping methods. Generally, the Eulerian method has lower MSE, except for a large motion 'Top neus rechts'.

in Figure 3.12. The MSE shows an increasing trend with the window size. The curve is erratic because when the head or the tail of the signal does not include a complete period, it leaks part of the power and causes some error. A drawback of this method is that a small window size also decreases the resolution of the spectrum, and thus decreases the accuracy. This means that we should find a balance such that keeps spectrum resolution and also limit the motion.

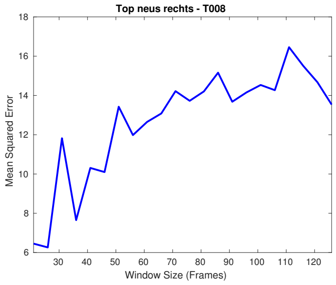


Figure 3.12: MSE vs Window size - T008 'Top neus rechts'. If a large motion is estimated by the Eulerian approach, the MSE increases with the window size.

To choose a proper window size for a large motion, we extract a window of frames and limit the joint in the cropped box. Thus we find an upper limit for the window size. To avoid the problem of spectrum leakage, we test different window sizes and observe the power spectrum. If a dominant peak exists, we regard the window size as a proper number.

Experiment 3: Performance Comparison

We compare our method with the similarity method as in the controlled experiment. Both methods use the same pose estimation module and cropping method. We evaluate two approaches on video ’T008’ and ’T011’. The estimations are presented in Figure 3.13 and Figure 3.14. The MSE results are shown in Figure 3.15a and Figure 3.15b. The STD results are listed in Table A.3a and Table A.4a.

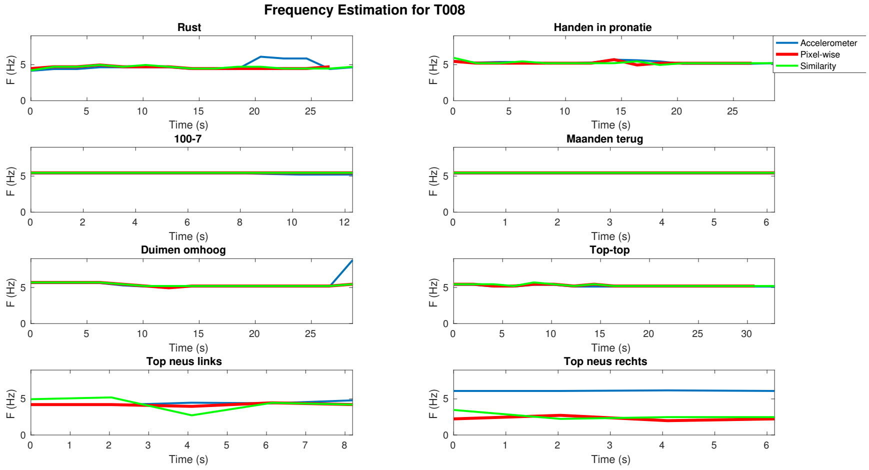


Figure 3.13: Frequency estimations of different approaches on ’T008’. Both approaches have an accurate estimation on 7 videos. And the pixel-wise method performs a bit better on ’Top neus links’ video.

