

Rotation Angle Estimation for Visually Impaired Assistance Using Camera and IMU Sensors

Projektarbeit in Computer Science

submitted
by

Isa Sadigli

born 06.03.1999 in Baku, Azerbaijan

Written at

Lehrstuhl für Mustererkennung (Informatik 5)
Department Informatik
Friedrich-Alexander-Universität Erlangen-Nürnberg.

Advisor: Hakan Calim

Started: Start

Finished: Ende

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Übersicht

Diese Studie demonstriert die präzise Schätzung von Gierwinkeln, um blinden oder sehbehinderten Menschen (BVIPs) bei der Bestimmung ihrer räumlichen Orientierung zu helfen. Die Studie bewertete zwei Hauptkategorien von Verfahren, die in praktischen Anwendungen eingesetzt werden: kamerabasierte optische Flussverfahren und sensorbasierte Datenerfassungsmethoden. Kamerabasierte Verfahren, die optischen Fluss nutzen, funktionierten in kontrollierten Umgebungen gut, konnten jedoch aufgrund ihres hohen Speicher- und Rechenaufwands nicht in der Praxis eingesetzt werden. Um diese Einschränkungen zu überwinden, wurde eine integrierte Android-Anwendung entwickelt, die die gleichzeitige Erfassung von Videosequenzen und Sensormessungen von Gyroskopen, Magnetometern und GPS-Modulen ermöglicht. Jeder Sensor wurde evaluiert, und es zeigte sich, dass GPS-Daten ein erhebliches Problem aufweisen. Selbst unter optimalen Bedingungen treten erhebliche Signallücken auf, was die Daten für Aufgaben, die eine genaue Richtung erfordern, unzuverlässig macht. Im Gegensatz dazu zeigten Sensorfusionsverfahren mit komplementärer Filterung zur Kombination von Gyroskop- und Magnetometerdaten in verschiedenen Testumgebungen eine höhere Genauigkeit und Robustheit. Die Fusion dieser Sensoren lieferte konsistente und zuverlässige Gierwinkelschätzungen. Diese Arbeit fördert die Entwicklung belastbarer und zuverlässiger Hilfsmittel zur räumlichen Orientierung, die in verschiedenen realen Betriebsszenarien eine hohe Leistung erbringen können.

Abstract

This study demonstrates the accurate estimation of yaw rotation angles to assist blind or visually impaired persons (BVIPs) in determining their spatial orientation. The study evaluated two primary categories of processes employed in practical applications: camera-based optical flow techniques and sensor-based data acquisition methods. Camera-based methods that used optical flow worked well in controlled settings, however they couldn't be used in real life since they used a lot of memory and computational time. To overcome these constraints, an integrated Android application was developed, enabling concurrent acquisition of video sequences and sensor measurements from gyroscopes, magnetometers, and GPS modules. Each sensor was evaluated and it is found that GPS data has a major problem. There are significant signal gaps that happen even when conditions are perfect which makes it unreliable for jobs that require exact direction. In contrast, sensor fusion techniques employing complementary filtering to combine gyroscope and magnetometer data demonstrated superior accuracy and robustness across diverse testing environments. The

fusion of these sensors gave consistent and reliable yaw angle estimates. This work advances the development of resilient and reliable spatial orientation aids capable of maintaining high performance across varied real-world operational scenarios.

Contents

1	Introduction	1
2	Related Work	3
2.1	Camera-Based Rotation Angle Estimation	3
2.2	Sensor-Based Rotation Angle Calculation	3
3	Methodology	5
3.1	Camera-Based Approach	5
3.2	Sensor-Based Approach	12
4	Results	15
4.1	Static In-Place Rotation (Left Turn)	15
4.2	Bidirectional In-Place Rotation	17
4.3	Translation Combined with Rotation	19
4.4	Complex Diagonal Motion with Rotation	21
4.5	Performance	23
4.6	Discussion	24
5	Future Directions and Conclusion	25
5.1	Future Directions	25
5.2	Conclusion	26
List of Abbreviations		27
List of Figures		29
List of Tables		31
Bibliography		33

Chapter 1

Introduction

Recent advancements in technology have made it possible to construct more sophisticated assistive technology. GPS-based navigation systems have been introduced, but they suffer from limited accuracy in urban environments, with errors of up to 15 meters and frequent signal loss [Tap⁺17]. Computer vision-based approaches offer superior interpretation of real scenes but often require significant computational resources, making real-time implementation challenging on mobile devices.

For BVIP, the correct orientation is an important part of navigation. Knowing your rotation angle in relation to a reference point or direction can greatly improve your ability to think about space and navigate with confidence. This is especially important in dynamic environments where users must constantly adjust their orientation. The ability to accurately measure rotation angles can help BVIP maintain their intended path, avoid disorientation, and make appropriate directional adjustments when navigating.

The main goal of this project is to find strong ways to calculate rotation angles for BVIP using both camera-based and sensor-based methods. First, an optical flow-based approach was explored to estimate camera rotation via planar motion estimation. By tracking the motion of features across consecutive frames and fitting a 2D affine transformation, yaw rotation was approximated. While this method doesn't recover full 3D pose, it proved to be a simple and efficient way to estimate rotational change under controlled conditions. The implementation was comparing features between the first frame and the current frame. But this strategy didn't work for huge rotation degrees because optical flow algorithms have difficulty matching characteristics between frames that are very different in angle. As noted by Kunyankin [Kun20], optical flow techniques are limited by their inability to identify corresponding features when the viewpoint changes substantially, such as when comparing

the reference frame (where the rotation is 0°) and frame with significant rotation angle (for example 90°).

To overcome this issue, I modified the camera-based method to employ an accumulative computation technique. Instead of directly comparing the current frame with an initial reference frame, it compares successive frames and sums the rotation angles between them. For slow, steady rotations where a person remains stationary and turns slowly, this step-by-step method works well. However, when the person starts to walk, the accumulative approach revealed another limitation: it amplifies noise over time. Even in the absence of actual rotation, the accumulated calculation shows phantom rotation due to the compounding of small measurement errors and motion-induced noise.

To address the limitations of planar affine-based yaw estimation, I developed a method combining essential matrix decomposition with camera intrinsics. This method detects keypoints between consecutive frames and adds yaw angles from the recovered 3D rotations, resulting in more accurate results for slow, in-place movements. But like the original optical flow-based technology, it uses lots of memory and takes long time to compute, which limits its use in real-time or embedded systems until it is optimized further.

To overcome these challenges inherent to vision-based methods, I implemented a sensor-based approach that uses the inertial measurement unit (IMU) sensors because of its cheap prices. This approach processes data from gyroscopes and magnetometers to calculate yaw rotation angles. Complementary filter is used to combine data from the gyroscope and magnetometer. This took advantage of the strengths of each sensor while limiting their weaknesses.

I developed an Android application to simultaneously record video and sensor data, allowing me to directly compare both approaches under identical conditions. This method combines gyroscope and magnetometer data to determine how much the yaw angle has changed. Through testing, it was found that while the GPS data contained significant gaps, making it unsuitable for rotation calculations, the fusion of gyroscope and magnetometer data using a complementary filter provided the most reliable results.

The following sections detail the methodology, results, future directions, and conclusions. A thorough evaluation of both methods is provided, including their advantages and disadvantages, as well as suggestions for improving BVIP's orientation and navigation skills in the future.

Chapter 2

Related Work

2.1 Camera-Based Rotation Angle Estimation

Camera-based approaches to rotation angle estimation have evolved significantly with advancements in computer vision algorithms. These methods are extremely useful for BVIPs because they can be used with standard mobile device cameras and do not require any additional hardware.

Optical flow approaches determine how pixels move between frames, which can be separated into translational and rotational components. Lucas-Kanade [Luc⁺81] developed one of the most influential optical flow algorithms, tracking sparse feature points across frames. Optical flow algorithms struggle when faced with broad rotation angles because they frequently fail to recognize matching features when the viewpoint changes rapidly [Bea⁺95].

Feature-based methods track distinctive points across frames to estimate camera motion parameters, including rotation. SIFT [Low04] and SURF [Bay⁺08] algorithms detect scale-invariant features that are partially robust to viewpoint changes, making them potentially suitable for rotation estimation.

