Evaluating Wearable Sensor Locations for Dual-PPG Pulse Transit Time Measurement

Ronny Abdullah
Stanford University
Stanford, California, USA

Abstract

Pulse transit time (PTT) is frequently studied as a potential surrogate for cuffless blood-pressure sensing, but the practicality of a wearable PTT system depends heavily on signal quality and sensor placement. In this project, a dual-photoplethysmography (PPG) platform was implemented using a Raspberry Pi, an ADS1256 analog-to-digital converter, and two reflective optical pulse sensors. The goal was not to claim clinical blood-pressure accuracy, but to determine which practical wearable sensor-location pairs produced the most stable and repeatable beat-to-beat timing estimates. Four configurations were tested: wrist-to-fingertip, finger-root-to-fingertip, palm-to-fingertip, and ear-to-fingertip. Signals were acquired at an effective sampling rate of approximately 218 Hz and processed using baseline removal, smoothing, peak detection, beat pairing, and outlier rejection. The results showed that wrist-to-fingertip measurements were highly unreliable, while same-hand short-path configurations produced substantially better matched-beat yield and timing stability. Among the tested geometries, finger-root-to-fingertip produced the most reliable overall performance in this single-subject dataset. Ear-to-fingertip measurements produced longer delays, as expected from a longer anatomical path, but were more variable across repeated trials. These results demonstrate that low-cost dual-PPG PTT measurement is feasible, but strongly dependent on body location, sensor stability, and acquisition methodology.

Keywords

photoplethysmography, pulse transit time, wearable sensing, biomedical instrumentation, cuffless blood pressure

1 Introduction

Continuous cardiovascular monitoring remains an important goal for wearable health systems, yet conventional cuff-based blood-pressure measurement is inherently intermittent and inconvenient. Pulse transit time (PTT), the time required for a pulse wave to travel between two body locations, is often explored as a possible pathway toward cuffless blood-pressure estimation [4]. However, a practical wearable PTT system requires more than simply recording two pulse waveforms. It also requires that both sensing locations produce stable optical signals, that beats can be matched reliably, and that the resulting delay estimates remain repeatable.

This project addresses those practical issues by focusing on sensor placement. A dual-PPG acquisition platform was built using low-cost reflective sensors and a Raspberry Pi, then used to compare several wearable body-location pairs. The purpose of the project was not to validate absolute blood-pressure estimation, but rather to evaluate which locations produce the most reliable PTT measurements under realistic low-cost sensing conditions.

The central research question was:

> Which wearable-style dual-PPG sensor locations produce the most stable and repeatable pulse transit time estimates?

2 Motivation/Background

2.1 Photoplethysmography

Photoplethysmography (PPG) is an optical sensing technique that measures changes in blood volume in tissue by illuminating the skin and detecting changes in reflected or transmitted light [1, 2]. Because the pulsatile component of the PPG waveform is linked to the cardiac cycle, PPG has become widely used in wearable devices for heart-rate sensing and related physiological monitoring [3].

Although PPG is attractive for wearable systems due to its low cost and compact form factor, signal quality is highly dependent on local anatomy, tissue perfusion, ambient light, motion, and contact pressure [1, 3]. As a result, a body location that is acceptable for simple heart-rate monitoring may still be a poor choice for precise beat-to-beat timing measurement.

2.2 Pulse Transit Time

In this work, PTT was defined as

$$\mathrm{PTT} = t_{\mathrm{distal}} - t_{\mathrm{proximal}}, \tag{1}$$

where t_{proximal} is the pulse-arrival time at the sensor closer to the heart and t_{distal} is the arrival time at the more distant sensor.

PTT is often investigated because arterial stiffness and pulse-wave velocity are related to blood pressure; in general, a faster pulse wave corresponds to a shorter transit time [4]. At the same time, the measured value depends strongly on the chosen waveform fiducial, the sampling method, and the quality of the recorded signals [4]. Therefore, before using PTT for any blood-pressure-related application, the sensing geometry itself must be evaluated.

2.3 Motivation for a Location Study

Many introductory demonstrations of PTT assume that two sensors can simply be attached at different points on the body and yield a useful timing difference. In practice, this assumption is weak. Some locations produce faint optical signals, some are especially vulnerable to motion, and others distort the waveform enough that consistent beat matching becomes difficult.

The motivation for this project was therefore to move beyond a single proof-of-concept setup and instead compare several practical wearable locations. By treating sensor placement as the key design variable, the project becomes a reliability study of wearable dual-PPG sensing rather than only a demonstration of pulse timing.

3 Design and Implementation

3.1 Hardware

The system consisted of:

- a Raspberry Pi for acquisition and control,
- an ADS1256 24-bit analog-to-digital converter board [5],
- two reflective PPG sensors,
- Python scripts for recording and analysis.

Each sensor output was connected to one ADC input channel. The Raspberry Pi controlled the ADS1256 and saved paired dual-channel time-series data to CSV files for offline analysis.

3.2 Sampling and Acquisition

Although the ADS1256 supports high-rate conversion, the effective data rate in this project was limited by Python loop overhead, channel switching, and software timing. Across trials, the measured effective sampling rate was approximately 218 Hz.

A critical limitation of the acquisition method is that the two channels were not sampled simultaneously. Instead, the software loop read the proximal channel first and then the distal channel. For a system measuring delays on the order of 10–40 ms, this sequential channel access introduces a nonzero inter-channel time skew of approximately 4.6 ms (one sample period at 218 Hz). While small relative to a full heartbeat, this offset is non-negligible compared to the measured PTT values, which ranged from roughly 14 to 44 ms across configurations, and introduces an unresolved systematic component in all reported estimates. Consequently, the reported PTT values should be interpreted as approximate timing estimates for comparing *placement reliability*, not as precise absolute physiological transit times.

Each trial recorded roughly 60 seconds of data. Four sensor-location pairs were evaluated:

(1) wrist → fingertip,
(2) finger root → fingertip,
(3) palm → fingertip,
(4) ear → fingertip.

Two trials were collected for each configuration.