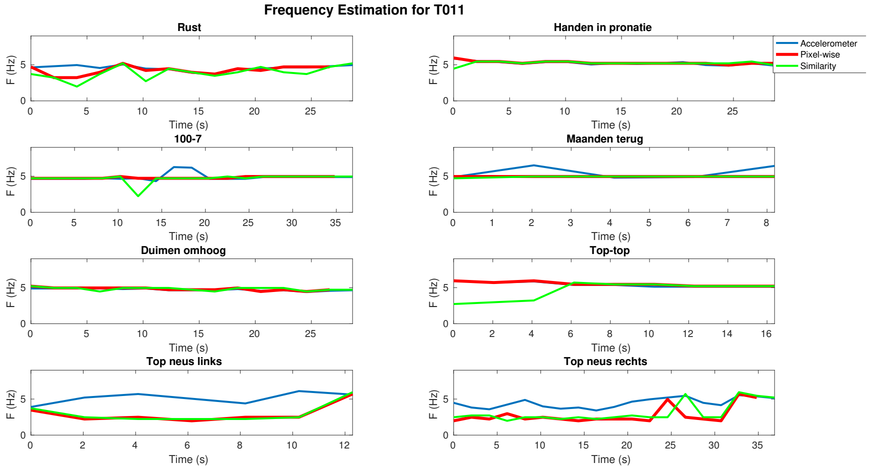
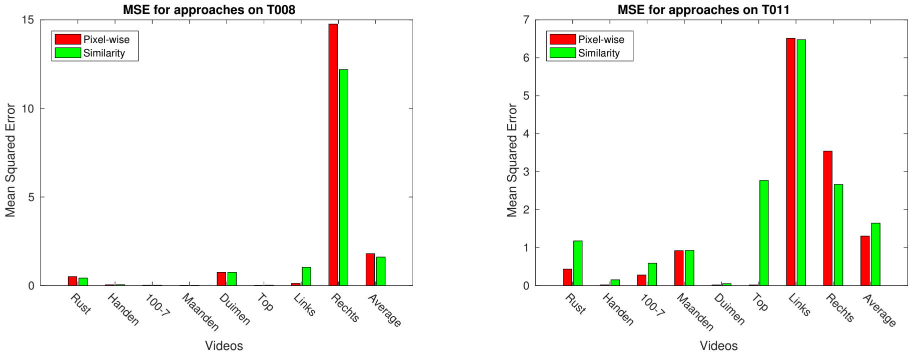


Figure 3.14: Frequency estimations of different approaches on ’T011’. The pixel-wise method has a better performance on 7 videos.



(a) 'T008'

(b) 'T011'

Figure 3.15: MSE of different approaches. The pixel-wise method has a lower error on most videos, except for a large-motion video.

We can see that both methods have an impressive performance on video 'T008'. Our method has a more accurate estimation on video 'Top neus links', while the similarity method does poorly. However, both methods fail on the moving action 'Top neus rechts' and the similarity method has a slight advantage. The test on video 'T011' shows the advantage of our method except for the moving action. Besides, generally our method has a good stability for all estimations.

3.5 Frequency Estimation on Complete Real Data

3.5.1 Evaluation Methods

In this section, the proposed method is evaluated on all 35 'Rust' videos to show the overall performance. To get a benchmark, we first examine the periodicity of the motion by using acceleration data and transforming it to frequency as discussed in Section 3.4.2. In the first experiment, we detect the periodicity and estimate the tremor frequency along time for each video. In the second experiment, we estimate a frequency for each periodic video and compute the absolute error.

3.5.2 Frequency Estimations Along Time

From the benchmark, 21 out of 35 videos are determined as periodic. The average PSD from the benchmark and our approach are shown in Figure 3.19. The frequency estimations are plotted in Figure 3.18.

After experiments, 8 out of 35 videos are detected as periodic videos by our method, 1 of 8 videos is quantified wrongly. The correct detection rate is 33.33%. For the video whose frequency is detected correctly, the spectrum is close to the spectrum from acceleration data, including video 'T008', 'T013', 'T027' and 'T036'. However, though some periodic video is detected correctly, the spectrum is noisy, for example, video 'T011', 'T015' and 'T037'. Mainly the noise comes from the hardware, a camera in our case. The magnitude of the noise mainly depends on the light and complexity of the image.

For those periodic videos that cannot be detected, two cases may exist. For one case, only part of the video shows a periodic property, which includes 6 videos, 'T006', 'T014', 'T016' 'T018', 'T025' and 'T034'. The periodic signal is short and intermittent and the determination of periodicity is based on the general PSD, which leads a negative result. For this case, part of the frequency estimations from the video is still possible to be correct, though the entire video is not regarded as periodic.

For another case, there's no obvious tremor shown in the video, or there is no dominant peak presented in all spectrum along time. This includes 7 videos, 'T002', 'T003', 'T005', 'T009', 'T010', 'T035' and 'T041'. One possibility is that only a small part of the signal in the cropped box shows periodicity. This can be caused by multiple factors, for example, self-occlusion or only finger trembles. Our method looks at the weighted spectrum over the whole cropped box. If only a small part of the power spectrum shows a power peak, it is highly possible that the potential peak is covered by the noise power. Another reason is that the video frame is a 2-dimensional signal, the direction of the tremor could be vertical to the video plane, so that no obvious tremors show in the video. Or the tremor is too weak to be captured by a video camera.

