University of Pannonia
Faculty of Information Technology
Doctoral School of Information Science and Technology

METHODS FOR VISIBLE LIGHT-BASED INDOOR POSITIONING USING ANGLE DIFFERENCE MEASUREMENTS AND LED LIGHTING INFRASTRUCTURE

DOI:10.18136/PE.2024.895

PhD Dissertation
by
Márk Rátosi

Head of Doctoral School
Dr. Ferenc Hartung, DSc

Supervisors
Dr. Gyula Simon, DSc
Dr. Attila Fodor, PhD

Veszprém
Hungary
2024

Methods for visible light-based indoor positioning using angle difference measurements and LED lighting infrastructure

Thesis for obtaining a PhD degree in the Doctoral School of Information Science and Technology of the University of Pannonia

in the research field of Computer Sciences

Written by Márk Rátosi

Supervisors: Dr. Gyula Simon, DSc; Dr. Attila Fodor, PhD

propose acceptance (yes / no)

…………………………
Dr. Gyula Simon, DSc

propose acceptance (yes / no)

…………………………
Dr. Attila Fodor, PhD

As reviewer, I propose acceptance of the thesis:

Name of Reviewer: …....................... …................. yes / no

…………………………
signature

Name of Reviewer: …....................... …................. yes / no

…………………………
signature

The PhD-candidate has achieved …..........% at the public discussion.

Veszprém,

…………………………
Chairman of the Committee

The grade of the PhD Diploma …...................................... (…….. %)
Veszprém,

…………………………
Chairman of UDHC

Acknowledgement

First and foremost, I would like to thank Dr. Gergely Vakulya and Dr. Gergely Zachár for fostering my interest and enthusiasm in technical sciences throughout my university years with various technological curiosities, problems, and their solutions. I am grateful for the support from my university colleagues and the pleasant atmosphere that made my everyday experiences enjoyable.

I express my gratitude for the dedicated work of my supervisor, Dr. Gyula Simon, and the time spent together in contemplation and work. His professional and personal guidance has been invaluable, and I have learned a great deal from him. Additionally, I extend my thanks to Dr. Attila Fodor, my other supervisor, for the abundant support during my university studies.

I thank Szabolcs Szekér and Dávid Noel Tóth, my "comrades" and friends, for standing by me since the beginning of university, providing constant support.

I owe immense thanks to my parents and my sibling for the warm home atmosphere, their support, and love, helping me overcome difficult days. Special thanks to my wife, Noémi, for the tremendous support, encouragement, and prodding, which significantly contributed to the completion of this dissertation.

Without them, undoubtedly, I would not have reached this point.

Köszönetnyilvánítás

Először is szeretném megköszönni dr. Vakulya Gergelynek és dr. Zachár Gergelynek, hogy az egyetemi évek alatt tovább növelték a műszaki tudományok iránti érdeklődésem és lelkesedésem különféle technológiai érdekességekkel, problémákkal és azok megoldásaival. Köszönöm az egyetemi kollégáknak a sok segítséget; és a kellemes légkört, melynek köszönhetően mindig jól teltek a hétköznapok.

Köszönöm dr. Simon Gyula témavezetőmnek kitartó és odaadó munkásságát, a sok elmélkedéssel és munkával együtt töltött időt, szakmai és emberi iránymutatását, melyből rengeteget tanulhattam. Továbbá köszönöm dr. Fodor Attila témavezetőmnek a sok egyetemi segítséget.

Köszönöm Szekér Szabolcs és Tóth Dávid Noel „bajtársaimnak” és barátaimnak, hogy az egyetemi évek elejétől fogva mindvégig mellettem voltak és bármikor számíthattam rájuk.

Hálával tartozom szüleimnek és testvéremnek a meghitt otthoni légkörért, támogatásukért és szeretetükért, hogy mindig segítettek túljutni a nehezebb napokon. Köszönöm feleségemnek, Noéminek a rengeteg támogatást, biztatást és noszogatást, mellyel hozzájárult eme disszertáció elkészültéhez.