2.2 Sensor-Based Rotation Angle Calculation

Inertial Measurement Units (IMUs) have become central to rotation angle calculation due to their ability to capture angular velocity and orientation changes with high temporal resolution. The most modern IMUs have three sensors that work together in different ways to estimate rotation. Gyroscopes measure angular velocity directly, which makes

them appear like the best way of finding yaw rotation. However, as Woodman [Woo07] demonstrated, gyroscope data must be integrated over time to determine angle changes, leading to significant drift. Magnetometers measure the Earth’s magnetic field direction, functioning effectively as digital compasses. However, magnetic interference from electronic devices and ferromagnetic materials can introduce errors [Bac⁺01].

The fact that each sensor has its own strengths and weaknesses has led to the development of fusion methods that are especially designed to accurately calculate the rotation angle. These approaches can be categorized by their underlying mathematical frameworks.

Tang et al. [Tan⁺15] demonstrated that complementary filtering approach could achieve a high rotation angle accuracy with minimal computational overhead, even during dynamic movement. Kalman filtering represents a more sophisticated approach to sensor fusion. Sabatini [Sab06] developed a quaternion-based extended Kalman filter for IMU orientation estimation that explicitly modeled sensor noise characteristics. However, it’s computationally costly.

Despite significant advances, most systems perform poorly in diverse or challenging environments, especially in the presence of magnetic interference or on uneven surfaces. Many sophisticated algorithms require substantial computational resources, limiting their deployment on portable devices with battery constraints.

The current study covers these gaps by implementing and comparing camera-based and sensor-based rotation estimation methods, with a focus on how well they work in real-world situations, how quickly they can be computed, and how well they can be integrated. By evaluating both approaches systematically, we aim to identify the most promising direction for rotation angle estimation in assistive technologies for BVIPs.

Chapter 3

Methodology

This chapter presents the technical approaches developed to estimate yaw rotation angles through both camera-based and sensor-based systems.

3.1 Camera-Based Approach

The camera-based approach utilizes computer vision techniques to estimate a person's orientation by analyzing changes between consecutive video frames. This method used a smartphone camera put on the chest to shoot video at 30 frames per second at a resolution of 1920×1080 .

To enhance robustness and focus feature detection on relevant environmental elements, semantic segmentation was implemented to identify sidewalk areas in each frame. The Scale-Invariant Feature Transform (SIFT) algorithm was used to identify distinctive properties that remain constant regardless of scale, rotation, or light level changes [Low04]. A UNet++ architecture with an EfficientNet-B0 encoder was used to isolate the sidewalk regions, restricting SIFT feature detection to these stable surfaces. The feature detection was then restricted to these segmented regions:

Algorithm 1 Detect Keypoints and Descriptors using SIFT

Require: Color image $\mathbf{F} \in \mathbb{R}^{H \times W \times 3}$, mask $\mathbf{M} \in \{0, 1\}^{H \times W}$

Ensure: Set of keypoints \mathbf{k}

```

1: function DETECTKEYPOINTS( $\mathbf{F}, \mathbf{M}$ )
2:    $\mathbf{G} \leftarrow \text{GRAY}(\mathbf{F})$ 
3:    $\mathbf{k} \leftarrow \text{SIFT.detectAndCompute}(\mathbf{G}, \mathbf{M})$ 
4:   return  $\mathbf{k}$ 
5: end function

```

The function shown in 1 takes as input a 3 channel color image \mathbf{F} and a binary mask \mathbf{M} then outputs a set of detected keypoints \mathbf{k} . The process begins by converting the input image to grayscale, which reduces computational complexity while preserving essential structural information needed for feature detection. Subsequently, the system uses SIFT detection on the grayscale image with the mask to limit the detection region to the sidewalk segment of the image and optimize computing resources. The keypoints that result from this demonstrate important parts of the image with distinct local image patterns.

The first implementation shown in 2 used a direct comparison method between the current frame and a fixed reference frame (typically the first frame). The algorithm processes a sequence of frames $\{\mathbf{f}_t\}_{t=1}^T$ along with their corresponding keypoints $\{\mathbf{k}_t\}_{t=1}^T$ at specified intervals s , producing an angle vector $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_n]$ to represent rotational angle between frames. The method uses optical flow tracking to find matches between keypoints in the first frame (the reference frame) and the current frame. It converts the reference and current frames into grayscale before using the OpticalFlow algorithm to track the movement of keypoints from their original positions \mathbf{p}_0 to their new positions \mathbf{p}_1 . The Lucas-Kanade [Luc⁺81] algorithm is used for optical flow to track these features between frames. The algorithm constructs coordinate sets \mathbf{G}_{old} and \mathbf{G}_{new} from successfully tracked points (where tracking status $s_{tj} = 1$) and applies a robust filtering criterion requiring at least 3 valid correspondences in each set to ensure reliable geometric estimation. When sufficient correspondences exist, the method estimates an affine partial 2D transformation matrix \mathbf{M} between the point sets, from which the rotation angle is extracted using the arctangent function applied to the transformation matrix elements $M_{0,1}$ and $M_{0,0}$, with conversion from radians to degrees. This approach enables robust estimation of inter-frame rotation angles while handling tracking failures and maintaining computational efficiency through selective processing at specified intervals.

Algorithm 2 Track Keypoints and Compute Rotation Angles

Require: Frames $\{\mathbf{f}_t\}_{t=1}^T$, keypoints $\{\mathbf{k}_t\}_{t=1}^T$, interval s

Ensure: Angle vector $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_n]$

```

1: function TRACKKEYPOINTS( $\{\mathbf{f}_t\}$ ,  $\{\mathbf{k}_t\}$ ,  $s$ )
2:   Initialize  $\boldsymbol{\theta} \leftarrow []$ ,  $\theta_{last} \leftarrow 0$ ,  $ref \leftarrow 0$ 
3:    $N \leftarrow$  number of frames
4:   for  $i = 0$  to  $N - s$  step  $s$  do
5:      $\mathbf{I}_{old} \leftarrow \text{GRAY}(\mathbf{F}_{ref})$ ,  $\mathbf{I}_{new} \leftarrow \text{GRAY}(\mathbf{F}_{i+s})$ 
6:      $\mathbf{p}_0 \leftarrow \mathbf{k}_{ref}$  as coordinates
7:      $\mathbf{p}_1 \leftarrow \text{OPTICALFLOW}(\mathbf{I}_{old}, \mathbf{I}_{new}, \mathbf{p}_0)$ 
8:      $\mathbf{G}_{old} = \{p_{0j} \mid s_{tj} = 1\}$ 
9:      $\mathbf{G}_{new} = \{p_{1j} \mid s_{tj} = 1\}$ 
10:    if  $|\mathbf{G}_{old}| > 3$  and  $|\mathbf{G}_{new}| > 3$  then
11:       $\mathbf{M} \leftarrow \text{ESTIMATEAFFINEPARTIAL2D}(\mathbf{G}_{old}, \mathbf{G}_{new})$ 
12:      if  $\mathbf{M}$  is not None then
13:         $\theta \leftarrow -\arctan 2(M_{0,1}, M_{0,0}) \cdot \frac{180}{\pi}$ 
14:      end if
15:    end if
16:     $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} \cup \theta$ 
17:  end for
18:  return  $\boldsymbol{\theta}$ 
19: end function

```

However, this direct comparison approach led the feature correspondence to fail consistently at large rotation angles (beyond approximately 45°) when the features visible in the reference frame would move out of view. According to Kunyankin [Kun20], optical flow methods have struggle with finding matching features when the viewpoint changes dramatically, especially when comparing a reference frame (rotation at 0°) with significant rotational displacement (e.g., 90°).

To address the limitations of the direct comparison approach, an accumulative calculation method was developed that tracks incremental rotations between successive frames rather than comparing to a fixed reference. By this way, each frame is compared to its immediate predecessor, and these incremental angles are summed to obtain the total rotation.

