Performance Evaluation of Visual Object Detection and Tracking Algorithms Used in Remote Photoplethysmography

Changchen Zhao[1], Peiyi Mei[1], Shoushuai Xu[2], Yongqiang Li[1], Yuanjing Feng*[1]
[1]Zhejiang University of Technology, Hangzhou 310023, China
[2]Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
fyjing@zjut.edu.cn

Abstract

While most existing remote photoplethysmography (rPPG) approaches employ off-the-shelf visual object detection and tracking algorithms, these algorithms may not be well suited for rPPG problem. The detection and tracking algorithms are designed to be robust to fast deformations, non-distinctive color, fast translations, etc. while rPPG cares about background intervention, region consistency, smoothness of the traces, etc. Hence, there is a gap between a good detection and tracking algorithm and the rPPG measurement accuracy. This paper aims at bridging this gap by evaluating the performance of popular detection and tracking algorithms widely used in rPPG methods. We establish a processing pipeline and choose four detection and tracking algorithms. Experiments are conducted on two publicly available datasets and one self-collected dataset. We find three key factors that affect the rPPG accuracy: 1) stability of the tracking trajectory, 2) content consistency, and 3) robustness to deformation and fast translation. This study highlights the need for developing novel detection and tracking algorithms dedicated to rPPG and gives some useful insights.

1. Introduction

Remote photoplethysmography (rPPG) has recently attracted much more attention and has already become a cutting-edge technique which measures one's physiological parameters including heart rate, respiratory rate [23], blood oxygen saturation [7], blood pressure [15], etc, without the need of any physical contact with the subject [12]. The consumer-level digital cameras are sensitive enough to detect the subtle variations of the light reflected from the skin, in which part of the incident light is absorbed by the blood in the micro-vessels beneath the skin. The color variations of the skin region carry information associated with the heart activities, with the oscillation frequency being equal to the heart rate. The significance of rPPG lies in the non-contact feature in comparison with those conventional photoplethysmography (PPG) measurements, which eliminates the inconvenience during measurement [36]. This feature enables rPPG applicable to both clinical and non-clinical scenarios, such as incubator monitoring [18], home health monitoring [23], fitness training [29], driver monitoring [36], etc.

Visual object detection and tracking has long been a classic computer vision task and undergone intensive investigation. Some well-known challenges like PASCAL Visual Object Classes (VOC) challenge [10], ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [9], and Visual Object Tracking (VOT) challenge [24] are attracting numerous participants every year. However, the main focus of these detection and tracking algorithms may not be of interest of rPPG. For example, the performance of detection and tracking algorithms is evaluated by mean average precision (mAP) and intersection over union (IoU) and does not take into account the continuity between successive frames, which is particularly important for rPPG. Current detection and tracking algorithms aim to improve the accuracy within one single frame while rPPG is determined to maintain consistency between frames, or the smoothness of the tracking trajectory. It is important for rPPG that the tracked area between frames keep as same as possible, no new area involved and no old area removed. Otherwise, the resulting rPPG measurement after spatial averaging (also called raw trace) will introduce noise. Unfortunately, the detection and tracking algorithms are not designed to solve these problems. Therefore, it is necessary to investigate the impact of visual object detection and tracking algorithms on the rPPG measurement and the final accuracy.

This paper focuses on the visual detection and tracking algorithms in rPPG. Most of existing rPPG processing pipeline consists of three components: 1) image preprocessing, 2) region of interest (ROI) detection and tracking, 3) pulse extraction and heart rate calculation. Image preprocessing refers to the preparation of visual data. ROI detec-

tion and tracking determine the location of skin regions in every frame. Pulse extraction converts the visual data to the time series that represents the pulse signal. Researchers have paid a large amount of attention to the first and third components so far. For example, in the image preprocessing phase, the impact of video compression [26, 38, 37] and image resolution [25] on the estimated pulse has been studied. In the pulse extraction and heart rate calculation phase, various models have proposed for pulse extraction, *e.g.*, blind source separation [27], skin reflection model [34], camera acquisition model [19], etc. However, to the best of our knowledge, there is little research regarding the ROI detection and tracking in the field of rPPG.

As a key component in the rPPG processing pipeline, ROI detection and tracking plays a role as important as the other two steps. First, the tracking robustness refers to whether the ROI can be reliably tracked if the target exhibits translation, rotation, occlusion, etc, which ensures the raw trace has minimum background noise. Second, the tracking smoothness refers to whether the same area is tracked during the tracking process, which ensures the raw trace has a minimum disturbance. The tracking robustness and smoothness have equal significance for rPPG because both of them have direct impact on the trace waveform and thus the pulse extraction model design.

In this paper, we systematically investigate the impact of detection and tracking algorithms on the trace waveform and the final measurement accuracy and show that a good visual object detection and tracking algorithm for rPPG is more than robustness. We first perform an extensive literature review, summarizing the commonly used face detection and tracking algorithms. Then, we establish a typical rPPG processing pipeline and choose four representative detection and tracking algorithms in the pipeline. The performance is evaluated by signal-to-noise-ratio (SNR), mean of absolute error (MAE), and root mean squared error (RMSE). Three datasets are used, two publicly available datasets and one self-collected dataset. In addition to the robustness of the tracking algorithm, we are particularly interested in the smoothness. We gain some deep insights from the experiments, which can be a good tip for designing a dedicated ROI detection and tracking algorithm for rPPG. We believe that this is the first attempt to revealing the problems of using off-the-shelf visual object detection and tracking algorithms in rPPG and the start of designing dedicated detection and tracking algorithms.