Nélkülük minden bizonnyal nem jutottam volna el idáig.

Abstract

Object positioning is becoming more and more important, and thus, nowadays is integrated in our everyday life. Be it the positioning of a mobile phone, a passenger vehicle, or any autonomous transportation vehicle in a warehouse, the demand after new and more reliable positioning methods and technologies is high. While the outdoor case of object positioning is dominated by the Global Navigation Satellite Systems, indoor positioning is still under heavy research with various techniques competing with each other trying to provide the best solution for the different use cases and their requirements.

Indoor positioning can be achieved with numerous technologies, e.g., using radio signals, ultrasound, RFID tags, graphical markers, fingerprinting, and optical solutions. The thesis provides contributions to visible light-based indoor positioning systems utilizing cameras as to be positioned objects, and LED light fixtures as reference points, which are pre-installed in known positions and can also be part of the environment's lighting infrastructure. The reference points (LED beacons) are identified by utilizing a Visible Light Communication protocol. The thesis introduces a possible implementation of such an indoor positioning system on which the results are based.

One of the main requirements for the LED beaconing in the positioning system is that the LED's modulation should not be disturbing for the human eye. The other important requirement is that the camera should not be required to operate in a synchronized manner with the LED beacons. These requirements inevitably cause undersampling for the communication, since an ordinary camera operates in the 30-120 Hz region, and a flicker free modulation requires a much greater frequency, e.g., more than 1000 Hz. The detailed error analysis of two VLC protocols found in the literature - namely Undersampled Frequency-Shift On-Off Keying, and Undersampled Phase-Shift On-Off Keying - are provided in the thesis. The effect of various error sources - e.g., thresholding error, saturation, noise, jitter, and frequency error - on the performance properties of the protocols are analyzed.

A key feature for robust, real-time operation for such positioning systems is the trackability of the LED beacons on the camera's image stream. A steadily trackable beaconing solution is presented in the thesis, which builds on the basics of the UPSOOK protocol. It incorporates two LEDs, a circular and a ring shaped one, of which at least one is always active. Another important feature is the identifiability and availability of the beacons, but due to the camera is unsynchronized to the LED beacons reliable operation is not possible with the aforementioned protocols from the literature. A novel, robust VLC protocol is presented in the thesis which can be used to stream a beacon's ID periodically, and provide error-free operation in a wide range of parameters. The protocol uses equivalent sampling, and is also based on the UPSOOK protocol.

The thesis also presents two positioning algorithms, which uses the measured angle differences between detected LED beacons pairs to estimate the object's (camera's) position and orientation. The first method utilizes a heuristic, geometric approach using inscribed angles, and two threshold parameters for outlier filtering. The second method uses a random sample consensus-based method, where the position estimate is further refined with the Nelder-Mead method using a consistent set of beacons.

Kivonat

A különféle objektumok helymeghatározása egyre fontosabb szerepet játszik a mindennapokban, melynek köszönhetően mára már szinte teljesen integrálódott az életünkbe. Legyen szó akár egy mobiltelefon, személygépjármű vagy egy raktárban működő autonóm szállítójármű lokalizációjáról, nagy az igény az újabb, megbízhatóbb helymeghatározási módszerek és technológiák iránt. Míg az objektumok helymeghatározását kültéri esetben a globális helymeghatározó műholdrendszerek dominálják, a beltéri helymeghatározás mind a mai napig erősen kutatott tématerület egymással versengő módszerekkel, melyek mindegyike a legjobb megoldást próbálja kínálni a különböző felhasználási területek számára.