Algorithm 3 Track Keypoints with Cumulative Angle and Drift Correction

Require: Frames $\{\mathbf{f}_t\}_{t=1}^T$, keypoints $\{\mathbf{k}_t\}_{t=1}^T$, interval s , scalar multiplier α

Ensure: Angle vector $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_n]$

```

1: function TRACKKEYPOINTSV1( $\mathbf{f}_t\}$ ,  $\{\mathbf{k}_t\}$ ,  $s$ ,  $\alpha$ )
2:   Initialize  $\boldsymbol{\theta} \leftarrow []$ ,  $\Theta \leftarrow 0$ ,  $ref \leftarrow 0$ 
3:    $N \leftarrow$  number of frames
4:   for  $i = 0$  to  $N - s$  step  $s$  do
5:      $\mathbf{I}_{old} \leftarrow \text{GRAY}(\mathbf{F}_{ref})$ ,  $\mathbf{I}_{new} \leftarrow \text{GRAY}(\mathbf{F}_{i+s})$ 
6:      $\mathbf{p}_0 \leftarrow \mathbf{k}_{ref}$  as coordinates
7:      $\mathbf{p}_1 \leftarrow \text{OPTICALFLOW}(\mathbf{I}_{old}, \mathbf{I}_{new}, \mathbf{p}_0)$ 
8:      $\mathbf{G}_{old} = \{p_{0j} \mid s_{tj} = 1\}$ 
9:      $\mathbf{G}_{new} = \{p_{1j} \mid s_{tj} = 1\}$ 
10:    if  $|\mathbf{G}_{old}| > 3$  and  $|\mathbf{G}_{new}| > 3$  then
11:       $\mathbf{M} \leftarrow \text{ESTIMATEAFFINEPARTIAL2D}(\mathbf{G}_{old}, \mathbf{G}_{new}, \text{RANSAC}, 3.0)$ 
12:      if  $\mathbf{M}$  is not None then
13:         $\theta_{last} \leftarrow -\arctan 2(M_{0,1}, M_{0,0}) \cdot \frac{180}{\pi} \cdot \alpha$ 
14:         $\Theta \leftarrow \Theta + \theta_{last}$ 
15:         $\Theta \leftarrow \text{CorrectDrift}(\Theta)$ 
16:      end if
17:    end if
18:     $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} \cup \theta$ 
19:     $ref \leftarrow i + s$ 
20:  end for
21:  return  $\boldsymbol{\theta}$ 
22: end function

```

The TrackKeypointsV1 function (shown in 3) extends the previous methodology by introducing a scalar multiplier α to prevent underestimation of rotation angle. The algorithm maintains a cumulative angle estimate Θ that accumulates rotational changes over time, with each instantaneous rotation $\boldsymbol{\theta}$ being scaled by the multiplier α before integration into the running total. Correcting drift that can accumulate over time made necessary to have CorrectDrift function (shown in 4) that addresses the inherent accumulation of estimation errors in sequential tracking systems through a threshold-based correction strategy. This drift correction mechanism takes the cumulative angle Θ , a drift threshold δ , and a correction factor λ as inputs, implementing a conditional correction scheme that activates when the

absolute cumulative angle exceeds the specified threshold. If drift correction is not needed ($|\Theta| < \delta$), the function keeps the original angle the same, if there is significant drift ($|\Theta| \geq \delta$), the algorithm implements a proportionate adjustment by determining a scaling factor $c = \lambda \cdot (|\Theta| - \delta)$ and subsequently reducing the cumulative angle magnitude via $\Theta' = \Theta - \text{sign}(\Theta) \cdot c$, which keeps the rotational direction while systematically decreasing the accumulated error. TrackKeypointsV1 function changes the reference frame index ref at each step of processing. This keeps tracking based on recent reliable keypoint distributions instead of spreading errors from faraway reference frames.

Algorithm 4 Drift Correction for Cumulative Angle

Require: Cumulative angle Θ , drift threshold δ , correction factor λ

Ensure: Corrected angle Θ'

```

1: function CORRECTDRIFT( $\Theta$ ,  $\delta$ ,  $\lambda$ )
2:   if  $|\Theta| < \delta$  then
3:     return  $\Theta$ 
4:   else
5:      $c \leftarrow \lambda \cdot (|\Theta| - \delta)$ 
6:      $\Theta' \leftarrow \Theta - \text{sign}(\Theta) \cdot c$ 
7:     return  $\Theta'$ 
8:   end if
9: end function

```

The random sample consensus (RANSAC) was included when estimating the affine transformation to reduce the impact of outlier feature matches [Fis⁺81], with a reprojection threshold of 3.0 pixels. The calculated angles were further refined using a Savitzky-Golay filter [Sav⁺64] to reduce noise while preserving the underlying rotation pattern. This cumulative method resulted in significant efficiency increases for slow, static rotations.

To address the limitations inherent in planar affine-based approaches, a method utilizing essential matrix decomposition with camera intrinsics was implemented. The TrackKeypointsV2 function (shown in 5) takes camera intrinsics matrix \mathbf{K} as an extra input parameter. This lets the system to geometrically correct 3D motion estimates instead of just 2D affine transformations. Following the established optical flow tracking pipeline, the method employs FindEssentialMat with RANSAC to estimate the essential matrix \mathbf{E} between corresponding keypoint sets, which encodes the relative camera pose between consecutive frames while accounting for the camera's intrinsic parameters. The algorithm validates

the essential matrix estimation by ensuring \mathbf{E} is not None and maintains the correct 3×3 dimensionality before proceeding to pose recovery through RecoverPose, which decomposes the essential matrix into rotation matrix \mathbf{R} and translation components using the matched keypoint correspondences and camera intrinsics.

Algorithm 5 Keypoint Tracking with Essential Matrix and Yaw Extraction

Require: Frames $\{\mathbf{f}_t\}_{t=1}^T$, keypoints $\{\mathbf{k}_t\}_{t=1}^T$, interval s , scalar multiplier α , camera intrinsics \mathbf{K}

Ensure: Angle vector $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_n]$

- 1: **function** TRACKKEYPOINTSV2($\{\mathbf{f}_t\}, \{\mathbf{k}_t\}, s, \alpha, \mathbf{K}$)
- 2: Initialize $\boldsymbol{\theta} \leftarrow []$, $\theta_{cumulative} \leftarrow 0$, $ref \leftarrow 0$
- 3: $N \leftarrow$ number of frames
- 4: **for** $i = 0$ **to** $N - s$ **step** s **do**
- 5: $\mathbf{I}_{old} \leftarrow \text{GRAY}(\mathbf{F}_{ref})$, $\mathbf{I}_{new} \leftarrow \text{GRAY}(\mathbf{F}_{i+s})$
- 6: $\mathbf{p}_0 \leftarrow \mathbf{k}_{ref}$ as coordinates
- 7: $\mathbf{p}_1 \leftarrow \text{OPTICALFLOW}(\mathbf{I}_{old}, \mathbf{I}_{new}, \mathbf{p}_0)$
- 8: $\mathbf{G}_{old} = \{p_{0j} \mid st_j = 1\}$
- 9: $\mathbf{G}_{new} = \{p_{1j} \mid st_j = 1\}$
- 10: **if** $|\mathbf{G}_{old}| > 8$ **and** $|\mathbf{G}_{new}| > 8$ **then**
- 11: $\mathbf{E} \leftarrow \text{FINDESENTIALMAT}(\mathbf{G}_{old}, \mathbf{G}_{new}, \mathbf{K}, \text{RANSAC}, 1.0)$
- 12: **if** \mathbf{E} is not None **and** \mathbf{E} is 3×3 **then**
- 13: $\mathbf{R} \leftarrow \text{RECOVERPOSE}(\mathbf{E}, \mathbf{G}_{old}, \mathbf{G}_{new}, \mathbf{K})$
- 14: $\theta_{last} \leftarrow \text{EXTRACTYAWFROMROTATION}(\mathbf{R})$
- 15: $\Theta \leftarrow \Theta + \theta_{last}$
- 16: $\Theta \leftarrow \text{CORRECTDRIFT}(\Theta)$
- 17: **end if**
- 18: **end if**
- 19: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} \cup \Theta$
- 20: $ref \leftarrow i + s$
- 21: **end for**
- 22: $\boldsymbol{\theta} \leftarrow -\boldsymbol{\theta} \cdot \alpha$
- 23: **return** $(\{\mathbf{F}_t\}, \boldsymbol{\theta})$
- 24: **end function**

The specific yaw extraction approach uses a Tait-Bryan angle decomposition, which Bryan (1911) first proposed as a way to break down Euler angles into three separate rotations [Cra09]. It is implemented using the ExtractYawFromRotation function (shown in 6), which takes the 3×3 rotation matrix \mathbf{R} and a threshold ε as input to handle potential gimbal lock conditions that can compromise angle estimation accuracy. This function employs a conditional approach where gimbal lock detection is performed by evaluating the magnitudes of specific rotation matrix elements $|R_{0,0}|$ and $|R_{1,0}|$ against the threshold ε , setting the yaw angle θ_{yaw} to zero when gimbal lock is detected to prevent numerical instabilities. For normal operating conditions, the yaw angle is computed using the arctangent function $\theta_{\text{yaw}} = \arctan 2(R_{2,0}, R_{0,0})$, which provides robust angle extraction from the rotation matrix elements, followed by conversion from radians to degrees through multiplication by $\frac{180}{\pi}$. The TrackKeypointsV2 function maintains the cumulative angle tracking and drift correction mechanisms from previous versions while scaling the final result by the multiplier α .