The spectrum of a non-periodic video is typically in trapezoid shape like the spectrum of video 'T021' which is different from that of the non-periodic acceleration signal. Low-frequency components are dominant since the object in a frame is continuous. Trapezoid shape results from a band-pass filter that eliminates the low- and high-frequency components(smaller than 2Hz or higher than 14.5Hz).

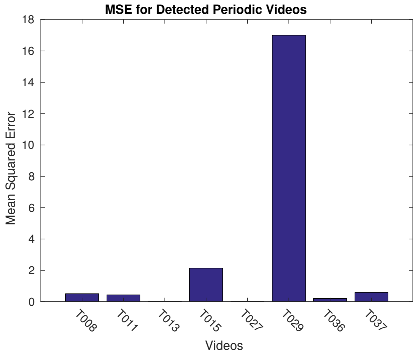


Figure 3.16: MSE for all detected periodic videos. Most of the estimations are accurate, only the estimations for 'T029' is far from correct.

The MSE of the estimations for the detected periodic videos is shown in Figure 3.16. From the results, we can see that if a periodicity is detected, our estimations are close to those from accelerations. There's a special case that the periodicity is detected, but the estimated frequency is far from a correct value because the video frame in 'T029' flickers at 10 Hz, which is wrongly taken as the frequency estimation.

3.5.3 Frequency Estimations for Videos

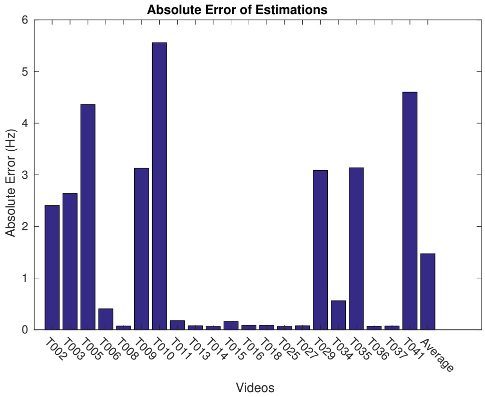
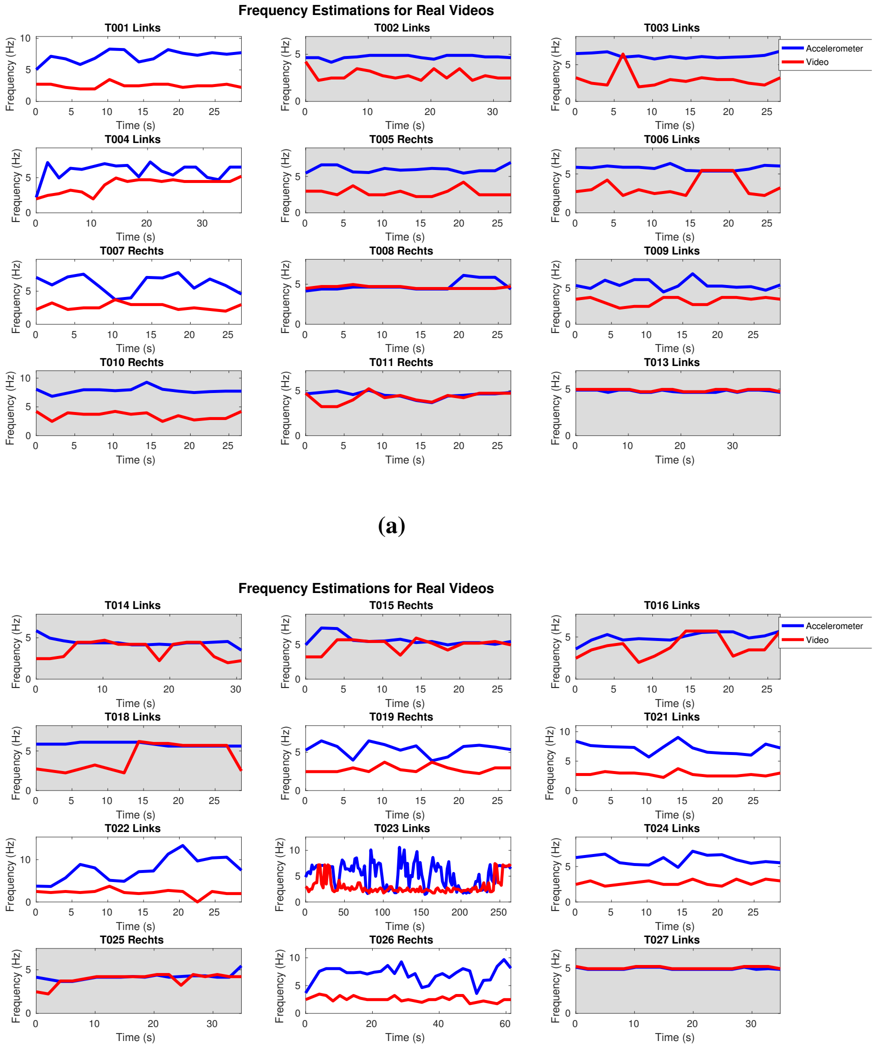