Beltéri helymeghatározás számos technológiával megvalósítható, például rádióhullámok, ultrahang, RFID címkék, grafikus markerek, lenyomat-alapú módszerek, vagy optikai megoldások segítségével. Az értekezés olyan látható fény-alapú beltéri lokalizációs rendszerekhez tartalmaz új eredményeket, amelyeknél kamerák helymeghatározása történik ismert pozíciókban előre elhelyezett LED lámpák segítségével, amelyek referencia pontokként kerülnek felhasználásra, és akár a helyszín megvilágítási infrastruktúrájának részét is képezhetik. A referencia pontok (LED jeladók) látható fény-alapú kommunikációs protokoll használatával kerülnek azonosításra. A dolgozat bemutatja az említett beltéri helymeghatározó rendszer egy lehetséges megvalósítását, amelyen az elért eredmények is alapulnak.

Az egyik fő követelmény a helymeghatározó rendszerben használt LED jeladók számára, hogy a LED-ek modulációja ne legyen zavaró az emberi szem számára. A másik fontos követelmény, hogy ne kelljen a kamerát és a LED jeladókat szinkronizáltan üzemeltetni. Ezek a követelmények elkerülhetetlenül alulmintavételezést fognak okozni a kommunikáció során, hiszen egy átlagos kamera a 30-120 Hz-es tartományban üzemel, viszont a villódzásmentes modulációhoz jóval nagyobb frekvencia szükséges, akár több mint 1000 Hz. A dolgozat tartalmazza kettő, a szakirodalomban fellelhető - név szerint, Undersampled Frequency-Shift On-Off Keying, and Undersampled Phase-Shift On-Off Keying - protokoll részletes hibaanalízisét. Bemutatásra kerül, hogy a különböző hibatényezők – például a küszöbérték szuboptimális megválasztása, a szenzor szaturációja, mérési zaj, jitter, frekvencia hiba - milyen hatást gyakorolhatnak a protokollok teljesítményére.

Az efféle helymeghatározó rendszerek robusztus, valós-idejű működésének egy kulcsfontosságú tényezője, hogy a LED jeladók nyomon követhetőek legyenek a kamera által szolgáltatott képsorozatban. A dolgozat bemutat egy stabilan nyomon követhető jeladózási megoldást, amely az UPSOOK protokoll alapjaira épít. Két LED-et használ, egy kör és egy gyűrű alakút, amelyek közül legalább egy állandóan aktív. Egy másik fontos tényező a jeladók azonosíthatósága és rendelkezésre állása, ám köszönhetően annak, hogy a kamera nincs szinkronizálva a LED jeladókhoz, lehetetlen biztosítani a már említett kettő, szakirodalomban fellelhető protokoll megbízható működését. A dolgozatban egy újszerű, robusztus VLC protokollt mutatok be, melynek segítségével a jeladók azonosítókódja periodikusan ismétlődve átküldhető, továbbá a különböző paraméterek egy széles tartományán belül képes hibamentesen üzemelni. A protokoll ekvivalens mintavételezést használ és szintén az UPSOOK megoldásra épül.

A dolgozatban bemutatok továbbá kettő helymeghatározási algoritmust, amelyek detektált jeladó-párok közt mért szögkülönbségek felhasználásával becslik a lokalizálandó objektum (kamera) pozícióját és orientációját. Az első módszer egy heurisztikus, geometriai-alapú megközelítés, amely látószögek felhasználásával működik, és két küszöbérték paramétert használ a kiugró mérések kiszűréséhez. A második módszer az iteratív robusztus becslési eljáráson alapul, ahol a pozíció becslő további finomításra kerül a Nelder-Mead módszerrel a jeladók egy konzisztensnek ítélt halmazának felhasználásával.