Algorithm 6 Extract Yaw from Rotation Matrix

Require: Rotation matrix $\mathbf{R} \in \mathbb{R}^{3 \times 3}$, threshold ε

Ensure: Yaw angle $\theta_{\text{yaw,deg}}$ (degrees)

```

1: function EXTRACTYAWFROMROTATION( $\mathbf{R}$ ,  $\varepsilon$ )
2:   if  $|R_{0,0}| < \varepsilon$  and  $|R_{1,0}| < \varepsilon$  then
3:      $\theta_{\text{yaw}} \leftarrow 0$                                       $\triangleright$  Gimbal lock case
4:   else
5:      $\theta_{\text{yaw}} \leftarrow \arctan 2(R_{2,0}, R_{0,0})$ 
6:   end if
7:    $\theta_{\text{yaw,deg}} \leftarrow \theta_{\text{yaw}} \cdot \frac{180}{\pi}$ 
8:   return  $\theta_{\text{yaw,deg}}$ 
9: end function

```

Similar to the affine approach, drift correction was applied to mitigate cumulative errors. The final smoothing was performed using the same Savitzky-Golay filter as in the affine approach. This methodology provided a more geometrically grounded estimation of yaw with improved tolerance to viewpoint changes and partial occlusions. Unlike the affine-based approach, the essential matrix method does not rely on the assumption of planar motion and can accurately represent true 3D rotation. Experimental evaluation demonstrated superior performance during in-place rotation scenarios where the environment exhibited sufficient structural features.

3.2 Sensor-Based Approach

The sensor-based approach leverages inertial measurement unit (IMU) sensors to detect and measure rotational movement with greater accuracy and reliability. This methodology was developed to overcome the inherent limitations of camera-based methods while maintaining real-time performance on consumer hardware platforms.

A custom Android application was developed to facilitate simultaneous recording of video footage and sensor data, enabling direct comparative analysis of both methodological approaches under identical testing conditions. The application was designed to capture data from multiple sensors concurrently, including the accelerometer, gyroscope, magnetometer, and GPS components. Each sensor stream was recorded at its maximum available sampling rate to ensure high fidelity data collection. Precise synchronization between sensor data and video frames was achieved through timestamp correlation, facilitating direct temporal alignment between visual information and corresponding sensor readings. This approach enabled accurate comparison of rotation estimates derived from both camera and sensor-based methodologies. All sensor data was systematically stored in CSV format with comprehensive metadata, including relative timestamps (milliseconds since recording start), absolute timestamps (system time), and estimated frame numbers based on the video frame rate. This structured formatting facilitated subsequent analysis and processing.

Gyroscope: This sensor directly measures angular velocity around the device's three axes. For yaw calculation, the z-axis measurement was primarily utilized, corresponding to rotation around the vertical axis when the device is oriented in standard position. The gyroscope demonstrated high sampling rates and excellent responsiveness to rotational movement. However, the primary limitation identified was integration drift—small measurement errors accumulate over time when integrating angular velocity to determine absolute orientation angle [Woo07].

Magnetometer: This sensor measures the Earth's magnetic field direction, providing an absolute reference for orientation relative to magnetic north. While the magnetometer does not suffer from drift, experimental testing revealed significant vulnerability to interference from nearby ferromagnetic materials and electromagnetic fields, particularly in indoor environments with electrical infrastructure [Woo07].

Accelerometer: The accelerometer measures linear acceleration forces along three axes. Through experimental evaluation, it was determined that for measuring rotation around the vertical axis (yaw), the accelerometer provides limited utility, as gravity primarily influences pitch and roll measurements but offers minimal information about yaw rotation [Woo07].

GPS: While data was collected to assess its potential for orientation detection [Kap⁺¹⁷], GPS measurements exhibited significant limitations for precise rotation calculation. Testing revealed substantial gaps in data availability even in optimal open-air environments, with position accuracy errors approximating 15 meters and inconsistent heading information. These limitations rendered GPS unsuitable for high-precision rotation calculation required for BVIP navigation assistance.

Algorithm 7 Yaw Angle Estimation from Gyroscope and Magnetometer (Complementary Filter)

Require: Gyroscope data $\{(t_j^g, g_{zj})\}$, Magnetometer data $\{(t_j^m, m_{xj}, m_{yj})\}$, filter coefficient α

Ensure: Fused yaw $\theta^{\text{fused,deg}}$

```

1: function CALCULATEYAWANGLE( $\{(t_j, g_{zj})\}$ ,  $\{(t_j, m_{xj}, m_{yj})\}$ ,  $\alpha$ )
2:    $\theta^{\text{mag}} \leftarrow \arctan 2(-m_{yj}, m_{xj})$ 
3:    $h_0 \leftarrow \theta_0^{\text{mag}}$ 
4:    $\theta^{\text{mag}} \leftarrow \text{unwrap}(\theta^{\text{mag}} - h_0)$ 
5:    $\tilde{\theta}^{\text{mag}} \leftarrow \text{Interp}(\theta^{\text{mag}}, t_j^m, t_i^g)$ 
6:    $\theta^{\text{gyro}} \leftarrow 0$ ,  $\theta^{\text{fused}} \leftarrow 0$ 
7:   for  $i = 1$  to  $N - 1$  do
8:      $\Delta t_i \leftarrow t_i^g - t_{i-1}^g$ 
9:      $\theta_i^{\text{gyro}} \leftarrow \theta_{i-1}^{\text{gyro}} + g_{zi} \cdot \Delta t_i$ 
10:     $\Delta \theta^{\text{gyro}} \leftarrow g_{zi} \cdot \Delta t_i$ 
11:     $\Delta \theta^{\text{mag}} \leftarrow \tilde{\theta}_i^{\text{mag}} - \tilde{\theta}_{i-1}^{\text{mag}}$ 
12:     $\theta_i^{\text{fused}} \leftarrow \theta_{i-1}^{\text{fused}} + \alpha \cdot \Delta \theta^{\text{gyro}} + (1 - \alpha) \cdot \Delta \theta^{\text{mag}}$ 
13:   end for
14:    $\theta^{\text{fused,deg}} \leftarrow \theta^{\text{fused}} \cdot \frac{180}{\pi}$ 
15:   return  $\theta^{\text{fused,deg}}$ 
16: end function

```

Following comprehensive evaluation of individual sensors, it was determined that fusion of gyroscope and magnetometer data would provide optimal accuracy and reliability for rotation angle estimation. The complementary filter method was selected for its computational efficiency and effectiveness in combining the complementary strengths of both sensors.