3.3 Signal Processing Pipeline

The software analysis pipeline followed these steps:

(1) load the two recorded PPG time series,
(2) remove the DC component from each channel,
(3) suppress slow baseline drift,
(4) smooth the residual waveform,
(5) detect local peaks in each channel,
(6) pair proximal and distal peaks using a constrained delay window,
(7) compute beat-to-beat PTT values,
(8) reject extreme outliers around the median PTT.

The filtering stage was intentionally simple. A long moving-average window of approximately 0.75 s was used to estimate and subtract slow baseline drift. This choice was longer than the width of an individual pulse waveform and therefore suppressed slow motion and contact changes while preserving beat-scale structure. A short smoothing window of approximately 30 ms was then used to suppress high-frequency noise without strongly distorting the pulse morphology. The pairing window was chosen empirically to retain plausible short-delay matches for the tested geometries while rejecting obviously incorrect pairings. Because this was an exploratory prototype rather than a clinically optimized pipeline, the parameters were selected as practical engineering compromises rather than through formal optimization.

For each trial, the script generated a three-panel figure containing:

- the raw DC-removed dual-PPG waveforms,
- the filtered waveforms with detected peaks,
- the beat-to-beat PTT values and mean line.

3.4 Evaluation Criteria

Because no ECG, PCG, or cuff-based blood-pressure reference was used, the project does not claim absolute physiological or clinical accuracy. Instead, each location pair was evaluated using:

- waveform quality,
- matched-beat yield,
- timing stability,
- repeatability across trials,
- plausibility of the resulting delay estimates.

4 Results

Table 1 summarizes the final results. The numerical values reported here follow the actual analyzed trial filenames and are intended to be the single source of truth for the report.

Table 1: Summary of dual-PPG PTT results across tested location pairs.

Location Pair	Trial	Mean PTT (ms)	Matched Beats	Assessment
Wrist → Finger	1	16.2	4	Poor
Wrist → Finger	2	98.4	2	Unstable
Finger Root → Finger	1	14.4	19	Best
Finger Root → Finger	2	23.2	14	Good
Palm → Finger	1	20.4	11	Moderate
Palm → Finger	2	14.1	12	Moderate
Ear → Finger	1	22.6	19	Moderate
Ear → Finger	2	43.8	12	Variable

The overall pattern was clear. Wrist-to-fingertip measurements were the least reliable. Finger-root-to-fingertip produced the strongest same-hand short-path results. Palm-to-fingertip was usable but not clearly superior to finger-root placement. Ear-to-fingertip produced larger delays, consistent with a longer anatomical path, but showed higher inter-trial variation.

4.1 Wrist-to-Fingertip Measurements

The wrist-to-fingertip configuration performed worst overall. The proximal wrist signal was visibly weaker and noisier than the fingertip signal, and the number of matched beats was very low. In one trial, only four matched beats survived, producing a mean PTT

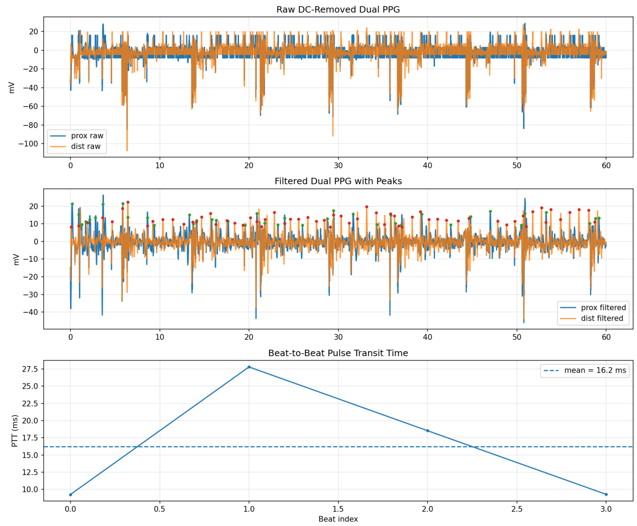


Figure 1: Representative wrist-to-fingertip result. The waveform quality at the wrist was weak relative to the fingertip, and only a small number of beat pairs survived the matching stage.

of 16.2 ms. In the second trial, only two matched beats survived and the mean shifted to 98.4 ms. That large swing is not physiologically convincing and instead indicates threshold sensitivity and poor proximal waveform quality.

Taken together, the wrist trials indicate that this was not a reliable proximal location for the reflective dual-PPG system used here.

4.2 Finger-Root-to-Fingertip Measurements

The finger-root-to-fingertip configuration produced the strongest results in the study. Both trials yielded substantially more matched beats than the wrist setup, and the resulting PTT values remained in a short and plausible range for a same-hand short-path geometry. Trial 1 produced 19 matched beats and a mean PTT of 14.4 ms. Trial 2 produced 14 matched beats and a mean PTT of 23.2 ms.

Within this single-subject dataset, finger-root-to-fingertip provided the best overall combination of waveform quality, matched-beat yield, and timing stability.

4.3 Palm-to-Fingertip Measurements

The palm-to-fingertip geometry produced usable measurements but was less consistently strong than the best finger-root trial. The two palm trials gave mean PTT values of 20.4 ms and 14.1 ms. These values were plausible, and the signals were measurable, but the palm placement did not clearly outperform the finger-root configuration.

The palm therefore appears to be a workable location, but not the strongest one among those tested.

4.4 Ear-to-Fingertip Measurements

The ear-to-fingertip configuration produced the largest mean delays in the dataset, especially in trial 2, which had a mean PTT of

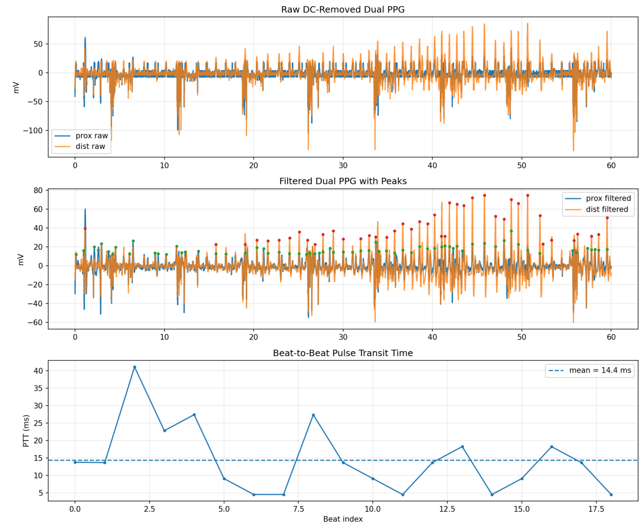


Figure 2: Representative finger-root-to-fingertip result. This configuration produced the cleanest overall measurement and the highest confidence in beat pairing.