Abstrakt

Die Lokalisierung verschiedener Objekte spielt eine zunehmend wichtige Rolle in unserem täglichen Leben und ist zu einem integralen Bestandteil unseres Daseins geworden. Ob es sich um die Lokalisierung eines Mobiltelefons, eines Personenkraftwagens oder eines autonomen Transportfahrzeugs in einem Lager handelt, es besteht eine wachsende Nachfrage nach neuen und zuverlässigeren Methoden und Technologien zur Bestimmung der Position von Objekten. Während die Lokalisierung von Objekten im Freien von globalen Satellitenpositionierungssystemen dominiert wird, bleibt die Lokalisierung in Innenräumen ein stark erforschtes Gebiet mit konkurrierenden Methoden, die jeweils die beste Lösung für verschiedene Anwendungsbereiche bieten möchten.

Die Innenraumlokalisierung kann mit verschiedenen Technologien realisiert werden, wie beispielsweise Funkwellen, Ultraschall, RFID-Tags, grafische Marker, fingerabdruckbasierte Methoden oder optische Lösungen. Diese Arbeit präsentiert neue Ergebnisse für sichtlichtbasierte Innenraumlokalisierungssysteme, bei denen die Lokalisierung von Kameras mithilfe von LED-Lampen in bekannten Positionen als Referenzpunkte erfolgt. Diese Referenzpunkte (LED-Sender) werden mithilfe eines sichtlichtbasierten Kommunikationsprotokolls identifiziert. Die Arbeit zeigt eine mögliche Umsetzung des genannten Innenraumlokalisierungssystems auf der Grundlage der erzielten Ergebnisse.

Eine der Hauptanforderungen für die in dem Lokalisierungssystem verwendeten LED-Sender ist, dass die Modulation der LEDs nicht störend für das menschliche Auge sein darf. Eine weitere wichtige Anforderung ist, dass die Kamera und die LED-Sender nicht synchron betrieben werden müssen. Diese Anforderungen führen zwangsläufig zu einer Untersampelung in der Kommunikation, da eine durchschnittliche Kamera im Bereich von 30-120 Hz arbeitet, während eine flimmerfreie Modulation eine wesentlich höhere Frequenz erfordert, die möglicherweise über 1000 Hz liegt. Die Arbeit enthält eine ausführliche Fehleranalyse von zwei Protokollen aus der Literatur, nämlich Undersampled Frequency-Shift On-Off Keying und Undersampled Phase-Shift On-Off Keying, und untersucht, wie verschiedene Fehlerfaktoren wie unsachgemäße Schwellenwerteinstellung, Sensorübersättigung, Messrauschen, Jitter und Frequenzfehler die Leistung dieser Protokolle beeinflussen können.

Ein Schlüsselfaktor für den robusten Echtzeitbetrieb solcher Lokalisierungssysteme ist die Rückverfolgbarkeit der LED-Sender in der von der Kamera bereitgestellten Bildfolge. Die Arbeit stellt eine stabile rückverfolgbare Signalisierungslösung auf der Grundlage des UPSOOK-Protokolls vor. Es werden zwei LEDs verwendet, eine kreisförmige und eine ringförmige, von denen mindestens eine ständig aktiv ist. Ein weiterer wichtiger Faktor ist die Identifizierbarkeit und Verfügbarkeit der Sender. Aufgrund des Mangels an Synchronisation zwischen Kamera und LED-Sendern ist es jedoch unmöglich, den zuvor erwähnten beiden Protokollen einen zuverlässigen Betrieb zu gewährleisten. In der Arbeit wird ein neues robustes Visible Light Communication (VLC)-Protokoll vorgestellt, das die periodische Übertragung von Senderidentifikationscodes ermöglicht und einen fehlerfreien Betrieb innerhalb eines breiten Parameterspektrums sicherstellt. Das Protokoll verwendet die äquivalente Abtastung und basiert ebenfalls auf der UPSOOK-Lösung.

Die Arbeit stellt auch zwei Lokalisierungsalgorithmen vor, die die Position und Ausrichtung des zu lokalisierenden Objekts (der Kamera) anhand der gemessenen Winkelunterschiede zwischen erkannten Senderpaaren schätzen. Die erste Methode ist ein heuristischer, geometriebasierter Ansatz, der Sichtwinkel verwendet und zwei Schwellenwertparameter zur Filterung von Ausreißern einsetzt. Die zweite Methode basiert auf einem iterativen robusten Schätzverfahren, bei dem der Positionsschätzer mithilfe der Nelder-Mead-Methode und einer konsistenten Gruppe von Sendern weiter verfeinert wird.