Figure 3.17: Absolute error of estimations for all periodic videos. We show absolute error for our estimations on each periodic video. 13 out of 21 videos have an error lower than 1 Hz.

We design another experiment against the situation that the tremor is short and intermittent. Instead of estimating the tremor frequency along time, we determine an overall value for each video by taking the frequency with the maximum score. The score is calculated by Equation 3.2 for each window of the signal, where S_{f_i} is the score for the frequency f_i, $P(f_i)$ is the power of the frequency, μ_P is the mean of the spectrum P and σ_P is the standard deviation. We simply accumulate the scores for each unique frequency along time. In this case, only the spectrum containing a strong periodic signal will be useful for an estimation.

$$S_{f_i} = P(f_i) - \mu_P - K\sigma_P \tag{3.2}$$

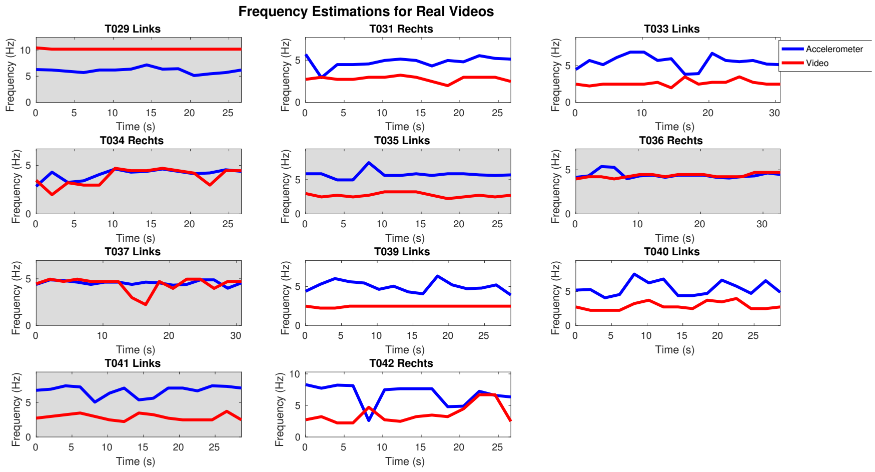
The absolute error of the estimations is shown as Figure 3.17. For 13 periodic videos, the error of the estimation is lower than 1, which shows a competitive performance.



(a)

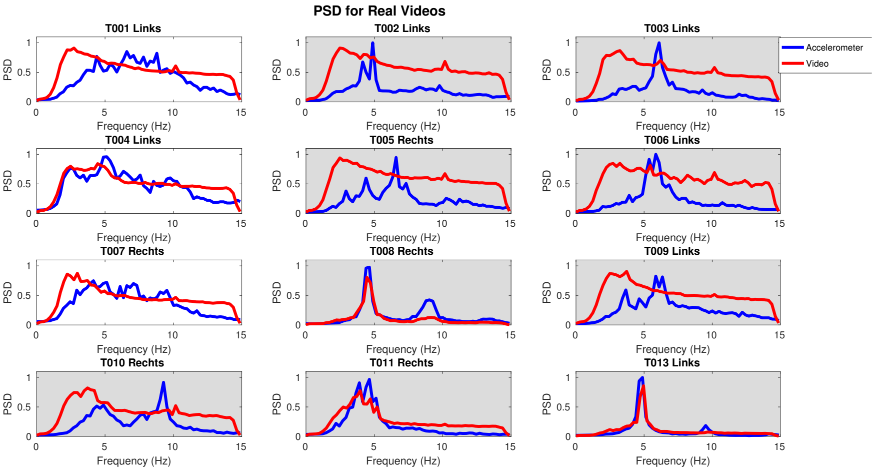
(b)