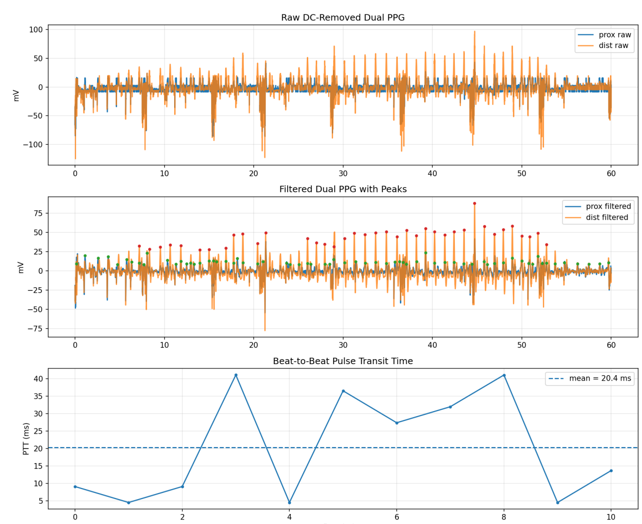


Figure 3: Representative palm-to-fingertip result. The waveform was measurable and the timing estimate was usable, though less clearly robust than the strongest finger-root trial.

43.8 ms. This is directionally consistent with a longer anatomical path between the ear and the finger. However, the two ear trials differed substantially from each other (22.6 ms versus 43.8 ms), indicating that ear placement and contact stability had a strong influence on the result.

Thus, ear-to-fingertip measurements were physiologically interesting because they yielded longer delays, but they were less repeatable than the best same-hand short-path configuration.

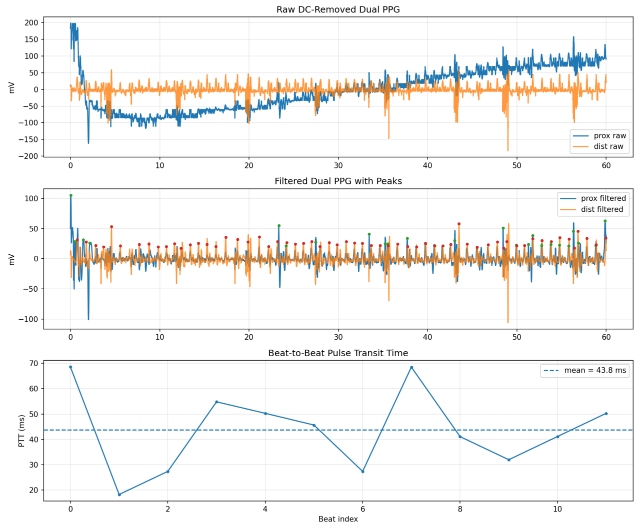


Figure 4: Representative ear-to-fingertip result. This geometry produced larger mean delays than the same-hand placements, but the results varied noticeably across repeated trials.

4.5 Key Findings

Three main observations emerged:

(1) **Placement mattered strongly.** The same hardware and analysis pipeline produced dramatically different reliability depending on body location.
(2) **Same-hand short-path sensing performed best.** In this dataset, finger-root-to-fingertip gave the most stable and repeatable results.
(3) **Longer-path sensing was measurable but more variable.** Ear-to-fingertip produced larger delays, but repeated trials differed substantially.

5 Discussion and Conclusion

This project implemented a complete dual-PPG sensing and analysis platform using a Raspberry Pi and low-cost reflective optical sensors, then used that platform to compare practical wearable body locations for pulse transit time measurement. The most important outcome is that measurement reliability depended strongly on body location rather than on the mere presence of two pulse waveforms.

The weakest location was the wrist. Its waveform quality was poor, beat matching was sparse, and the resulting delay estimates were highly unstable across repeated trials. In contrast, the finger-root-to-fingertip geometry produced the most convincing overall result in this single-subject dataset. Palm-to-fingertip was usable but less clearly superior, while ear-to-fingertip demonstrated that larger delays can be measured across longer body paths, albeit with increased variability.

The project also exposed several important limitations. First, the effective sampling rate of roughly 218 Hz limited timing resolution to a few milliseconds. Second, the ADC channels were read sequentially rather than simultaneously, which introduces an inter-channel timing skew that matters when the measured delays themselves are only on the order of tens of milliseconds. Third, the processing pipeline relied on simple empirically tuned smoothing, peak detection, and outlier rejection rather than a formally optimized or clinically validated fiducial method; the minimum beat-pairing delay was set to 4 ms, which is comparable to the inter-channel hardware skew, meaning a small fraction of matched pairs may partially reflect acquisition timing offsets rather than true physiological transit. Finally, the study included only one subject and two trials per configuration, so the results should be interpreted as an exploratory reliability comparison rather than a statistically definitive ranking of all wearable locations.

Even with those limitations, the project achieved its core objective. It built a working dual-PPG measurement system, collected real physiological data, and showed that some practical wearable placements are substantially more reliable than others. Within this dataset, the best overall placement was finger-root-to-fingertip, while wrist-to-fingertip was the least reliable.

Future work could improve the system by increasing the effective sampling rate, implementing truly simultaneous or better time-aligned acquisition, adding ECG or PCG as a proximal timing reference, using a more robust waveform fiducial such as the pulse foot instead of a simple peak, collecting more trials across multiple subjects, and calibrating the resulting timing features against cuff-based blood-pressure measurements.

6 Team Member Contributions

Ronny Abdullah designed the project, assembled the sensing platform, implemented the Raspberry Pi acquisition code, wrote the signal-processing and PTT-analysis scripts, collected the experimental data, generated the figures, and wrote the final report and presentation materials.