INDEX

1 Introduction

This section provides a brief overview of the current, widely used positioning principles (Section 1.1) and their applications (Section 1.2). Section 1.3 presents the problem statements and my research goals. The localization system's architecture, on which the various methods and algorithms of this thesis is based, is presented in Section 1.4.

1.1 Object positioning and tracking principles

In the field of object positioning several basic principles are used, where an object's position is calculated/estimated using measured quantities in the environment of the object. These methods can be categorized into the following main groups: geometrical methods (ToA, TDoA, AoA, ADoA), dead reckoning, and fingerprinting methods.

A geometrical method uses measured quantities of angles and distances from which the location estimate is calculated. These methods require reference anchor points. The position of anchor points is either calculated, estimated on the go, or measured at the installation or calibration phase of a positioning system.

Dead reckoning only uses measured quantities which represent object state differences (e.g., position differences) that are accumulated over time. In this case, for example, only the relative movement path can be estimated based on an initial state. The absolute position estimate can only be obtained if the initial state of the target object is known.

Fingerprinting methods compare various measured quantities (e.g., magnetic field, Wi-Fi RSSI) to previously measured value maps created in the object's environment at the installation or calibration phase of such systems.

These positioning principles will now be introduced with examples in 2-dimensional space, and without measurement noise for the sake of simplicity. Note that they can be trivially extended to the more general 3-dimensional case.

1.1.1 ToA/TDoA

Time of Arrival (ToA, also known as Time of Flight or ToF) approach utilizes range-based measurements between the target object and different anchor points in its environment. The distance is measured with some sort of signal's propagation: the distance between the target object and a reference point is proportional with the measured time required for that signal to travel through a medium, hence the name of this approach. When multiple reference points are available, the target object's position can be narrowed down to intersections of basic geometrical entities (e.g., hyperbolas or circles in a 2-dimensional case) constructed from the measurements and known location of the anchors. The principle is illustrated in Figure 1.1.

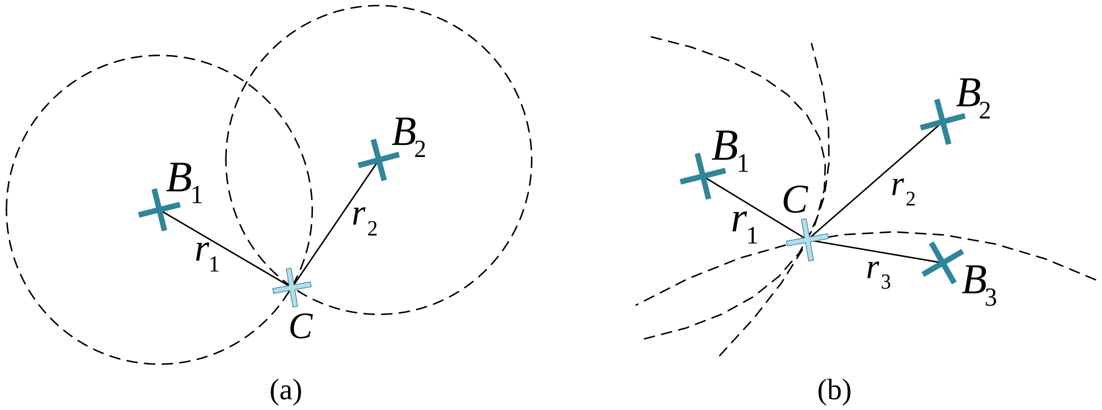


Figure 1.1 Range-based location estimation of object C using anchor points B_k with: (a) Time of Flight (ToA) approach utilizing measured ranges r_k, and (b) Time Difference of Arrival (TDoA) approach utilizing the measured range differences.