2. Detection and tracking algorithms in rPPG

Object detection refers to localize the position and size of a target in a given image while object tracking refers to localize the position and size of a target in a number of video frames. In rPPG, the face/skin area is often determined by a detection algorithm and tracked by a tracking algorithm. However, nowadays, the difference between object detection and tracking has become more and more vague. If one applies object detection to every frame, it can also be called object tracking. Moreover, the models of detection and tracking are merging at a deeper level, *e.g.*, detection model trained online during tracking [13]. Hence, in this paper, we discuss both of them.

Method	citation	Method	citation
Viola-Jones[33]	52	KCF[14]	2
Facial landmark	41	SSD[21]	1
KLT[28]	27	OpenFace[1]	1
Skin detection	14	EBGM[16]	1
CSK[13]	2	NPD[20]	1

Table 1. Detection and tracking methods used in rPPG literature and the number of papers used that method.

We performed an extensive search on the internet for the rPPG related papers from 1995 to 2019 and obtained 312 papers. We know that we cannot exhaust all related literature but we tried our best to search for the papers as comprehensive as possible. We recorded the ROI detection and tracking algorithms used in these papers. Table 1 reports several popular algorithms and the number of papers that used this algorithm.

The most frequently used detection algorithm is Viola-Jones face detector [33], which is a cascade of boosted classifier with 14 Haar-like digital image features. The detector is robust, easy to use, and has already been integrated in the OpenCV library. It can be applied to every frame of a video or the first frame and followed by a tracker. Facial landmark is the second frequently used method. Facial landmark refers to a set of algorithms [17, 2, 35] that detect various landmark points on the face. It is often used to select specific facial regions that contain strong rPPG signals such as nose, cheeks, mouth, etc. Similarly, facial landmark algorithms can either be applied to every frame or the first frame followed by a tracking algorithm.

Kanade-Lucas-Tomasi (KLT) is the most frequently used tracking algorithm. The tracker is based on the early work of Lucas and Kanade [22], then developed fully by Tomasi and Kanade [32], and explained clearly in the paper by Shi and Tomasi [28]. The tracker is simple, easy to implement, and is usually used in conjunction with a face detector.

Another main group is skin detection, which localizes the skin pixels. Unlike the face detection and tracking algorithms that output a bounding box, the output of skin detection is irregular shaped. Skin detection can be done by numerous approaches, including skin color thresholding [25], superpixel segmentation [5], convolutional neural networks [31], etc. An accurate skin detection algorithm can effectively eliminate background noise. However, it is usually

time-consuming.

The last several approaches in Table 1 include face detection algorithms (*e.g.*, single shot multi-box detector (SSD) [21], OpenFace library [1], elastic bunch graph matching (EBGM) [16], normalized pixel difference (NPD) [20]) and tracking algorithms (*e.g.*, circulant structure of tracking-by-detection with kernels (CSK) [13] and kernel correlation filter (KCF) [14]). They are recently developed face detection algorithms more advanced than Viola-Jones and KLT.

3. Method

This section describes the processing pipeline used to evaluate the performance of the detection and tracking algorithms, which follows the typical structure summarized in the Introduction section, *i.e.*, 1) face detection and tracking, 2) pulse extraction, and 3) heart rate calculation.

3.1. Face detection and tracking

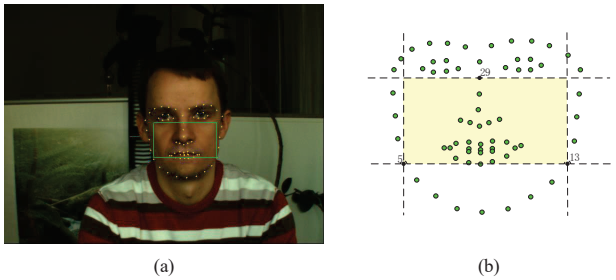
We examine four face detection and tracking algorithms in this paper: one face detection algorithm and three tracking algorithms. In order to ensure a fair comparison, the first ROIs for all the algorithms are the same, *i.e.*, the first ROI of dlib algorithm.

3.1.1 Dlib

Dlib is a modern C++ toolkit containing machine learning algorithms and tools and can be used in a wide range of domains including robotics, embedded devices, mobile phones, and large high performance computing environments [17]. We use this library to detect facial landmarks. Given an input image, dlib detects 68 facial landmarks (Figure 1(a)), each represents a different position on the face. We choose a rectangular ROI for pulse extraction, with its upper-left point based on landmark No. 5 and 29 and bottom-down point based on landmark No. 13 (Figure 1(b)). We use dlib to detect ROI on every frame of a video sequence. The processing speed is about 23 frames per second (fps) (Windows system, python environment, Core i7-8700 CPU, 32GB RAM, 640×480 pixels).

3.1.2 KLT

Kanade-Lucas-Tomasi (KLT) [28] is a traditional object tracking algorithm. Given an ROI in the current frame, KLT detects feature points within that area and then tracks these points in the next frame. A transformation matrix is estimated using the matched feature points between two frames. This transformation is applied to the bounding box to realize the translation, rotation, and scale. We use the Matlab implementation and the processing speed is about 73 fps.