This algorithm 7 implements a sensor fusion approach that combines gyroscope and magnetometer data to estimate yaw angles through a complementary filtering technique, pro-

viding an alternative to vision-based rotational tracking methods. The `CalculateYawAngle` function processes gyroscope angular velocity measurements $\{(t_j^g, g_{zj})\}$ and magnetometer readings $\{(t_j^m, m_{xj}, m_{yj})\}$ along with a filter coefficient α to produce a fused yaw estimate $\theta^{\text{fused,deg}}$. The method begins by computing the initial magnetometer-based yaw angle θ^{mag} using the arctangent function applied to the horizontal magnetic field components $\arctan 2(-m_{yj}, m_{xj})$, followed by angle unwrapping and interpolation procedures to ensure temporal consistency and handle potential discontinuities in the magnetometer data. The algorithm initializes both gyroscope and fused angle estimates to zero before entering the main processing loop that iterates through $N - 1$ measurement intervals. At each time step i , the method calculates the time interval $\Delta t_i = t_i^g - t_{i-1}^g$ and integrates the gyroscope data to estimate rotational change through $\theta_i^{\text{gyro}} = \theta_{i-1}^{\text{gyro}} + g_{zi} \cdot \Delta t_i$, while simultaneously computing the gyroscope-based angular increment $\Delta\theta^{\text{gyro}} = g_{zi} \cdot \Delta t_i$ and the change in interpolated magnetometer-derived angles $\Delta\theta^{\text{mag}} = \tilde{\theta}_i^{\text{mag}} - \tilde{\theta}_{i-1}^{\text{mag}}$. The complementary filter then blends these estimates using the weighted combination $\theta_i^{\text{fused}} = \theta_{i-1}^{\text{fused}} + \alpha \cdot \Delta\theta^{\text{gyro}} + (1 - \alpha) \cdot \Delta\theta^{\text{mag}}$, where the coefficient α determines the relative trust placed in gyroscope versus magnetometer measurements.

The high value of α (0.98) was empirically determined to ensure the system prioritized gyroscope data for short-term changes while gradually incorporating magnetometer readings for drift correction, thereby providing both temporal responsiveness and long-term stability.

Chapter 4

Results

This chapter presents the experimental results obtained from the comparative evaluation of camera-based and sensor-based yaw rotation estimation methods across various real-world scenarios. The testing was conducted on four representative video sequences that progressively increase in complexity, from static rotation to dynamic walking with rotational components. Each sequence was analyzed using three separate methodologies: camera-based v1 (affine transformation), camera-based v2 (essential matrix decomposition), and sensor-based fusion (gyroscope and magnetometer). The rotation angles computed using each method were analyzed and visualized with respect to time, alongside keyframe visualizations showing critical scene transitions at specific time points (SF_0 through SF_6).

The evaluation methodology focused on assessing accuracy relative to ground truth rotations, temporal stability, robustness to environmental variations, and computational efficiency. Ground truth rotations were estimated through visual inspection of keyframes. Each test scenario was designed to evaluate specific challenges: drift accumulation, translation-rotation discrimination, environmental interference, and real-time performance.

4.1 Static In-Place Rotation (Left Turn)

The first test scenario involved a controlled static rotation where the subject remained stationary while performing a gradual left turn from an initial frontal orientation (SF_0) to approximately 85 degrees clockwise (SF_5), followed by a slight counterclockwise adjustment (SF_6). This scenario provided optimal conditions for camera-based methods with minimal motion blur, consistent lighting, and abundant visual features.

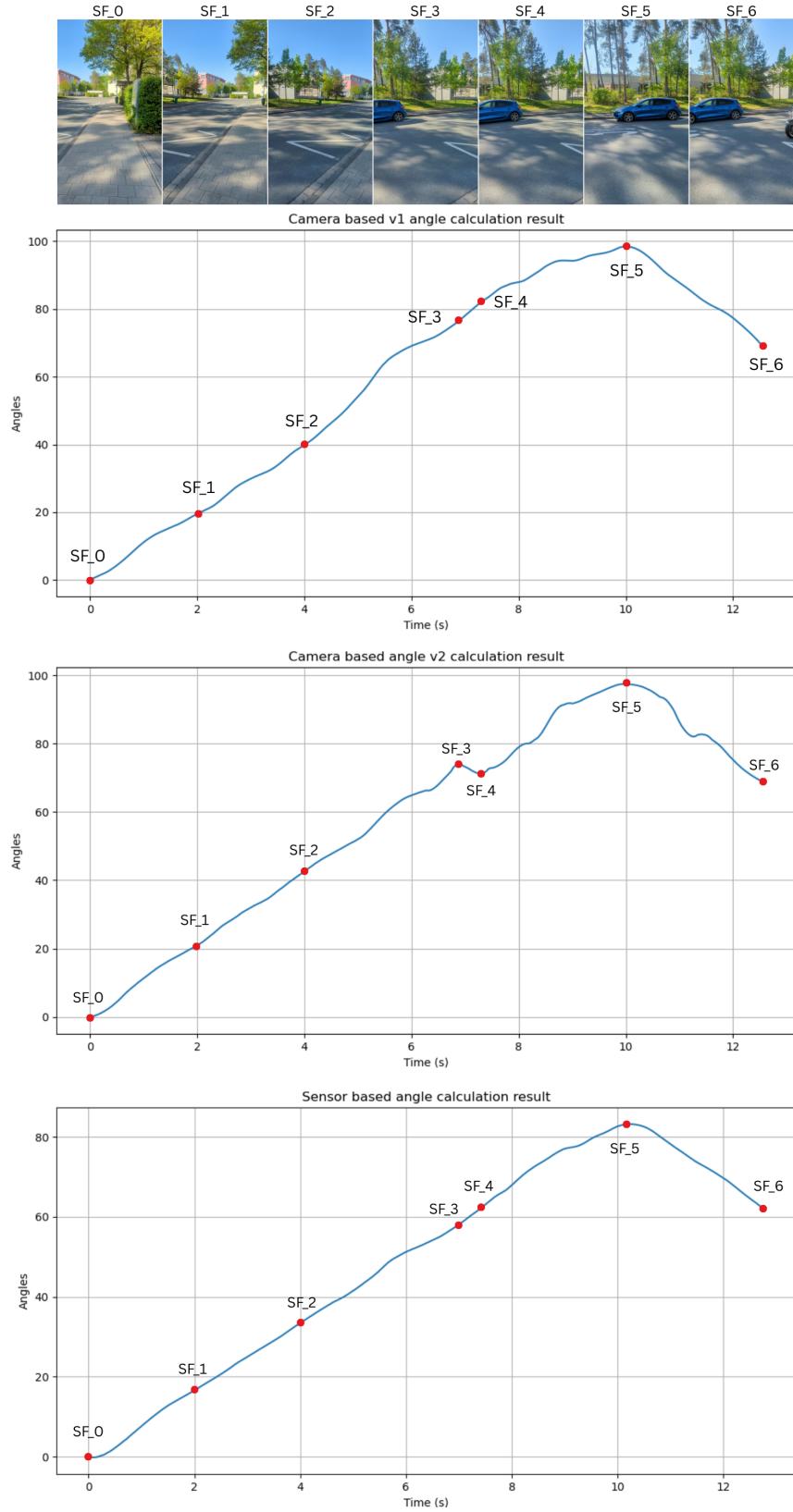


Figure 4.1

Camera-based v1 – the affine transformation approach produced a smooth and steadily increasing yaw angle trajectory. The estimated angle rose gradually from 0° at SF_0 to a peak near 100° at SF_5, followed by a slight decrease toward SF_6. This pattern accurately captured the directional change and the slight counter-rotation at the end. However, the method exhibited systematic overestimation of approximately 15-20° throughout the rotation, with the final angle at SF_5 reaching 100° compared to the expected 85°. This required calibration through the scalar multiplier parameter (set to 3.5) and demonstrated dependence on feature-rich environments.

Camera-based v2 – the essential matrix decomposition method demonstrated enhanced angular accuracy and geometric consistency. The trajectory pattern remained similar to v1 and the peak rotation at SF_5 reached approximately 100°, representing a 15-20° overestimation. A minor discontinuity was observed between SF_3 and SF_4, possibly due to transient keypoint correspondence mismatches.

Sensor-Based Fusion Performance – The IMU sensor fusion approach achieved superior performance across all evaluation metrics. The angular estimation followed a near-linear trend up to SF_5, reaching approximately 83°, which aligned closely with the ground truth rotation. The trajectory exhibited exceptional smoothness and temporal consistency without the computational and memory overhead associated with vision-based methods. Computational efficiency achieved processing times under 1ms per sample, enabling real-time performance. The method demonstrated complete immunity to visual environmental factors such as lighting variations or texture complexity that could affect camera-based approaches.