References

[1] John Allen. 2007. Photoplethysmography and its application in clinical physiological measurement. *Physiological Measurement* 28, 3 (2007), R1–R39.
[2] Mohamed Elgendi. 2012. On the analysis of fingertip photoplethysmogram signals. *Current Cardiology Reviews* 8, 1 (2012), 14–25.
[3] Toshiyo Tamura, Yuka Maeda, Masaki Sekine, and Masaki Yoshida. 2014. Wearable photoplethysmographic sensors—past and present. *Electronics* 3, 2 (2014), 282–302.
[4] Ramakrishna Mukkamala, Jin-Oh Hahn, Oscar T. Inan, Louis K. Mestha, Chang-Sei Kim, Joseph P. Toreyin, and Seulki Kyal. 2015. Toward ubiquitous blood pressure monitoring via pulse transit time: theory and practice. *IEEE Transactions on Biomedical Engineering* 62, 8 (2015), 1879–1901.
[5] Texas Instruments. 2013. *ADS1255/ADS1256 Very Low Noise, 24-Bit Analog-to-Digital Converter Datasheet.*

A Source Code

dual_ppg_record.py

```
import RPi.GPIO as GPIO
import time
import os
import csv
import numpy as np
import ADS1256

BASE_DIR = os.path.dirname(os.path.abspath(__file__))
DATA_DIR = os.path.join(os.path.dirname(BASE_DIR), "data"
    )
os.makedirs(DATA_DIR, exist_ok=True)

VREF = 3.3
DRATE_CODE = 0x92   # 500 SPS, same as EE217 scripts
GAIN_MAP = {1: 0, 2: 1, 4: 2, 8: 3, 16: 4, 32: 5, 64: 6}

CH_PROX = 0   # decide which sensor is closer to heart
CH_DIST = 1   # decide which sensor is farther from heart


def setup_adc(gain=1):
    adc = ADS1256.ADS1256()
    adc.ADS1256_init()
    adc.ADS1256_ConfigADC(GAIN_MAP[gain], DRATE_CODE)
    adc.ADS1256_WriteReg(0x00, 0x02)   # STATUS register
    adc.ADS1256_WriteCmd(0xF0)         # self-calibration
    time.sleep(0.1)
    return adc


def read_channel_voltage(adc, channel, gain):
    adc.ADS1256_SetChannal(channel)
    adc.ADS1256_WriteCmd(0xFC)   # SYNC
    adc.ADS1256_WriteCmd(0x00)   # WAKEUP
    raw = adc.ADS1256_Read_ADC_Data()
    scale = VREF / (gain * 0x7FFFFF)
    return raw * scale


def collect_dual(duration_sec=60, gain=1, filename="
    trial_rest_02.csv"):
    adc = setup_adc(gain=gain)

    for _ in range(10):
        read_channel_voltage(adc, CH_PROX, gain)
        read_channel_voltage(adc, CH_DIST, gain)

    input("Place both sensors, stay still, then press 
        Enter to begin... ")

    rows = []
    t0 = time.perf_counter()

    while True:
        now = time.perf_counter()
        t = now - t0
        if t >= duration_sec:
            break

        v0 = read_channel_voltage(adc, CH_PROX, gain)
        v1 = read_channel_voltage(adc, CH_DIST, gain)

        rows.append([t, v0, v1])

    out_path = os.path.join(DATA_DIR, filename)
    with open(out_path, "w", newline="") as f:
        writer = csv.writer(f)
        writer.writerow(["time_s", "ppg_prox_v", "
            ppg_dist_v"])
        writer.writerows(rows)

    print(f"Saved {len(rows)} samples to {out_path}")

    if len(rows) > 1:
        dt = rows[-1][0] - rows[0][0]
        fs_eff = (len(rows) - 1) / dt
        print(f"Effective sample rate = {fs_eff:.2f} Hz")

    GPIO.cleanup()


if __name__ == "__main__":
    fname = input("Enter filename (example: trial_rest_02
        .csv): ")
    collect_dual(duration_sec=60, gain=1, filename=fname)
```