(a) (b)

Figure 1. ROI detection based on 68 facial landmarks. (a) Sample image from PURE dataset [30]. Green rectangle denotes the ROI and yellow dots denote landmark points. (b) ROI (yellow rectangle) determined based on landmark points No. 5, 29, and 13.

3.1.3 CSK

Circulant structure of tracking-by-detection with kernels (CSK) [13] is another object tracking algorithm, which follows the tracking-by-detection paradigm [3, 11]. The core idea of tracking-by-detection is to train a classifier online with samples collected during tracking. In contrast to existing trackers that sample the subwindows sparsely, CSK employs a dense sampling strategy that trains a classifier with all subwindows of an image. By exploiting a circulant structure of the subwindows, CSK derives fast and exact solutions to the optimization problem, resulting in a more efficient training. We use the code provided by the authors with default parameters. The algorithm runs in our project under Matlab environment at 92 fps.

3.1.4 Staple

Sum of Template And Pixel-wise LEarners (Staple) [4], proposed by Bertinetto *et al.*, is built based on correlation filter and tracking-by-detection methods. The tracker combines template and color-based image representations to learn a model that is inherently robust to both color changes and deformations. Specifically, as shown in Figure 2, Staple first computes the histogram of gradients (HOG) features and color histogram feature of the given ROI, which are then convolved with HOG template and color histogram template, respectively, resulting in two response maps. The final response map is obtained by merging these two response maps. The position with maximum response in the final response map is assigned to be the position of the new ROI. Staple runs in our in our project under Matlab environment at 36 fps.

3.2. Pulse extraction

The pulse extraction step converts visual data to the time series that represents the extracted rPPG signal associated with human heart beat. Visual data are first converted to time series data called traces by spatial averaging. Given the ROI of the t-th frame, spatial averaging computes the mean

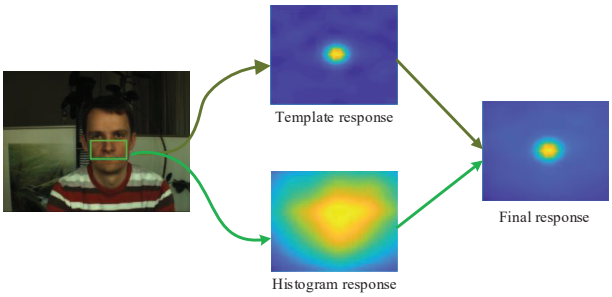


Figure 2. Staple tracker combines template response and color histogram response to estimate the final ROI.

value of all the pixels in each color channel respectively, resulting in a column vector $\boldsymbol{c}_t \in \mathbb{R}^3$ (suppose the image has three color channels). By concatenating all the vectors of the frames, we obtained the traces denoted by a matrix $\boldsymbol{C} \in \mathbb{R}^{3\times L}$, where L denotes the window length.

The traces are assumed to be a combination of the pulse signal, motion artifacts, and noise. The purpose of pulse extract algorithms is to extract the pulse signal from the traces. Numerous approaches have been proposed. We employ POS [34] method. First, the traces need to be temporally normalized by dividing traces by their temporal mean in respective color channels.

$$\boldsymbol{C}_n = \boldsymbol{N} \cdot \boldsymbol{C} \tag{1}$$

where $\boldsymbol{N} \in \mathbb{R}^{3\times 3}$ is a diagonal matrix with i-th diagonal being the reciprocal of mean value of the i-th row of $\boldsymbol{C}$, *i.e.*,

$$\boldsymbol{N}_{ii} = 1/\mu(\boldsymbol{C}_i) \tag{2}$$

A projection plane orthogonal to the skin-tone is used to project the temporally normalized traces onto that plane such that the skin-tone component in the skin-reflection model is eliminated. The algorithm defines two orthogonal directions on the plane orthogonal to the skin-tone, denoted by a projection matrix $\boldsymbol{P}$:

$$\boldsymbol{P} = \begin{pmatrix} 0 & 1 & -1 \\ -2 & 1 & 1 \end{pmatrix} \tag{3}$$

The normalized traces are projected on these two directions by multiplying $\boldsymbol{P}$ with $\boldsymbol{C}_n$:

$$\boldsymbol{S}_1 = \boldsymbol{C}_{nG} - \boldsymbol{C}_{nB} \tag{4}$$

$$\boldsymbol{S}_2 = \boldsymbol{C}_{nG} + \boldsymbol{C}_{nB} - 2\boldsymbol{C}_{nR} \tag{5}$$

The output is calculated by α-tuning:

$$\boldsymbol{p} = \boldsymbol{S}_1 + \alpha \boldsymbol{S}_2 \tag{6}$$

where $\alpha = \sigma(\boldsymbol{S}_1)/\sigma(\boldsymbol{S}_2)$, and $\sigma(\cdot)$ denotes the standard deviation.

Finally, the extracted pulse signal is constructed by concatenating the overlapping windowed signal together with stride equal to half of the window length.

3.3. Heart rate calculation

Heart rate is calculated based on the estimated pulse signal. Heart rate is updated every 1 second with 10s window length. The window starts 10 seconds prior to the current time instant. Fast Fourier Transform is applied to convert the frequency spectrum of the windowed pulse signal. The frequency with maximum power is assigned to be the heart rate, multiplied by 60 to convert the unit from Hertz (Hz) to beats per minute (bpm).