If we can measure the exact distance (or signal propagation time) between the target object and the anchor points, the target object is located at the intersection point of circles. These circles are located around the reference points and their radii are equal to the measured distances, as seen in Figure 1.1 (a). [1-5]

If we cannot measure the exact distances between the reference points and the target object, but only their differences, we talk about Time Difference of Arrival (TDOA). In this case the target object will be located at the intersection point of hyperbolas, as seen in Figure 1.1 (b). [6-10]

1.1.2 AoA/ADoA

Angle of Arrival (AoA) is an approach that utilizes angles as measured quantities. In most cases the angle between a global reference direction and the target object is measured at several reference points from which the target object's position can be trivially determined as the intersection point of vectors, as can be seen in Figure 1.2 (a). [11-16]

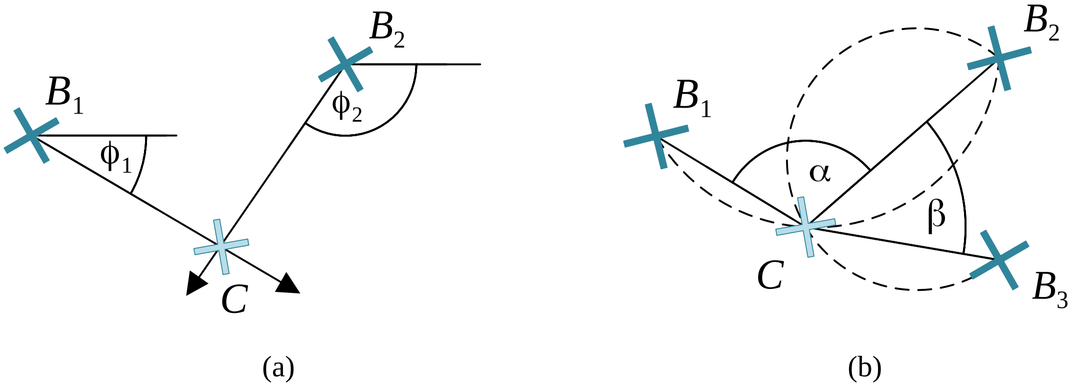


Figure 1.2 Angle-based location estimation of object C using anchor points B_k with: (a) Angle of Arrival (AoA) approach utilizing measured angles ϕ_1, ϕ_2 based on a reference direction, and (b) Angle Difference of Arrival (ADoA) utilizing measured angle differences α, β from object C's point of view between reference points $[B_1;B_2]$ and $[B_2;B_3]$, respectively.

If no global reference direction is available for the angle measurements, the difference of the angles can still be used to determine the target object's position. This approach is called Angle Difference of Arrival (ADoA). In this case the measured angle differences are the inscribed angles in circles: the angle between two reference points from the target object's view, as can be seen in Figure 1.2 (b). When an adequate number of measurements are available, the position can be estimated as the intersection point of such circles, constructed from the angle difference measurements and the reference point coordinates. [11,17-20]

1.1.3 Dead reckoning

Dead reckoning in itself is a relative positioning approach, where consecutive movement parameters (distance, direction) are measured or estimated (e.g., using rotary encoders, accelerometers or gyroscopes), and are integrated together to obtain the current position of the tracked object with respect to its previous position, as seen in Figure 1.3. The obvious downsides of this approach are that it also accumulates the measurement errors and requires correction from time to time, and that the position estimate itself is relative to the initial position of the system. Dead reckoning is typically used in combination with other approaches to minimize or eliminate these problems. [21-25]