analyze_ptt.py

```
import os
import csv
import numpy as np

# Try plotting if matplotlib exists
HAVE_PLOT = True
try:
    import matplotlib
    matplotlib.use("Agg")
    import matplotlib.pyplot as plt
except Exception:
    HAVE_PLOT = False

BASE_DIR = os.path.dirname(os.path.abspath(__file__))
ROOT_DIR = os.path.dirname(BASE_DIR)
DATA_DIR = os.path.join(ROOT_DIR, "data")
FIG_DIR = os.path.join(ROOT_DIR, "figures")
RES_DIR = os.path.join(ROOT_DIR, "results")

os.makedirs(FIG_DIR, exist_ok=True)
os.makedirs(RES_DIR, exist_ok=True)


def load_csv(path):
    t = []
    x0 = []
    x1 = []
    with open(path, "r") as f:
        reader = csv.DictReader(f)
        for row in reader:
            t.append(float(row["time_s"]))
            x0.append(float(row["ppg_prox_v"]))
            x1.append(float(row["ppg_dist_v"]))
    return np.array(t), np.array(x0), np.array(x1)


def estimate_fs(t):
    dt = np.diff(t)
    return 1.0 / np.median(dt)


def moving_average(x, n):
    if n < 1:
        return x.copy()
    kernel = np.ones(n) / n
    return np.convolve(x, kernel, mode="same")


def bandpass_like_ppg(x, fs):
    # simple practical substitute for bandpass without
        scipy:
    # 1) remove DC
    # 2) remove very slow drift with long moving average
    # 3) smooth high-frequency noise with short moving
        average
    x = x - np.mean(x)
    long_n = max(3, int(fs * 0.75))   # ~0.75 s trend
    short_n = max(3, int(fs * 0.03))  # ~30 ms smoothing
    trend = moving_average(x, long_n)
    y = x - trend
    y = moving_average(y, short_n)
```

```
    return y


def find_peaks_simple(x, min_distance, threshold):
    peaks = []
    last_peak = -min_distance

    for i in range(1, len(x) - 1):
        if i - last_peak < min_distance:
            continue
        if x[i] > threshold and x[i] > x[i - 1] and x[i]
             >= x[i + 1]:
            # refine within local window
            win_left = max(0, i - 2)
            win_right = min(len(x), i + 3)
            local_idx = np.argmax(x[win_left:win_right])
                 + win_left
            peaks.append(local_idx)
            last_peak = local_idx

    # remove duplicates if any
    peaks = np.array(sorted(set(peaks)), dtype=int)
    return peaks


def pair_peaks(t, peaks0, peaks1, min_delay_s=0.0,
     max_delay_s=0.25):
    pairs = []
    j = 0
    for p0 in peaks0:
        while j < len(peaks1) and peaks1[j] <= p0:
            j += 1
        if j < len(peaks1):
            delay = t[peaks1[j]] - t[p0]
            if min_delay_s < delay < max_delay_s:
                pairs.append((p0, peaks1[j], delay))
    return pairs


def save_summary(path, fs, peaks0, peaks1, ptt_ms):
    with open(path, "w") as f:
        f.write(f"fs_hz = {fs:.3f}\n")
        f.write(f"prox_peaks = {len(peaks0)}\n")
        f.write(f"dist_peaks = {len(peaks1)}\n")
        f.write(f"matched_beats = {len(ptt_ms)}\n")
        if len(ptt_ms) > 0:
            f.write(f"mean_ptt_ms = {np.mean(ptt_ms):.3f
                 }\n")
            f.write(f"std_ptt_ms = {np.std(ptt_ms):.3f}\n
                 ")
            f.write(f"median_ptt_ms = {np.median(ptt_ms)
                 :.3f}\n")
            f.write(f"min_ptt_ms = {np.min(ptt_ms):.3f}\n
                 ")
            f.write(f"max_ptt_ms = {np.max(ptt_ms):.3f}\n
                 ")


def main(filename="data/trial_finger_rest_01.csv"):
    path = os.path.join(DATA_DIR, filename)
    t, x0, x1 = load_csv(path)

    fs = estimate_fs(t)

    x0f = bandpass_like_ppg(x0, fs)
    x1f = bandpass_like_ppg(x1, fs)

    min_dist = max(1, int(0.35 * fs))  # ~350 ms minimum
         spacing
    thr0 = 0.20 * np.max(x0f)
    thr1 = 0.30 * np.max(x1f)

    peaks0 = find_peaks_simple(x0f, min_dist, thr0)
    peaks1 = find_peaks_simple(x1f, min_dist, thr1)

    pairs = pair_peaks(t, peaks0, peaks1, min_delay_s
         =0.004, max_delay_s=0.080)
    ptt_ms = np.array([1000.0 * p[2] for p in pairs])

    if len(ptt_ms) > 0:
        med = np.median(ptt_ms)
        keep = np.abs(ptt_ms - med) < 30.0
        ptt_ms = ptt_ms[keep]

    print(f"Estimated fs = {fs:.2f} Hz")
    print(f"Prox peaks = {len(peaks0)}")
    print(f"Dist peaks = {len(peaks1)}")
    print(f"Matched beats = {len(ptt_ms)}")

    if len(ptt_ms) > 0:
        print(f"Mean PTT = {np.mean(ptt_ms):.2f} ms")
        print(f"Std PTT = {np.std(ptt_ms):.2f} ms")
        print(f"Median PTT = {np.median(ptt_ms):.2f} ms")
        print(f"Min PTT = {np.min(ptt_ms):.2f} ms")
        print(f"Max PTT = {np.max(ptt_ms):.2f} ms")
    else:
        print("No valid PTT matches found.")

    summary_path = os.path.join(RES_DIR, filename.replace
         (".csv", "_ptt_summary.txt"))
    save_summary(summary_path, fs, peaks0, peaks1, ptt_ms
         )
    print(f"Saved summary to {summary_path}")

    if HAVE_PLOT:
        fig, axes = plt.subplots(3, 1, figsize=(12, 10))

        axes[0].plot(t, 1000 * (x0 - np.mean(x0)), label=
             "prox_raw")
        axes[0].plot(t, 1000 * (x1 - np.mean(x1)), label=
             "dist_raw", alpha=0.8)
        axes[0].set_title("Raw DC-Removed Dual PPG")
        axes[0].set_ylabel("mV")
        axes[0].legend()
        axes[0].grid(True, alpha=0.3)

        axes[1].plot(t, 1000 * x0f, label="prox_filtered"
             )
        axes[1].plot(t, 1000 * x1f, label="dist_filtered"
             , alpha=0.8)
        if len(peaks0) > 0:
            axes[1].plot(t[peaks0], 1000 * x0f[peaks0], "
                 o", markersize=3)
        if len(peaks1) > 0:
            axes[1].plot(t[peaks1], 1000 * x1f[peaks1], "
                 o", markersize=3)
        axes[1].set_title("Filtered Dual PPG with Peaks")
        axes[1].set_ylabel("mV")
        axes[1].legend()
        axes[1].grid(True, alpha=0.3)

        if len(ptt_ms) > 0:
            axes[2].plot(ptt_ms, ".-")
            axes[2].axhline(np.mean(ptt_ms), linestyle="
                 --", label=f"mean = {np.mean(ptt_ms):.1f
                 } ms")
            axes[2].legend()
        axes[2].set_title("Beat-to-Beat Pulse Transit 
             Time")
        axes[2].set_xlabel("Beat index")
        axes[2].set_ylabel("PTT (ms)")
        axes[2].grid(True, alpha=0.3)

        fig.tight_layout()
        fig_path = os.path.join(FIG_DIR, filename.replace
             (".csv", "_ptt.png"))
        fig.savefig(fig_path, dpi=150)
        plt.close(fig)
        print(f"Saved figure to {fig_path}")
    else:
        print("matplotlib not available, skipping figures
             .")


if __name__ == "__main__":
    main("trial_ear_finger_02.csv")
```