4. Experimental setups

4.1. Datasets

Three datasets are used in the experiment, two publicly available datasets (PURE and UBFC-RPPG) and one self-collected dataset. The self-collected dataset has been used in our previous researches and we call it Self-RPPG in this paper.

PURE [30]: This dataset consists of 10 persons (8 male, 2 female) performing 6 different, controlled head motions in front of a camera, resulting in a total number of 60 sequences of 1 minute each. During these scenarios, the image sequences of the head, as well as reference pulse measurements, were recorded. The videos were captured with a digital camera at a frame rate of 30 Hz with a cropped resolution of 640×480 pixels and a 4.8 mm lens. Reference data have been captured in parallel using a finger clip pulse oximeter (CMS50E) that delivers pulse rate wave and SpO_2 readings with a sampling rate of 60 Hz. The six different setups were as follows:

- **Steady**. The subject was sitting still and looks directly into the camera avoiding head motion.
- **Talking**. The subjects were asked to talk while avoiding additional head motion.
- **Slow Translation**. These sequences comprise head movements parallel to the camera plane. The average speed was 7% of the face height per second, where the average face height was 100 pixels.
- **Fast Translation**. This setup has the same setup as slow translation, except twice the speed of the moving target.
- **Small Rotation**. This setup comprises different targets that were placed at 35 cm around the camera. The subjects were told to look at these targets in a predefined sequence. They were asked to move not only their eyes but orient their head. The head rotation angles are about 20°.
- **Medium Rotation**. These sequences had the same setup as for small rotation, but with the average head angle of 35°.

UBFC-RPPG [6]: The UBFC-RPPG dataset was created using a custom C++ application for video acquisition with a simple low cost webcam (Logitech C920 HD Pro) at 30fps with a resolution of 640x480 in uncompressed 8-bit RGB format. A CMS50E transmissive pulse oximeter was used to obtain the ground truth PPG data comprising of the PPG waveform as well as the PPG heart rates. During the recording, the subject sits in front of the camera (about 1m away from the camera) with his/her face visible. All experiments are conducted indoors with a varying amount of sunlight and indoor illumination. The dataset contains a total number of 42 videos of 1 min each. The subject sits steadily in front of the video with the face almost motionless.

Self-RPPG: The self-collected dataset contains 83 videos, recorded with a regular webcam (Logitech C920), 30 fps, 640 480 pixels, 1-minute duration, and stored in uncompressed AVI format. Ground-truth pulse waveforms are either the contact fingertip PPG signal measured by a pulse oximeter (for the stationary case) or the ECG signal measured by a self-made 2-electrode ECG measurement device (for the motion case). 18 healthy subjects (14 males and 4 females, aged 21 to 35) were recruited. The dataset contains two major categories:

- **Steady**. The subject was asked to look at the camera and keep still while recording. The illumination conditions include: sunlight, fluorescent light, and mixed of both. The subject conditions include: normal condition, after drinking alcohol, after exercise.
- **Motion**. The subjects were performing two different movements including moving horizontally, exercising on a biking machine.

Figure 3 shows some sample images of the three datasets.

Figure 3. Sample images of the datasets. First row: PURE, second row: UBFC-RPPG, last row: Self-RPPG.

4.2. Evaluation metrics

Three commonly used metrics are used to evaluate the performance of the detection and tracking algorithms.

Signal-to-noise-ratio (SNR) is used to measure the quality of the estimated pulse signal, which is first defined by De Haan *et al*. [8]. SNR is calculated as follows:

$$\text{SNR} = 10\log_{10}\left(\frac{\sum_{f=0.8}^{5} U(f)S^2(f)}{\sum_{f=0.8}^{5}(1-U(f))S^2(f)}\right) \quad (7)$$

where $S(f)$ denotes the power spectrum of the extracted pulse waveform, f denotes the frequency in Hz, and $U(f)$ denotes a template separating signal and noise, which is defined as:

$$\hat{U}(f) = \begin{cases} 1, & f_r - \frac{\omega}{2} \le f \le f_r + \frac{\omega}{2} \\ 1, & 2f_r - \frac{\omega}{2} \le f \le 2f_r + \frac{\omega}{2} \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

where f_r denotes the ground-truth heart rate calculated according to either pulse oximeter or an Electrocardiogram (ECG) measurement device, and ω denotes the spectral window length.

The definition of the template window in this paper is slightly different from that in [8]:

$$U(f) = \begin{cases} 1, & f_r - \frac{\omega}{2} \le f \le f_r + \frac{\omega}{2} \\ 1, & 2f_r - \frac{\omega}{2} \le f \le 2f_r + \frac{\omega}{2} \\ 1, & 3f_r - \frac{\omega}{2} \le f \le 3f_r + \frac{\omega}{2} \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

In this definition, we include the second harmonic of the dominant frequency as the valid signal. This is based on the observation that the second harmonic is visible in most of the power spectrums of the extracted pulse. It will be more accurate if the second harmonic is involved in the numerator.

The mean absolute error (MAE) and root mean squared error (RMSE) are used to measure the error between the estimated heart rate $HR(t)$ and the ground-truth heart rate $HR_r(t)$:

$$\text{MAE} = \frac{1}{N}\sum_{t=1}^{N}|HR(t) - HR_r(t)| \quad (10)$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(HR(t) - HR_r(t))^2} \quad (11)$$

5. Results and discussion

In order to compare the performance of different detection and tracking algorithms, for each video, we run the four compared face detection and tracking algorithms once at a time with other processing steps being the same, calculate the evaluation metrics, and average the results over all the videos in one category. The results are reported in Tables 2, 3, and 4.