4.2 Bidirectional In-Place Rotation

The second scenario tested the methods' ability to handle bidirectional motion through a complete rotation cycle. The subject performed an approximately 90° clockwise rotation (SF_0 to SF_2) followed by a return to the original orientation (SF_2 to SF_6). This test specifically evaluated drift accumulation characteristics, directional symmetry in measurements, and baseline recovery capabilities after completing a full motion cycle.

Camera-based v1 method captured the general U-shaped trajectory characteristic of bidirectional rotation but exhibited significant magnitude underestimation throughout the sequence. The peak rotation reached only -65° instead of the expected -90°, representing a substantial 25° underestimation of the actual rotational magnitude. During the return

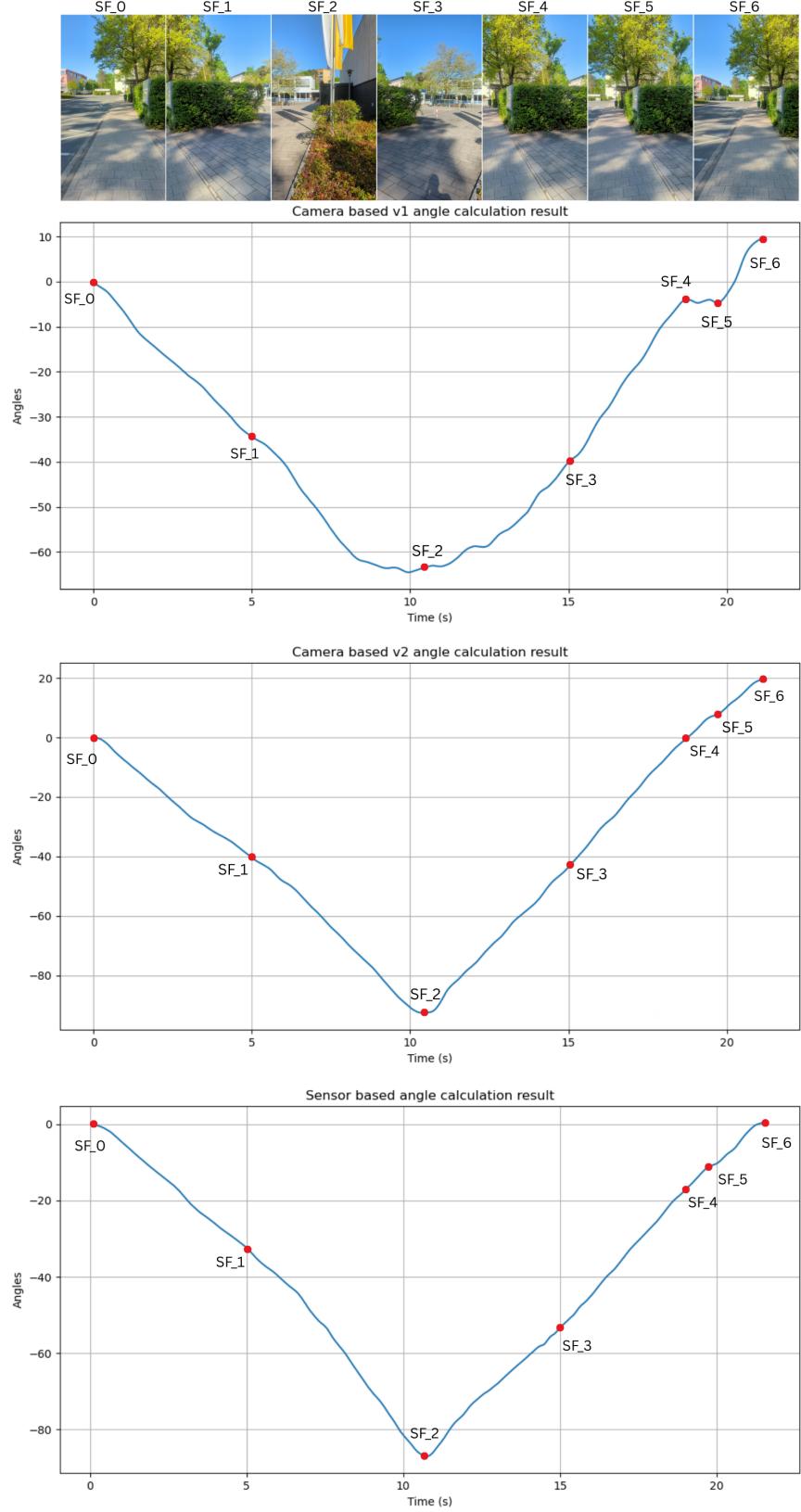


Figure 4.2

motion, the method showed incomplete baseline recovery, terminating at $+10^\circ$ rather than returning to 0° , clearly indicating cumulative error accumulation over the sequence duration.

Camera-based v2 demonstrated substantial improvement in rotational fidelity compared to the affine method. The peak rotation accurately reached -90° , precisely matching the ground truth motion as observed in the keyframe sequence. However, the return phase exhibited overestimation, ending at approximately $+20^\circ$ due to drift accumulation effects. While the method showed better geometric consistency and motion modeling, it remained susceptible to cumulative errors over extended measurement sequences.

The sensor-based fusion approach achieved the most accurate and symmetric performance characteristics. The trajectory formed a clean, symmetric U-shape with the peak rotation reaching -90° and a precise return to 0° at SF_6. The method demonstrated excellent temporal consistency without irregular oscillations or drift-related artifacts that plagued the camera-based approaches.

A critical finding emerged regarding the different drift characteristics exhibited by each method. Camera-based approaches accumulated both systematic and random errors, particularly during the return phase of bidirectional motion. The affine method showed progressive baseline shift, while the essential matrix approach exhibited directional bias in error accumulation. In contrast, the sensor fusion method maintained baseline stability through complementary filter correction mechanisms, effectively eliminating long-term drift through the combination of gyroscope responsiveness and magnetometer absolute reference.

4.3 Translation Combined with Rotation

The third scenario introduced translational motion complexity by combining forward walking with rotational components. The subject walked forward along a sidewalk (SF_0 to SF_4) with natural gait-induced oscillations of approximately $\pm 5^\circ$, then performed a 60° clockwise rotation (SF_4 to SF_5), followed by a return to forward orientation (SF_5 to SF_6). The test evaluated the fundamental challenge of discriminating between translational and rotational motion components in realistic navigation scenarios. It also revealed fundamental differences in how each method handles combined motion scenarios.

The camera-based v1 method exhibited significant confusion between translational and rotational components during the pure translation phase. The method incorrectly detected rotational motion, showing artificial rightward rotation trends despite the absence of actual yaw changes. This behavior highlighted the method's fundamental inability to