B Presentation Slides

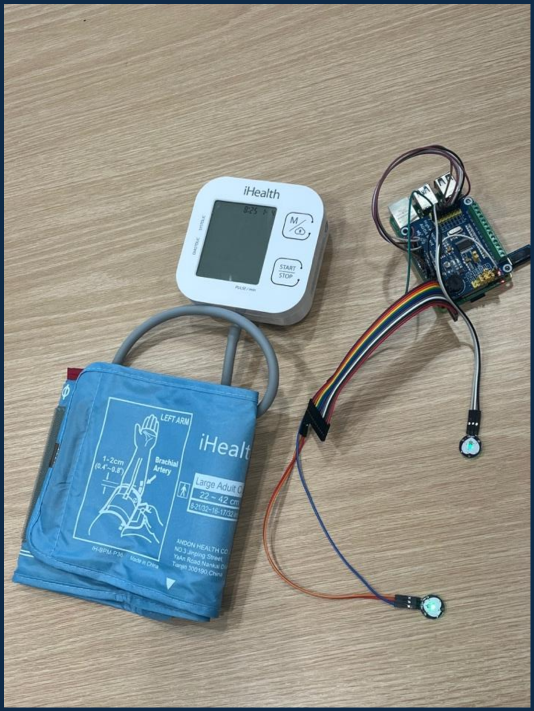
EE392D: WIRELESS SENSING SYSTEMS

Evaluating Wearable Sensor Locations for Dual-PPG Pulse Transit Time Measurement

Professor Zerina Kapetanovic

Ronny Abdullah

Stanford University · Final Project Presentation



Introduction

The Problem

High blood pressure is a leading cardiovascular condition worldwide. Traditional cuff-based monitors can't provide continuous or wearable monitoring.

Our Approach

Pulse Transit Time (PTT) enables cuffless BP estimation. Two optical PPG sensors at different body sites measure the pulse wave arrival delay.

Project Goal

Build a dual-PPG system on Raspberry Pi and find which wearable sensor placements yield the most stable and repeatable PTT measurements.

Background: PPG & Pulse Transit Time

What is PPG?

Photoplethysmography is an optical technique measuring blood volume changes in tissue.

- Sensor emits light into skin
- Reflected light intensity varies with blood flow
- Each heartbeat creates a visible peak in signal
- Sampling rate: 218 Hz · Trial duration: ~60 s

What is PTT?

Pulse Transit Time = delay for a pulse wave to travel between two measurement sites:

PTT = t_distal – t_proximal

↑ BP → faster pulse wave → **smaller PTT**
↓ BP → slower pulse wave → **larger PTT**

Proximal sensor = closer to heart
Distal sensor = farther from heart

System Implementation

Hardware

- Raspberry Pi (data acquisition + processing)
- 2× Optical reflective PPG sensors
- Analog-to-Digital Converter (ADC)
- 218 Hz sampling · ~60 s per trial

Signal Processing Pipeline

① Raw dual-PPG acquisition
② DC removal (baseline offset)
③ Bandpass filtering
④ Peak detection per sensor
⑤ Beat matching (proximal ↔ distal)
⑥ **PTT = t_distal – t_proximal**

Sensor Configurations Evaluated

Wrist → Fingertip
High motion sensitivity, lower perfusion

Finger Root → Fingertip
Short, same-finger arterial path

Palm → Fingertip
Mid-hand placement, moderate path

Earlobe → Fingertip
Cross-body path, longer arterial distance

Challenges & Solutions

⚠ Motion Artifacts

Small movements caused large signal disturbances, especially at the wrist — the dominant noise source across all trials.

→ Stabilized sensors physically during recording; kept participant still.

⚠ Sensor Contact Pressure

Too little pressure → weak signal. Too much pressure → blood flow obstruction. Both drastically degrade signal quality.

→ Consistent moderate pressure required for each trial placement.

⚠ Peak Detection Errors

Noise and waveform distortion led to false or missed peaks, causing beat mismatches between the proximal and distal sensors.

→ Tuned detection thresholds; only matched beat pairs used for PTT.

⚠ Weak Wrist PPG Signal

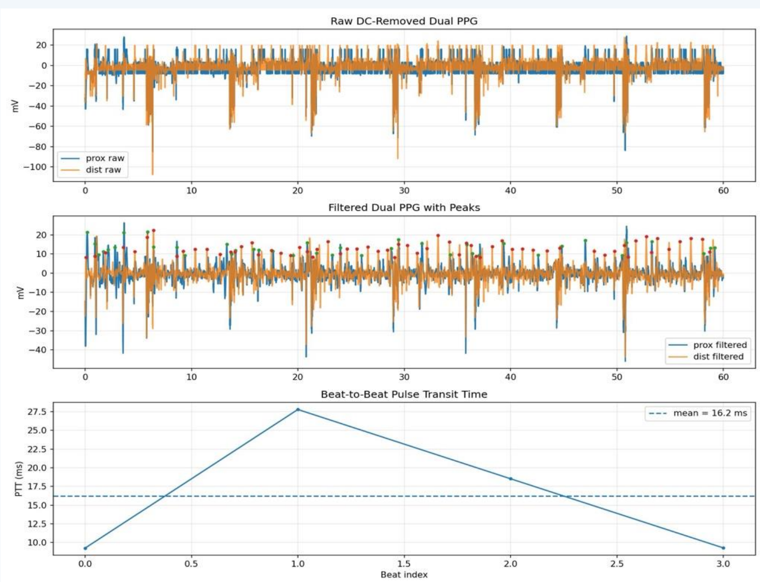
Wrist has lower microvascular perfusion than fingers or ear, producing a faint and noisy waveform that challenges peak detection.

→ Applied bandpass filtering; wrist ultimately flagged as unsuitable.