Tables 2 shows the results on PURE dataset, in which the results of every category are reported and the last row

		KLT	Dlib	CSK	Staple
Steady	SNR	13.26	12.82	**13.42**	13.11
	MAE	1.75	1.69	1.68	**1.67**
	RMSE	2.52	2.18	**2.16**	**2.16**
Talking	SNR	3.40	**4.06**	3.40	3.36
	MAE	**8.99**	10.84	10.19	10.15
	RMSE	**13.46**	15.10	14.53	13.77
Slow transl.	SNR	10.58	**11.12**	10.97	9.53
	MAE	1.71	1.63	**1.60**	1.64
	RMSE	2.25	2.13	**2.12**	2.22
Fast transl.	SNR	9.27	**10.54**	9.61	8.27
	MAE	2.31	**1.79**	1.81	1.93
	RMSE	3.19	**2.40**	2.41	2.95
Small rotation	SNR	10.04	**10.46**	9.87	8.34
	MAE	1.71	**1.45**	1.62	1.79
	RMSE	3.09	**1.91**	2.19	2.53
Medium rotation	SNR	8.11	**8.72**	7.39	7.05
	MAE	3.41	2.47	3.35	**2.09**
	RMSE	6.17	4.55	5.14	**2.86**
Overall	SNR	9.21	**9.71**	9.21	8.36
	MAE	3.22	3.18	3.26	**3.09**
	RMSE	4.97	4.54	4.59	**4.26**

Table 2. Accuracy results on PURE dataset. SNR in dB, MAE in bpm, RMSE in bpm.

	KLT	Dlib	CSK	Staple
SNR	**4.50**	3.19	3.91	4.02
MAE	**2.45**	3.15	3.44	3.24
RMSE	**5.25**	6.48	6.80	6.30

Table 3. Accuracy results on UBFC-RPPG dataset. SNR in dB, MAE in bpm, RMSE in bpm.

		KLT	Dlib	CSK	Staple
Steady	SNR	10.06	8.60	**10.56**	9.83
	MAE	0.87	1.01	**0.79**	0.88
	RMSE	2.37	2.68	**2.06**	2.36
Motion	SNR	4.50	3.31	**4.68**	3.30
	MAE	5.12	5.98	**3.87**	5.01
	RMSE	7.71	9.73	**6.38**	8.40

Table 4. Accuracy results on Self-RPPG dataset. SNR in dB, MAE in bpm, RMSE in bpm.

denotes the average results on the entire dataset. In PURE dataset, only the Steady category is the stationary case, all the other categories are the motion case. It can be seen that, for the stationary case, KLT, CSK, and Staple achieve comparative SNR scores, *e.g.*, SNR is greater than 13 dB, with CSK achieving the highest SNR. Dlib has the least SNR score, *i.e.*, 12.82 dB. On the contrary, dlib achieves the highest SNR scores on all the motion categories.

The results in Table 3 and Table 4 demonstrate that KLT achieves the highest scores in SNR, MAE, and RMSE on UBFC-RPPG dataset while CSK achieves the highest scores in both steady and motion categories on Self-RPPG dataset. In both datasets, KLT and CSK perform better than other algorithms while dlib exhibits the worst performance.

The overall performance of each algorithm can be summarized as follows. Dlib exhibits significant fluctuations during tracking in both stationary case and motion case. The position of the landmark points changes dramatically even if the subject is keeping still. This problem leads to a large amount of noise in the rPPG measurements. KLT and CSK perform relatively better than other algorithms. The object can be stably tracked in both stationary and motion cases. But when the subject rotates his/her head, part of the ROI will go out of the facial region. This is the common problem of most bounding box based algorithms. The performance of Staple is moderate, *e.g.*, neither significant fluctuations nor the smooth tracking trajectory. The biggest problems of Staple is that the position and scale of the ROI are updated in a discontinuous way, *e.g.*, the ROI position will not change until the subject has a sufficiently large translation, resulting in a jumping tracking trajectory.

It seems that it is difficult to decide which algorithm performs the best. In fact, every algorithm has its merits and demerits. By considering the relationship between the tracking algorithm performance and the corresponding measurement traces, we summarize the following three factors that affect the rPPG accuracy.

5.1. Stability

While tracking robustness refers to whether the detected object can keep track of the target, tracking stability refers to the smoothness of the tracking trajectory. This is of significant importance to the rPPG problem. Since the trace is obtained by spatial averaging, the stability of tracking trajectory and the ROI size have a direct impact on the trace waveform. We analyze the stability in stationary and motion cases, respectively.

5.1.1 Stability in stationary case

Figure 4 shows an example of the tracking process in the stationary case, where the subject keeps almost still while recording. The first four rows denote the trajectory of the ROI, *i.e.*, x, y coordinates of the upper-left point, width, and height, respectively. The x and y coordinates represent the position of the ROI while the width and height repre-

sent the scale. Ideally, both of the position and the scale of the ROI should keep unchanged during the whole process. However, it can be seen that dlib is jittering all the time, both in position and scale. The direct result is that the trace of dlib has a large amount of noise, which can be seen in comparison with the traces of other trackers. KLT has less fluctuations than dlib. CSK and Staple are the most stable trackers, where the position and scale keep almost the same during the whole process.