Figure 3.18: Frequency estimations for real videos.



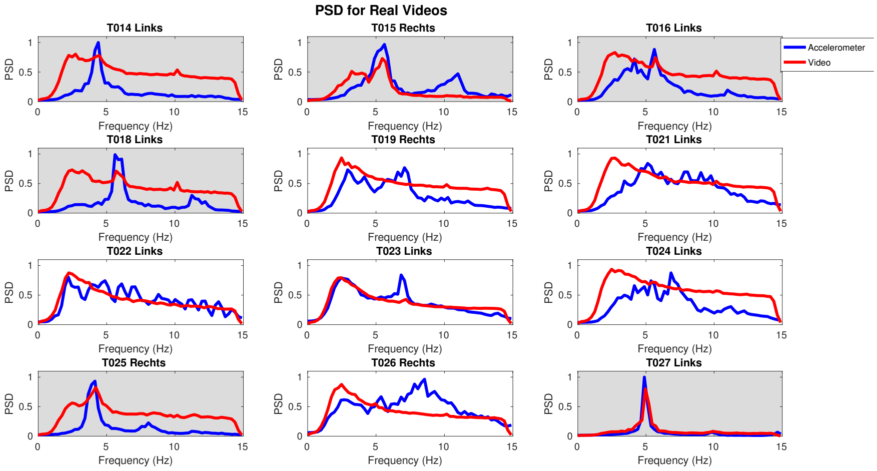
(c)

Figure 3.18: Frequency estimations for real videos. 21 periodic videos detected from the acceleration data are labeled with a gray background. Among them, 7 videos are correctly detected and quantified by our approach. T029 is detected but wrongly quantified. 6 periodic videos are not detected but hit by part of the estimations. The rest 7 videos are not able to be detected.

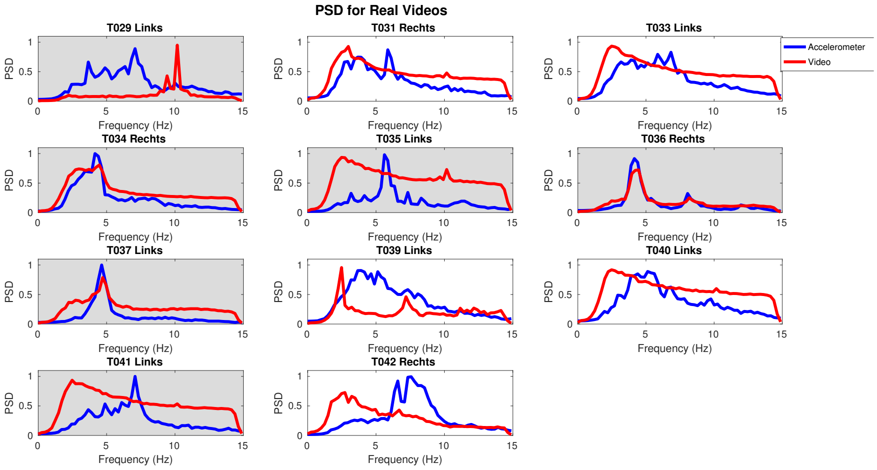


(a)

Figure 3.19: PSD for all videos.



(b)



(c)

Figure 3.19: PSD for all videos. The periodic videos detected from the acceleration data are labeled with a gray background.