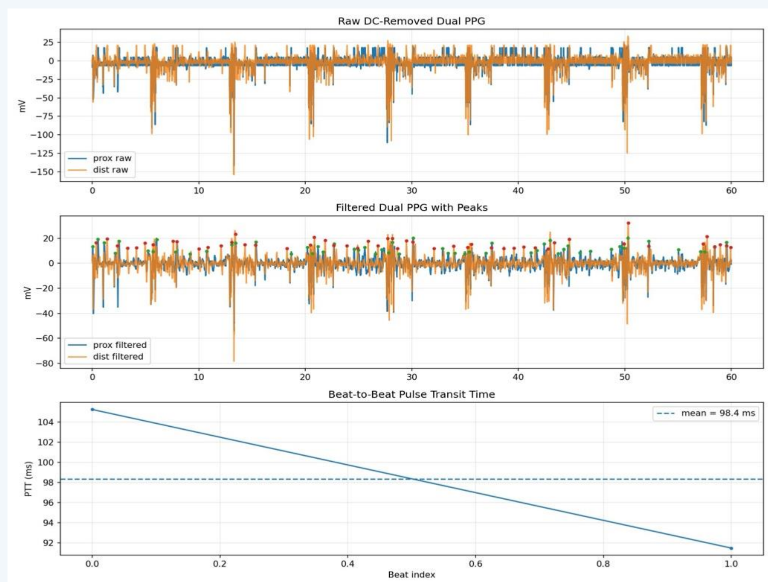
Results: Wrist → Fingertip

Very few matched beats across both trials. PTT values swung wildly from ~16 ms to ~98 ms — wrist reflective PPG is too noisy and motion-sensitive for reliable PTT.

✗ POOR RELIABILITY



Trial 1 | Mean PTT: 16.2 ms | 3 matched beats

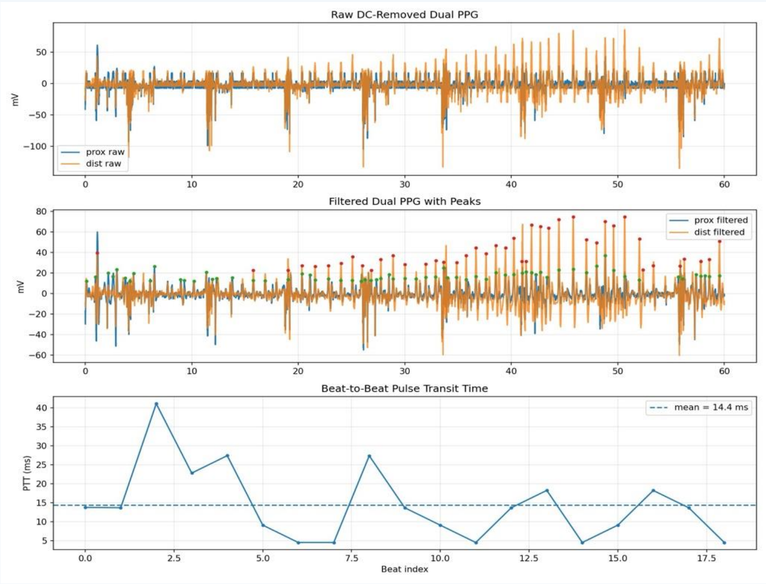


Trial 2 | Mean PTT: 98.4 ms | 2 matched beats

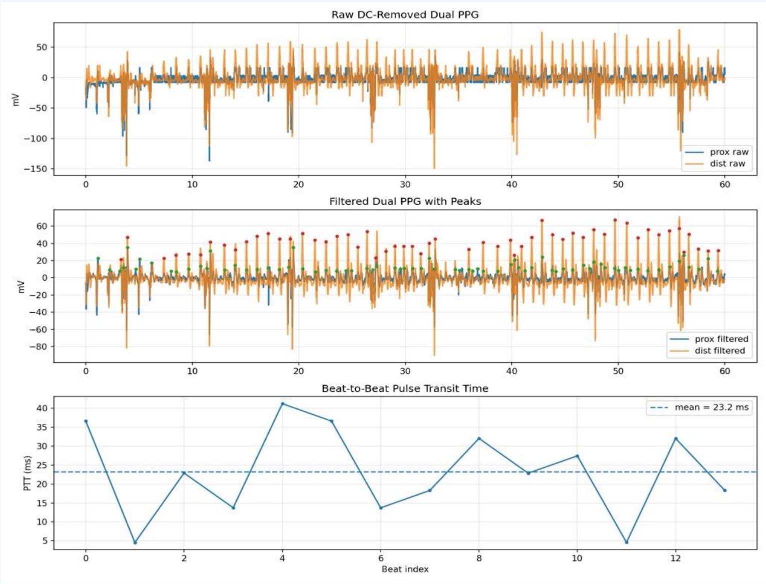
Results: Finger Root → Fingertip

Most stable configuration overall. Short, consistent arterial path within the same finger. Both trials produced many matched beats with tightly clustered PTT values (14–23 ms).