5.1.2 Stability in motion case

Figure 5 shows a motion example where the subject is moving his head horizontally. Similar to the stationary case, dlib exhibits the most significant fluctuations in both ROI position and scale. Staple exhibits discontinuity in the tracking trajectories. The ROI does not move until the target has a large motion, which means that staple tracker cannot reflect the target's movement in time. The direct impact is that the trace signal has similar discontinuities, which introduces undesirable artifacts.

5.2. Content consistency

Almost all the detection and tracking algorithms output the bounding box to denote the target's position and size. However, the bounding box cannot describe precisely the actual shape of the target, *i.e.*, background may be involved and this problem may lead to inaccuracy in rPPG measurement. On the contrary, the landmark point based method, dlib, does not have this problem. Because the facial landmark points can adjust their positions according to the face deformation and the ROI selection rule, the resulting ROI covers the facial region all the time during tracking. Therefore, content consistency can be guaranteed.

Figure 6 shows an example, in which the subject takes a rotation of his head. The bounding box methods, *e.g.*, KLT, CSK, and Staple, rotate their ROIs accordingly, and thus, the background is involved. On the contrary, the landmark points method, *e.g.*, dlib, fully covers the facial region all the time. The traces given by different methods can be compared in Figure 6 (b). When the ROI covers the facial region, the trace intensity is under a certain level (denoted by gray region). When the background region is involved, the trace intensity goes up above the normal level because, in this case, the background color is brighter than the facial color. In such a case, the trace involves background noise and thus, degenerates the performance of the rPPG pulse extraction model.

5.3. Robustness

5.3.1 Robustness to deformation

Deformation occurs when the subject is talking or blinking. The tracker may be subjected to the deformation, exhibiting instability. For example, when the subject is talking, the position of landmarks around the mouth change. The ROI changes accordingly. Similarly, for the KLT tracker, the tracked feature points are changing, and thus, the ROI is no longer stable.

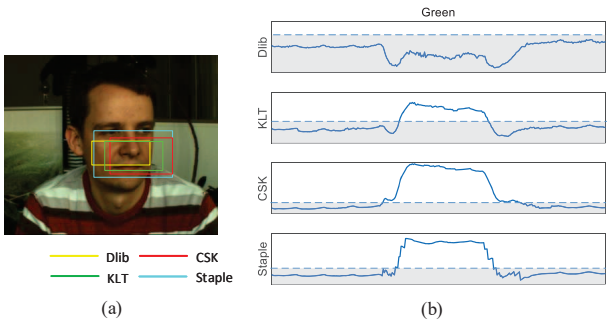


Figure 6. Performance comparison between bounding box based methods and landmark point based methods. (a) Sample image of one subject in PURE dataset with ROIs of four trackers. (b) Traces computed from four trackers. Gray shaded regions denote the normal level when the ROI fully covers the facial region (no background region is involved).

On the contrary, CSK and Staple are less sensitive to deformation, *i.e.*, the ROI keeps stable while the subject is talking. This is partially because of the use of correlation filters, *e.g.*, the global minimum will not be affected by local variations. For example, Staple employs HOG and color histogram models, which are both spatially invariant features.

5.3.2 Robustness to fast translation

Translation is one of the most commonly occurred motion type in many real-life applications, *e.g.*, the subject's head is moving left and right or forward and backward. The tracking robustness to fast translation is one of the most important metrics for the design of a tracker. In rPPG, the loss of target means that the traces contain no longer the rPPG signal, or that the rPPG signal is contaminated by significant background noise, which definitely undermines the heart rate measurement accuracy. Figure 7 shows an example of tracking failure, in which the CSK tracker cannot fully overlap with the face region. The shaded region denotes the loss of target period, in which one can see that the signal value is much higher than previous time period.

6. Conclusions

There exists a gap between the ROI detection and tracking algorithms and the rPPG measurement accuracy, *i.e.*, an accurate detection and tracking algorithm in the conventional definition does not necessarily guarantee an accurate heart rate measurement result. This paper investigates this gap by analyzing four visual object detection and tracking