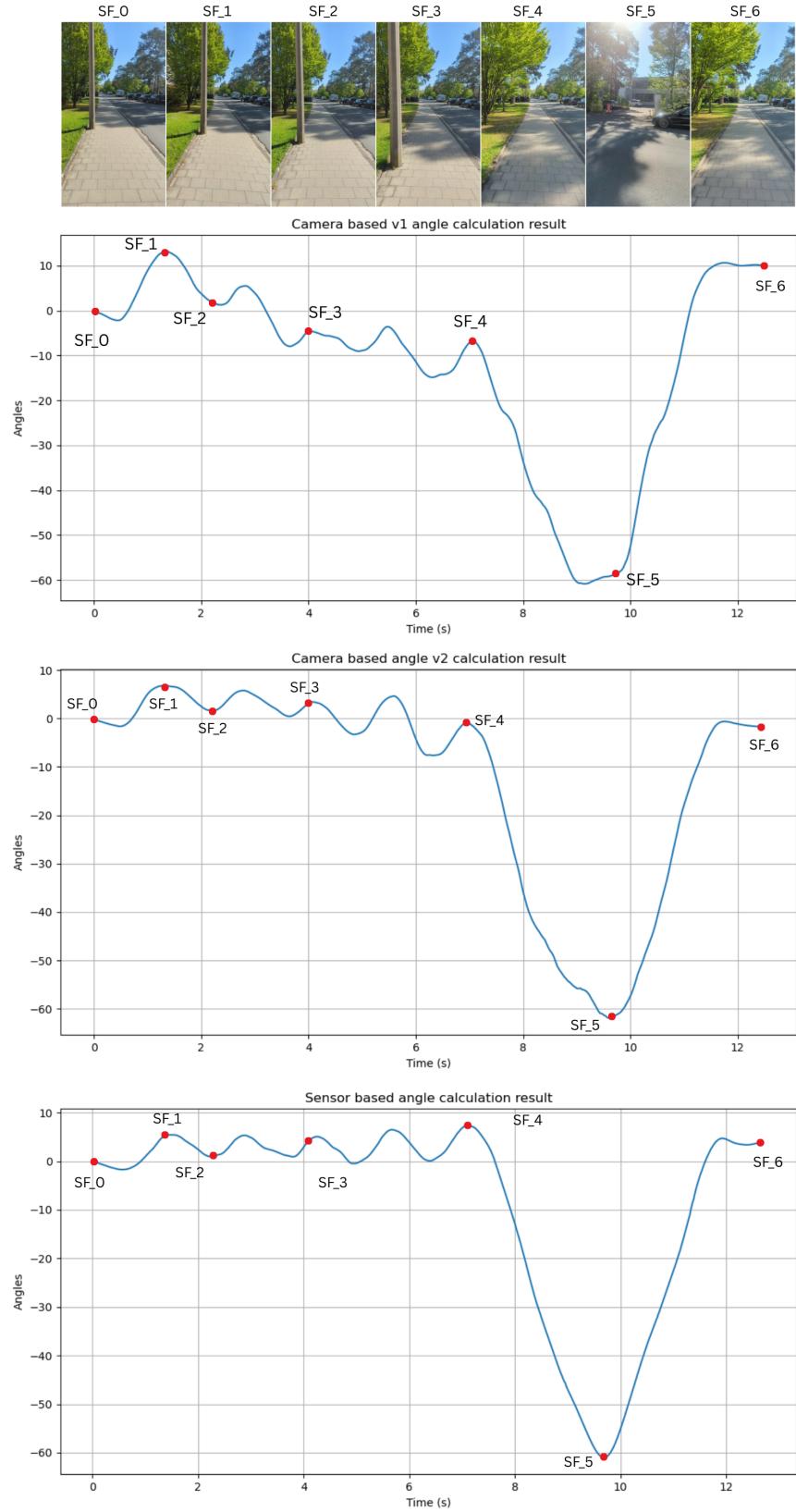


Figure 4.3

distinguish between optical flow patterns caused by forward motion versus those generated by actual rotational movement.

Camera-based v2 showed improved discrimination capabilities with $\pm 5^\circ$ oscillations correctly attributed to natural gait motion while maintaining an overall 0° rotation estimate during the translation phase. The essential matrix approach better isolated rotational components from translational motion through its more sophisticated geometric motion model, which accounts for the 3D nature of camera movement rather than assuming planar motion.

The sensor-based fusion method demonstrated superior motion discrimination by correctly maintaining $\pm 5^\circ$ oscillations around 0° during pure translation phases. The method successfully isolated yaw rotation from translational acceleration components, providing clean separation of motion types. This discrimination capability stems from the fundamental difference in sensor physics: gyroscopes directly measure angular velocity around specific axes, while magnetometers provide absolute orientation reference. The complementary filter effectively combines these measurements to isolate rotational motion from translational movement, as walking-induced accelerations do not affect the yaw angle calculation derived from these sensors.

The outdoor testing environment introduced additional challenges, including variable lighting conditions with patches of shadow and direct sunlight, the textural complexity of the paving stones and vegetation, and dynamic elements such as moving shadows. The results demonstrated that sensor-based methods remained completely unaffected by these visual variations, maintaining consistent performance regardless of environmental conditions. Camera-based approaches showed increased measurement noise and occasional tracking failures in challenging lighting transitions, particularly when moving between shadowed and illuminated areas.

4.4 Complex Diagonal Motion with Rotation

The fourth scenario represented the most challenging test case, involving diagonal walking patterns followed by in-place rotation. The subject initially walked diagonally left, then shifted to diagonal rightward walking without rotation, followed by an in-place clockwise yaw rotation of approximately -60 degrees, concluding with a counter-clockwise return to the original orientation. This sequence tested the methods' resilience under

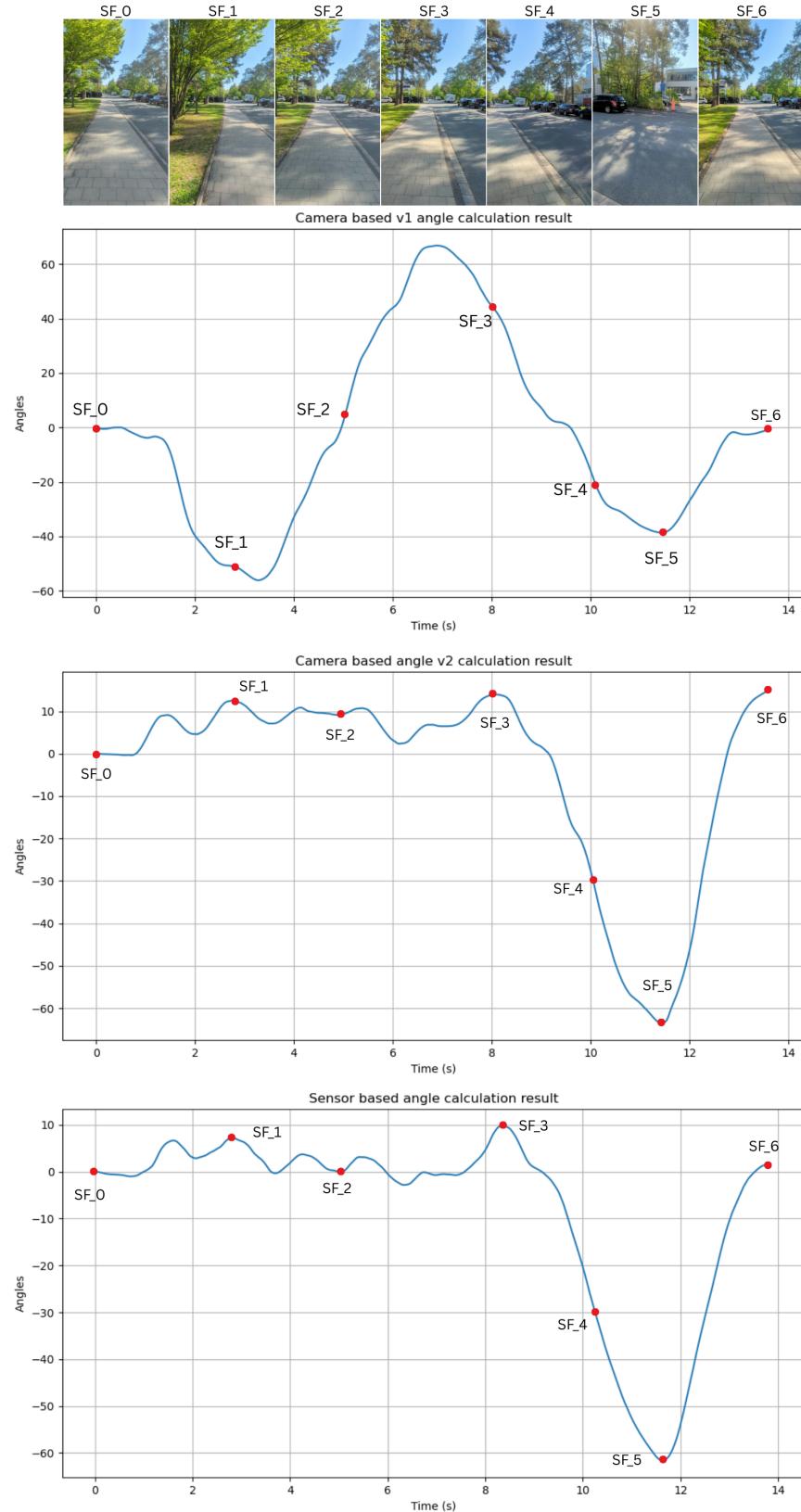


Figure 4.4

naturalistic navigation conditions with complex motion combinations that commonly occur in real-world navigation scenarios.