★ BEST RELIABILITY



Trial 1 | Mean PTT: 20.4 ms | 11 matched beats

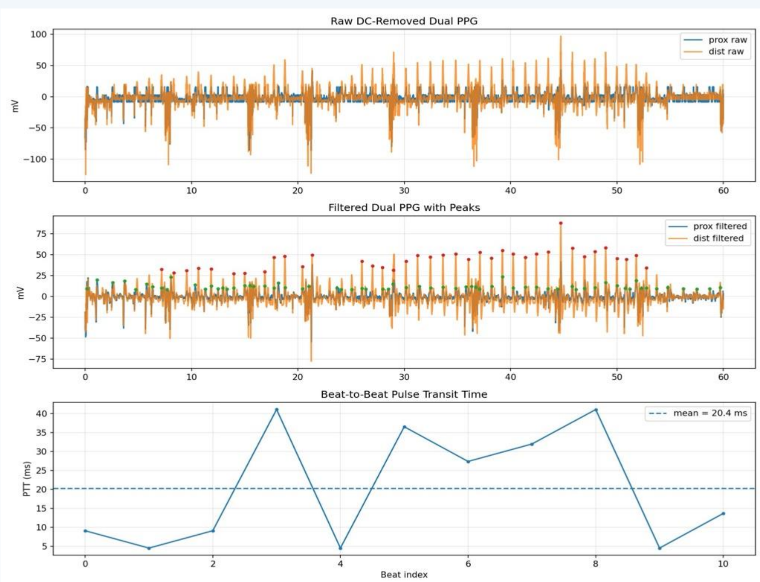


Trial 2 | Mean PTT: 14.1 ms | 11 matched beats

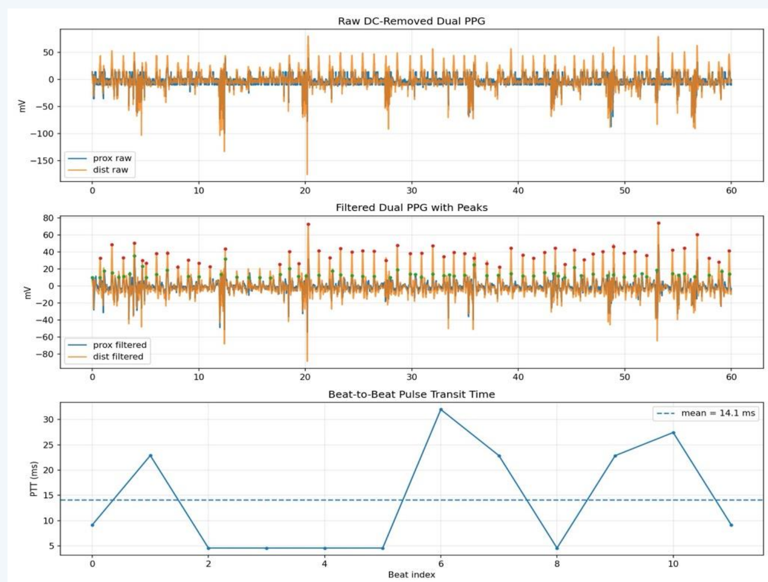
Results: Palm → Fingertip

Usable signals with slightly more variability than finger-to-finger. Palm sensor is harder to stabilize. Mean PTT (14–23 ms) is physiologically consistent with finger results.

◑ MODERATE RELIABILITY



Trial 1 | Mean PTT: 14.4 ms | 19 matched beats

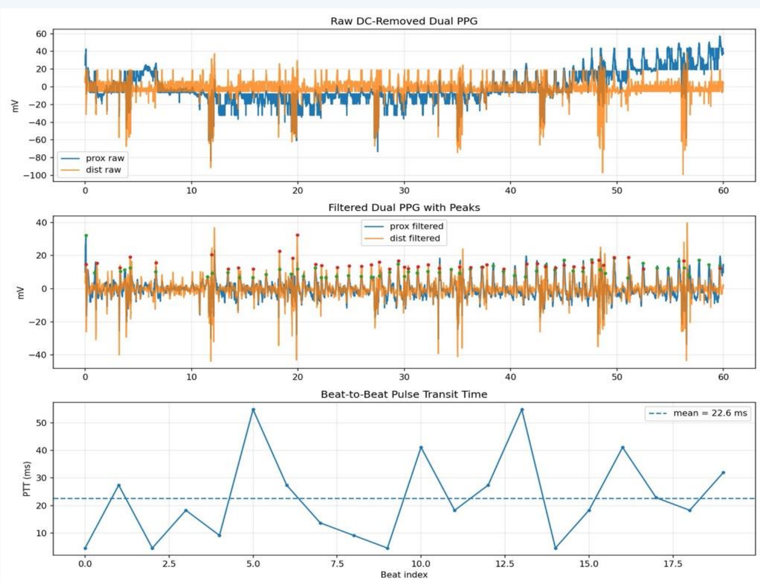


Trial 2 | Mean PTT: 23.2 ms | 14 matched beats

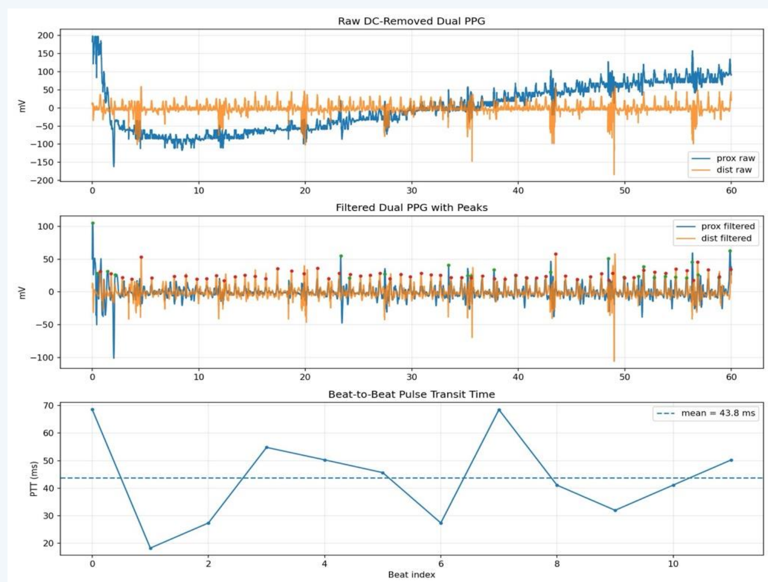
Results: Earlobe → Fingertip

Longer PTT reflects the greater arterial distance from ear to fingertip. Trial 2 captures the expected ~44 ms delay. Physiologically meaningful for cross-body BP estimation.

✓ GOOD — LONGER PATH



Trial 1 | Mean PTT: 22.6 ms | 20 matched beats



Trial 2 | Mean PTT: 43.8 ms | 11 matched beats

Results Summary

Sensor Placement	Trial 1 PTT	Trial 2 PTT	Matched Beats	Reliability
Wrist → Fingertip	16.2 ms	98.4 ms	2–3 beats	✗ POOR
Finger Root → Fingertip	20.4 ms	14.1 ms	~11 beats each	★ BEST
Palm → Fingertip	14.4 ms	23.2 ms	14–19 beats	◑ MODERATE
Earlobe → Fingertip	22.6 ms	43.8 ms	11–20 beats	✓ GOOD

Key Insight: Shorter same-limb arterial paths provide the most stable timing measurements. Longer cross-body paths (ear→finger) introduce variability but remain physiologically meaningful.

Conclusion & Future Work

Key Findings

Finger Root → Fingertip
Most stable PTT; best for wearable use

Palm → Fingertip
Moderately reliable; usable signal quality

Earlobe → Fingertip
Good; longer ~44 ms delay — physiologically valid

Wrist → Fingertip
Highly unreliable; unsuitable for PTT sensing

Future Work

→ Add ECG sensing for precise pulse arrival timing

→ Improve sensor mounting to reduce motion artifacts

→ Test with multiple participants for generalizability

→ Explore ML models to estimate BP directly from PTT