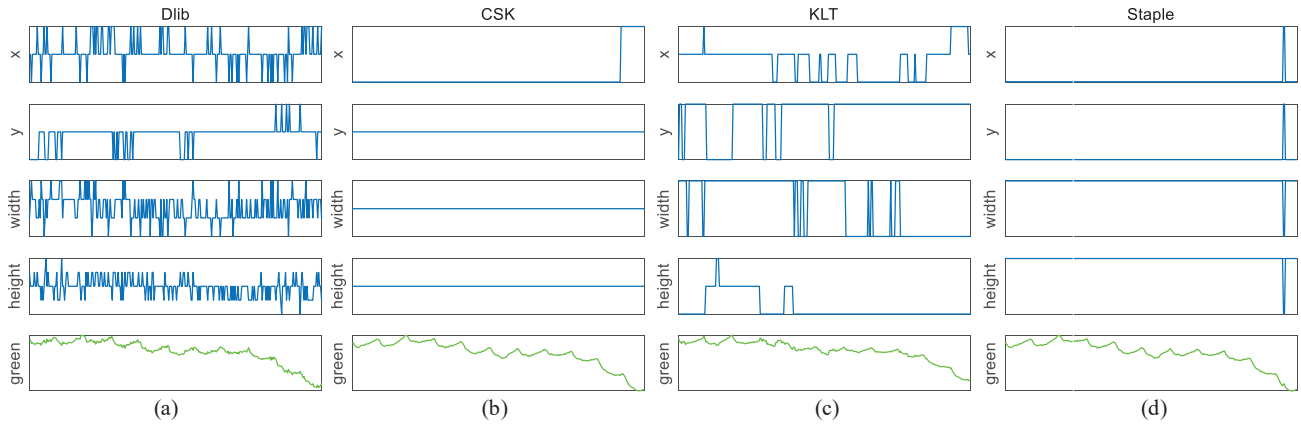


Figure 4. Fluctuations in stationary case. Rows 1 to 4 represent the x, y coordinates of the upper-left point, width, and height of the ROI, respectively. Row 5 represents the trace, in which only the green channel is shown.

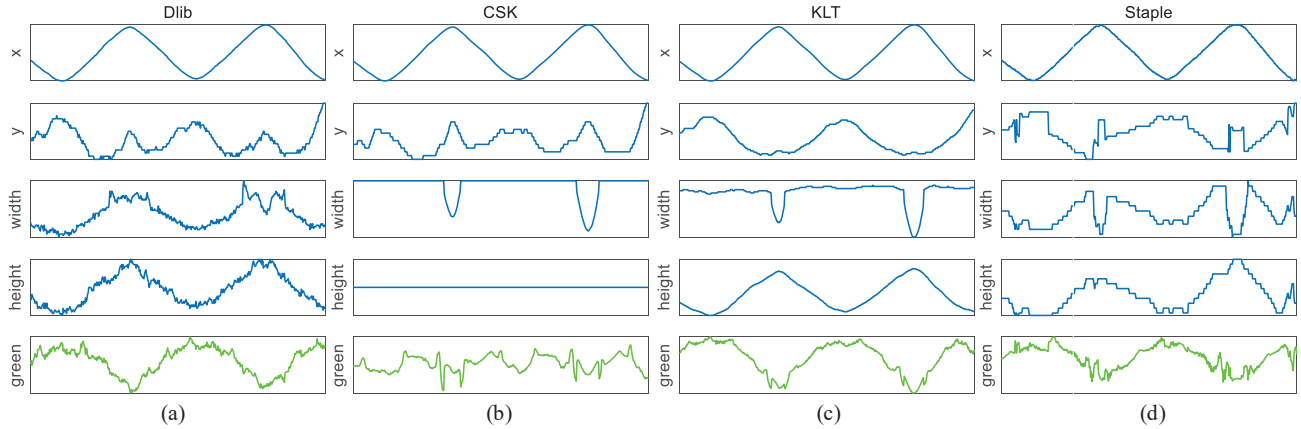


Figure 5. Fluctuations in motion case. Rows 1 to 4 represent the x, y coordinates of the upper-left point, width, and height of the ROI, respectively. Row 5 represents the trace, in which only the green channel is shown.

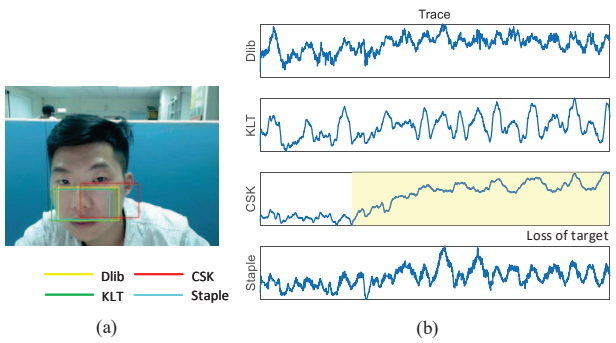


Figure 7. Example of tracking failure. (a) Sample image of one subject in Self-RPPG dataset with ROIs of four trackers. (b) Traces computed from four trackers. Yellow shaded region denotes the time when the tracker losses the target.

algorithms on three datasets in terms of SNR, MAE, and RMSE. We observe three factors that affect the measurement accuracy: 1) stability of the tracking trajectory, 2) content consistency, and 3) robustness to deformation and fast translation. The first two factors are new problems when detection and tracking algorithms are applied to rPPG while the third factor is the classical problem in the field of visual object detection and tracking. This study highlights the need for developing novel detection and tracking algorithms dedicated to rPPG. There is room for accuracy improvement if the three factors are considered.

Acknowledgement

This research was sponsored by National Natural Science Foundation of China under Grant No. 61903336, 61976190, 61703369.

References

[1] Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. Openface: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science, 2016.

[2] Akshay Asthana, Stefanos Zafeiriou, Shiyang Cheng, and Maja Pantic. Robust discriminative response map fitting with

constrained local models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[3] Boris Babenko, Ming Hsuan Yang, and Serge Belongie. Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 33(8):1619–32, 2011.

[4] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip HS Torr. Staple: Complementary learners for real-time tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[5] Serge Bobbia, Duncan Luguern, Yannick Benezeth, Keisuke Nakamura, Randy Gomez, and Julien Dubois. Real-time temporal superpixels for unsupervised remote photoplethysmography. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018.

[6] Serge Bobbia, Richard Macwan, Yannick Benezeth, Alamin Mansouri, and Julien Dubois. Unsupervised skin tissue segmentation for remote photoplethysmography. *Pattern Recognition Letters*, 124:82–90, 2019.