Chapter 4

Conclusions and Future Work

4.1 Conclusions

We design a new architecture to detect and quantify the frequency of pathological tremors in a video, which jointly uses pose estimation and periodicity detection techniques. We use a state-of-the-art convolutional neural network, Convolutional Pose Machine, to predict the pose in the frame. Based on the prediction, we propose to crop the joint boxes by using two different coordinate spaces. We finally prove that the approach with the Eulerian coordinate generally has a better performance than the approach using Lagrangian coordinate. A further reason to prefer the Eulerian approach is that the Lagrangian method is computationally more expensive. We propose a new frequency detection method that integrates spectral information using belief maps generated by the pose estimation module. We prove that our frequency estimation method has a better performance on both synthetic videos and real videos, compared with a classic similarity method.

We finally evaluate our approach on real videos. We estimate the frequency along time for each video and our method shows a limited performance because compared to a synthetic video signal, a real video signal contains more complex frequency components and thus is noisier. And some tremor signal is too tiny and short to be detected. The frequency estimation for individual video presents a promising result that our estimations on 13 out of 21 periodic videos have an error lower than 1 Hz.

4.2 Future Work

We prove that our strategy is able to detect a human pathological tremor from a video under certain circumstances. However, it is far from complete. To further improve the performance of the work, one could apply more techniques on different levels.

From Chapter 3, the power spectrum of a real video is noisy. One of the methods to reduce the noise is using phase information decomposed from an image instead of directly using pixel intensity. An image can be decomposed to phase and radial information and specifically phase shift records the motion contents [27], which is especially useful for our case. Besides, to resolve the deficiency of some invisible tremor in 2D plane, one solution is using multiple cameras capturing from different angles to record more information.

For pose estimation, the neural network is not able to accurately predict a pose with self-occlusion. Since patients did several specific actions in the video, the network can be fine tuned on the customized dataset, so that the network is more familiar with our pose and has a better prediction.

Our frequency detection module cannot detect a weak tremor in a video. An additional magnification module is expected to improve the performance, which magnifies implicit motion signal in a video. Several related approaches [23, 10, 27] have been published.

From system level, the structure is designed to be convenient to deploy. Based on this structure, some changes to further improve the overall performance is possible. For example, a frequency heat map is found to be useful to refine the pose estimation from the first module. All in all, there is still some work to do before applying the approach to the real life.

Bibliography

[1] Adriano O Andrade, Adriano Alves Pereira, Maria Fernanda Soares de Almeida, Guilherme Lopes Cavalheiro, Ana Paula Souza Paixao, Sheila Bernardino Fenelon, and Valdeci Carlos Dionisio. Human tremor: origins, detection and quantification. In *Practical Applications in Biomedical Engineering*. InTech, 2013.

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3686–3693, 2014.

[3] Vasileios Belagiannis and Andrew Zisserman. Recurrent human pose estimation. In *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pages 468–475. IEEE, 2017.

[4] A. Beuter, L. Glass, M.C. Mackey, and M.S. Titcombe. *Nonlinear Dynamics in Physiology and Medicine*. Springer New York, 2013.

[5] Adrian Bulat and Georgios Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *European Conference on Computer Vision*, pages 717–732. Springer, 2016.

[6] Angelo Cappello, Alberto Leardini, Maria Grazia Benedetti, Rocco Liguori, and Andrea Bertani. Application of stereophotogrammetry to total body three-dimensional analysis of human tremor. *IEEE Transactions on Rehabilitation Engineering*, 5(4):388–393, 1997.

[7] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4733–4742, 2016.

[8] Ross Cutler and Larry S. Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):781–796, 2000.

[9] Matthias Dantone, Juergen Gall, Christian Leistner, and Luc Van Gool. Human pose estimation using body parts dependent joint regressors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3041–3048, 2013.

[10] Mohamed Elgharib, Mohamed Hefeeda, Fredo Durand, and William T Freeman. Video magnification in presence of large motions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4119–4127, 2015.

[11] Pedro F Felzenszwalb and Daniel P Huttenlocher. Pictorial structures for object recognition. *International journal of computer vision*, 61(1):55–79, 2005.

[12] Roman Goldenberg, Ron Kimmel, Ehud Rivlin, and Michael Rudzsky. Behavior classification by eigendecomposition of periodic motions. *Pattern Recognition*, 38(7):1033–1043, 2005.

[13] Fredric J Harris. On the use of windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE*, 66(1):51–83, 1978.