Camera-based v1 method produced highly erratic and unreliable estimates under these complex motion conditions. The method misinterpreted diagonal translation as significant rotation, showing a false negative excursion to -60° during the initial translation phase when no actual rotation occurred. Transitions between SF_2 and SF_3 indicated fundamental tracking failures under complex motion patterns. The method's reliance on planar motion assumptions broke down completely when confronted with the perspective effects and feature tracking challenges introduced by diagonal movement patterns.

Camera-based v2 maintained substantially greater stability and interpretability despite the complex motion scenario. The diagonal walking phases (SF_0 to SF_3) correctly showed minimal rotation with ±10° oscillations around the baseline, accurately reflecting the natural variations in walking motion without interpreting translation as rotation. The actual rotational component was accurately detected and measured, reaching -65° at SF_5 with smooth, realistic transitions that corresponded to the observed motion in the keyframe sequence.

Sensor-based fusion method delivered consistent and accurate performance throughout the entire complex sequence. The method successfully discriminated between all motion components, maintaining baseline stability during diagonal translation phases and accurately measuring the rotational component with smooth, realistic transitions. The fusion approach showed no degradation in performance despite the increased complexity of the motion pattern, demonstrating robust operation under challenging real-world navigation scenarios.

4.5 Performance

Accuracy analysis revealed systematic differences across scenarios. Camera v1 errors ranged from ±15° (static) to >40° (complex motion), while camera v2 achieved ±10-15° accuracy with less degradation under complexity. Sensor fusion maintained ±2-3° precision across all scenarios.

Computational requirements differed substantially. Camera methods required 150-300ms processing per frame with 500-600MB memory usage, precluding real-time mobile deployment. Sensor fusion achieved <1ms processing with <10MB memory usage.

Environmental robustness testing showed camera methods degraded 15-25% under lighting variations and failed in feature-poor environments. Sensor fusion remained unaffected by visual conditions.

4.6 Discussion

Camera-based approaches demonstrated reasonable accuracy under controlled conditions. However they faced critical limitations for practical deployment. Feature dependency, computational demands, and motion discrimination challenges make them unsuitable for robust BVIP assistance. The affine method's planar assumptions break down under complex motion, while essential matrix decomposition, though geometrically superior, remains computationally prohibitive.

Sensor fusion emerged as the optimal solution, which provides consistent accuracy independent of environmental conditions while meeting mobile device constraints. The complementary filter effectively combines gyroscope responsiveness with magnetometer absolute reference, eliminating drift while maintaining real-time performance. These results indicate that IMU-based approaches offer superior practical viability for navigation assistance systems, providing the reliability and efficiency required for effective BVIP support in diverse real-world conditions.

Chapter 5

Future Directions and Conclusion

5.1 Future Directions

This chapter outlines several potential enhancements and future research directions identified through the course of this study. These avenues aim to address existing limitations and expand upon the promising findings of this project, thereby enhancing the practical applicability and robustness of yaw rotation estimation methods for visually impaired navigation.

The current implementation uses a complementary filter approach combining gyroscope and magnetometer data effectively. But more advanced sensor fusion algorithms like the Extended Kalman Filter (EKF) or the Unscented Kalman Filter (UKF) could be looked into in the future. These advanced filtering techniques can model sensor noise more accurately and dynamically adapt to changing environmental conditions, potentially reducing drift and improving robustness, especially in magnetically noisy environments.

To enable wider practical application and integration into wearable assistive devices, the computational efficiency of both camera-based and sensor-based methods should be optimized. Future work could explore optimizing software algorithms for mobile CPUs or they can use lightweight machine learning models for efficient keypoint detection and tracking could significantly enhance real-time capabilities.

Although the current study found sensor-based methods superior in terms of robustness and computational efficiency, there remains untapped potential in hybrid approaches that leverage the strengths of both camera-based and sensor-based systems. Future studies can use adaptive weighting algorithms that can change how camera and sensor data are combined based on the surroundings and the sensor's level of trust.

These outlined future directions would further enhance the accuracy, robustness, usability, and overall effectiveness of rotation angle estimation technologies, thereby contributing significantly to improving spatial orientation support for visually impaired individuals.

5.2 Conclusion

This project has looked at and compared camera-based and sensor-based methods for determining yaw rotation angles in a systematic way to help BVIPs with their spatial orientation. Thorough evaluations performed across various real-world scenarios demonstrated distinct advantages and drawbacks associated with each approach.

At first, the camera-based methods that used optical flow techniques and essential matrix decomposition looked promising since they could directly understand visual scenes. However, these methods exhibited significant drawbacks, including high computational complexity, large memory requirements, and sensitivity to environmental factors such as lighting conditions and texture availability. Additionally, the accumulation of errors over time, especially noticeable in complex motion scenarios, considerably limits their practical deployment in real-time mobile navigation systems.

The sensor-based solution that combined gyroscope and magnetometer data with a complementary filter, on the other hand, was much better in all test scenarios. It provided accurate, consistent, and real-time rotational angle measurements without the computational overhead associated with vision-based methods. Its natural ability to handle changes in the environment, together with its low drift and great accuracy, made it clear that it would be useful for navigation assistance for BVIP.

Based on these results, the sensor-based fusion approach is definitely the best and most reliable way to estimate the yaw rotation angle in assistive technologies. Future enhancements might include integrating additional sensor inputs or advanced filtering techniques to further improve accuracy and robustness. The outcomes of this research offer valuable contributions toward developing more dependable and effective navigation systems, significantly enhancing spatial orientation and mobility independence for visually impaired users in diverse real-world environments.

List of Abbreviations

SVM Support Vector Machine

List of Figures

4.1	16
4.2	18
4.3	20
4.4	22

List of Tables

Bibliography

- [Bac⁺01] E. R. Bachmann, R. B. McGhee, X. Yun, and M. J. Zyda. Inertial and magnetic posture tracking for inserting humans into networked virtual environments. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 9–16, 2001 (cited on p. 4).
- [Bay⁺08] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008 (cited on p. 3).
- [Bea⁺95] S. S. Beauchemin and J. L. Barron. The computation of optical flow. *ACM computing surveys (CSUR)*, 27(3):433–466, 1995 (cited on p. 3).
- [Cra09] J. J. Craig. *Introduction to robotics: mechanics and control, 3/E*. Pearson Education India, 2009 (cited on p. 11).
- [Fis⁺81] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981 (cited on p. 9).
- [Kap⁺17] E. D. Kaplan and C. Hegarty. *Understanding GPS/GNSS: principles and applications*. Artech house, 2017 (cited on p. 13).
- [Kun20] I. Kunyankin. Camera motion estimation using optical flow, July 2020. URL: <https://medium.com/@ikunyankin/camera-motion-estimation-using-optical-flow-ce441d7ffec> (cited on pp. 1, 7).
- [Low04] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004 (cited on p. 3).
- [Luc⁺81] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI’81: 7th international joint conference on Artificial intelligence*, volume 2, pages 674–679, 1981 (cited on pp. 3, 6).

- [Sab06] A. M. Sabatini. Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing. *IEEE transactions on Biomedical Engineering*, 53(7):1346–1356, 2006 (cited on p. 4).
- [Sav⁺64] A. Savitzky and M. J. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964 (cited on p. 9).
- [Tan⁺15] Z. Tang, M. Sekine, T. Tamura, N. Tanaka, M. Yoshida, and W. Chen. Measurement and estimation of 3d orientation using magnetic and inertial sensors. *Advanced Biomedical Engineering*, 4:135–143, 2015 (cited on p. 4).
- [Tap⁺17] R. Tapu, B. Mocanu, and T. Zaharia. A computer vision-based perception system for visually impaired. *Multimedia Tools and Applications*, 76:11771–11807, 2017 (cited on p. 1).
- [Woo07] O. J. Woodman. An introduction to inertial navigation. Technical report, University of Cambridge, Computer Laboratory, 2007 (cited on pp. 4, 12).