[7] Sitthichok Chaichulee, Mauricio Villarroel, Joao Jorge, Carlos Arteta, Gabrielle Green, Kenny Mccormick, Andrew Zisserman, and Lionel Tarassenko. Multi-task convolutional neural network for patient detection and skin segmentation in continuous non-contact vital sign monitoring. In *Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition*, 2017.

[8] Gerard De Haan and Vincent Jeanne. Robust pulse rate from chrominance-based rppg. *IEEE Transactions on Biomedical Engineering*, 60(10):2878–2886, 2013.

[9] Jia Deng, Wei Dong, Richar Socher, Li-Ji Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[10] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015.

[11] Sam Hare, Amir Saffari, and Philip H. S. Torr. Struck: Structured output tracking with kernels. In *International Conference on Computer Vision*, 2011.

[12] Mohamed Abul Hassan, Aamir Saeed Malik, David Fofi, N M Saad, Babak Karasfi, Yasir Salih Ali, and Fabrice Meriaudeau. Heart rate estimation using facial video: A review. *Biomedical Signal Processing and Control*, 38:346–360, 2017.

[13] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *European Conference on Computer Vision*, pages 702–715. Springer, 2012.

[14] Joao F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 37(3):583–596, 2015.

[15] Monika Jain, Sujay Deb, and A V Subramanyam. Face video based touchless blood pressure and heart rate estimation. *Multimedia Signal Processing*, pages 1–5, 2016.

[16] J. K. Kamarainen, V. Kyrki, and H. Kalviainen. Invariance properties of gabor filter-based features-overview and applications. *IEEE Transactions on Image Processing*, 15(5):1088–1099, 2006.

[17] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[18] John H Klaessens, Marlies Van Den Born, Albert J Van Der Veen, Janine Sikkensvan De Kraats, Frank A Van Den Dungen, and Rudolf M Verdaasdonk. Development of a baby friendly non-contact method for measuring vital signs: first results of clinical measurements in an open incubator at a neonatal intensive care unit. *Proceedings of SPIE 8935, Advanced Biomedical and Clinical Diagnostic Systems XII*, 8935:57–62, 2014.

[19] Xiaobai Li, Chen Jie, 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*, 2014.

[20] Shengcai Liao, Anil K. Jain, and Stan Z. Li. A fast and accurate unconstrained face detector. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 38(2):211–223, 2016.

[21] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In *European Conference on Computer Vision*, 2016.

[22] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of International Joint Conference on Artificial Intelligence*, 1981.

[23] Carlo Massaroni, Daniel Simoes Lopes, Daniela Lo Presti, Emiliano Schena, and Sergio Silvestri. Contactless monitoring of breathing patterns and respiratory rate at the pit of the neck: A single camera approach. *Journal of Sensors*, pages 1–13, 2018.

[24] Kristan Matej, Matas Jiri, Leonardis Aleš, Vojir Tomas, Pflugfelder Roman, Fernandez Gustavo, Nebehay Georg, Porikli Fatih, and Čehovin Luka. A novel performance evaluation methodology for single-target trackers. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 38(11):2137–2155, 2016.

[25] Daniel Mcduff. Deep super resolution for recovering physiological information from videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018.

[26] Daniel J. Mcduff, Ethan B. Blackford, and Justin R. Estepp. The impact of video compression on remote cardiac pulse measurement using imaging photoplethysmography. In *Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition*, 2017.

[27] Poh Ming-Zher, 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.

[28] Jianbo Shi and Carlo Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1994.

[29] Radim Špetlík, Vojtech Franc, and Jirí Matas. Visual heart rate estimation with convolutional neural network. In *Proceedings of British Machine Vision Conference*, 2018.

[30] Ronny Stricker, Steffen Muller, and Horst Michael Gross. Non-contact video-based pulse rate measurement on a mobile service robot. In *Proceedings of the IEEE International Symposium on Robot & Human Interactive Communication*, 2014.

[31] Chuanxiang Tang, Jiwu Lu, and Jie Liu. Non-contact heart rate monitoring by combining convolutional neural network skin detection and remote photoplethysmography via a low-cost camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018.

[32] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. Technical report, Carnegie Mellon University Technical Report CMU-CS-91-132, 1991.

[33] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[34] Wenjin Wang, Brinker Bert Den, Sander Stuijk, and Haan Gerard De. Algorithmic principles of remote-ppg. *IEEE Transactions on Biomedical Engineering*, 64(7):1479–1491, 2017.

[35] Shuzhe Wu, Meina Kan, Zhenliang He, Shiguang Shan, and Xilin Chen. Funnel-structured cascade for multi-view face detection with alignment-awareness. *Neurocomputing*, 221:138–145, 2016.

[36] Qi Zhang, Qingtian Wu, Yimin Zhou, Xinyu Wu, Yongsheng Ou, and Huazhang Zhou. Webcam-based, non-contact, real-time measurement for the physiological parameters of drivers. *Measurement*, 100:311–321, 2017.

[37] Changchen Zhao, Weihai Chen, Chun-Liang Lin, and Xingming Wu. Physiological signal preserving video compression for remote photoplethysmography. *IEEE Sensors Journal*, 19(12):4537–4548, 2019.

[38] Changchen Zhao, Chun-Liang Lin, Weihai Chen, and Zhengguo Li. A novel framework for remote photoplethysmography pulse extraction on compressed videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018.