[14] Matthew J Johnson. Detection of parkinson disease rest tremor. 2014.

[15] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Proceedings of the British Machine Vision Conference*, 2010. doi:10.5244/C.24.12.

[16] Ita Lifshitz, Ethan Fetaya, and Shimon Ullman. Human pose estimation using deep consensus voting. In *European Conference on Computer Vision*, pages 246–260. Springer, 2016.

[17] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016.

[18] Alan V Oppenheim. *Discrete-time signal processing*. Pearson Education India, 1999.

[19] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4929–4937, 2016.

[20] Varun Ramakrishna, Daniel Munoz, Martial Hebert, Andrew J. Bagnell, and Yaser Sheikh. Pose machines: Articulated pose estimation via inference machines. In *ECCV*, 2014.

[21] Leonid Sigal and Michael J Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2041–2048. IEEE, 2006.

[22] Robert Neal Stiles and JE Randall. Mechanical factors in human tremor frequency. *Journal of applied physiology*, 23(3):324–330, 1967.

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

[24] Yang Wang and Greg Mori. Multiple tree models for occlusion and spatial constraints in human pose estimation. In *European Conference on Computer Vision*, pages 710–724. Springer, 2008.

[25] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. *arXiv preprint arXiv:1602.00134*, 2016.

[26] Jing Yang, Hong Zhang, and Guohua Peng. Time-domain period detection in short-duration videos. *Signal, Image and Video Processing*, 10(4):695–702, 2016.

[27] Yichao Zhang, Silvia L Pintea, and Jan C van Gemert. Video acceleration magnification. *arXiv preprint arXiv:1704.04186*, 2017.

Appendix A

Evaluation results

Methods	**Rust**	**Handen**	**100-7**	**Maanden**	**Duimen**	**Top**	**Links**	**Rechts**	**Average**
Eulerian	0.1602	0.1644	0	0	0.2444	0.1187	0.1753	0.3120	0.1305
Lagrangian	0.1602	0.4368	0	0	0.2270	0.1535	1.4876	0.1240	0.2877

(a) STD

Methods	**Rust**	**Handen**	**100-7**	**Maanden**	**Duimen**	**Top**	**Links**	**Rechts**	**Average**
Eulerian	0.5087	0.0422	0.0223	0.0070	0.7504	0.0128	0.1232	14.7594	1.8029
Lagrangian	0.5087	0.2218	0.0223	0.0070	0.7570	0.0192	2.2691	13.7484	1.9504

(b) MSE

Table A.1: Video ’T008’-Cropping Methods Comparison

Methods	**Rust**	**Handen**	**100-7**	**Maanden**	**Duimen**	**Top**	**Links**	**Rechts**	**Average**
Eulerian	0.5823	0.2257	0.1244	0	0.2111	0.2980	1.2883	1.1913	0.4357
Lagrangian	1.7697	1.6469	3.0216	0.7762	0.1874	0.6294	1.6842	1.4357	1.2390

(a) STD

Methods	**Rust**	**Handen**	**100-7**	**Maanden**	**Duimen**	**Top**	**Links**	**Rechts**	**Average**
Eulerian	0.4338	0.0180	0.2787	0.9212	0.0168	0.0165	3.5416	6.5127	1.3044
Lagrangian	3.0022	2.5283	12.0235	2.2400	0.0167	0.5650	2.4152	1.7864	2.7308

(b) MSE

Table A.2: Video ’T011’-Cropping Methods Comparison

Methods	**Rust**	**Handen**	**100-7**	**Maanden**	**Duimen**	**Top**	**Links**	**Rechts**	**Average**
Pixel-wise	0.1602	0.1644	0	0	0.2444	0.1187	0.1753	0.3120	0.1305
Similarity	0.2024	0.2191	0	0	0.2231	0.1533	0.9698	0.5498	0.2575

(a) STD

Methods	**Rust**	**Handen**	**100-7**	**Maanden**	**Duimen**	**Top**	**Links**	**Rechts**	**Average**
Pixel-wise	0.5087	0.0422	0.0223	0.0070	0.7504	0.0128	0.1232	14.7594	1.8029
Similarity	0.4253	0.0504	0.0223	0.0070	0.7489	0.0244	1.0351	12.1972	1.6123

(b) MSE

Table A.3: Video ’T008’-Different Approach Comparison

Methods	**Rust**	**Handen**	**100-7**	**Maanden**	**Duimen**	**Top**	**Links**	**Rechts**	**Average**
Pixel-wise	0.5823	0.2257	0.1244	0	0.2111	0.2980	1.2883	1.1913	0.4357
Similarity	0.8801	0.2561	0.6067	0.1109	0.2270	1.2153	1.3857	1.3231	0.6672

(a) STD

Methods	**Rust**	**Handen**	**100-7**	**Maanden**	**Duimen**	**Top**	**Links**	**Rechts**	**Average**
Pixel-wise	0.4338	0.0180	0.2787	0.9212	0.0168	0.0165	6.5127	3.5416	1.3044
Similarity	1.1770	0.1515	0.5903	0.9260	0.0522	2.7695	6.4765	2.6648	1.6453

(b) MSE

Table A.4: Video ’T011’-Different Approach Comparison

T008	**T011**	**T013**	**T015**	**T027**	**T029**	**T036**	**T037**	**Average**
0.5087	0.4338	0.0187	2.1410	0.0144	17.0059	0.2042	0.5782	2.3228

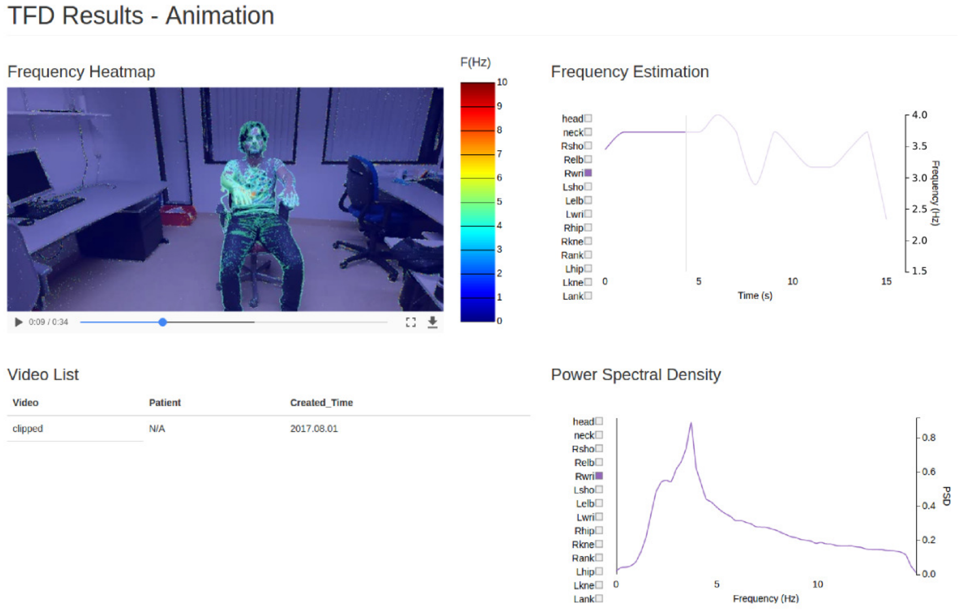
Table A.5: MSE for all detected periodic videos

T002	**T003**	**T005**	**T006**	**T008**	**T009**	**T010**	**T011**	**T013**	**T014**	**T015**
2.4035	2.6362	4.3604	0.4048	0.0721	3.1283	5.5583	0.1759	0.0759	0.0645	0.1607
T016	**T018**	**T025**	**T027**	**T029**	**T034**	**T035**	**T036**	**T037**	**T041**	**Average**
0.0872	0.0872	0.0645	0.0759	3.0852	0.5604	3.1359	0.0683	0.0721	4.6007	1.4704

Table A.6: Absolute error of estimations for all periodic videos

Appendix B

Visualization Module



TFD Results - Animation

Frequency Heatmap

Frequency Estimation

Video List

Video	Patient	Created_Time
clipped	N/A	2017.08.01

Power Spectral Density

Figure B.1: Visualization Module We show the visualization of the final results, mainly including 4 parts. Animations with two charts are possible.