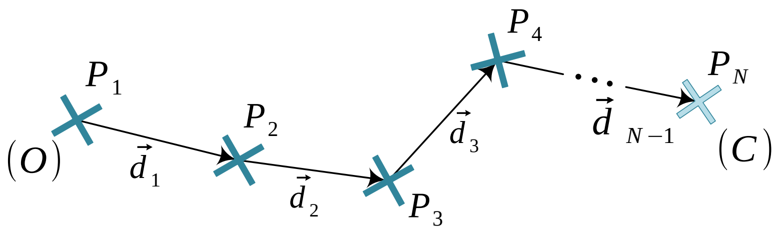


Figure 1.3 Dead reckoning localization of object C in estimated relative position P_N with respect to an initial position P_1 using consecutive displacement estimations d_k. If the exact initial position value O for P_1 is known, then the target object's relative position estimate P_N can be converted to an absolute position estimate C.

1.1.4 Fingerprinting

The fingerprinting approach requires a previously measured reference value map, where the measured quantities (fingerprints) are collected in several well-known coordinates, usually based on a grid. During localization the currently measured quantities are compared to the reference map and the position is estimated with the point that has the highest correlation between the measured and reference values, as seen in Figure 1.4. This approach typically uses multiple measurable quantities to further improve positioning accuracy, moreover, using a single quantity may result in ambiguous position estimates (due to multiple local optimums). The most commonly used quantities for fingerprinting are received signal strength and magnetic field. [26-30]

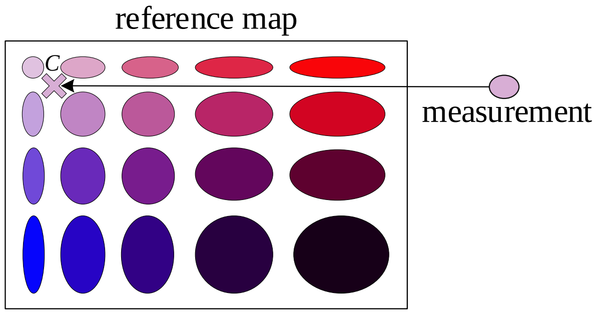


Figure 1.4 Fingerprinting approach for location estimation utilizing a reference map to which the actual measurement is compared. The measurement is taken at the target object's current position. The position estimate is located at point C, where the correlation is the highest.

1.2 Object positioning technologies

While Global Navigation Satellite Systems (GNSS) dominate in outdoor positioning, at certain conditions (e.g., at the base of tall buildings, underground or indoors) accurate position estimates cannot be provided or even the technology itself may fail due to reflecting, fading, multipath effect, deep shadowing effect, and interferences. It may also happen that certain applications simply require more precision that GPS can provide real-time in a specific environment. Several methods have been proposed in the recent decades to provide alternative methods for indoor positioning. This section provides a brief overview of these approaches and shows how the positioning principles are used with different technologies.

Radio signal-based localization [31] is among the most common approaches of indoor positioning. Several radio technologies can be utilized for this purpose, for example Wi-Fi, Bluetooth [32], UWB [6,12,18,33], ZigBee [34], LoRa [35], Cellular (e.g., 4G [36], and 5G [37]), and even RFID [38]. When used with fingerprinting, various signal parameters (or even their combination) can be used: Received Signal Strength (RSS) [39], Channel State Information (CSI) [40]. Although creating a reference map of the localization environment is quite costly and labor-intensive, good results can be achieved with it in Non-Line of Sight (NLoS) cases, where there may be additional signal distortions due to obstacles between the transmitter and receiver. Note that RSSI, since it is based on a path loss model, is prone to ambiguity, and is often combined with other parameters (e.g., CSI), so that reference measurements may provide a more unique description about the environment. On the other hand, range and angle-based methods (ToA, TDoA, AoA, ADoA) provide the best results when Line of Sight (LoS) is available and the signal can be transmitted without any interfering structural obstacles, lowering shadowing and multipath effect.

Another approach uses radio-interferometric measurements [41] for localization and tracking. The system performs low frequency signal processing on interference signals instead of using high frequency signal processing, and uses RSSI along with phase-difference measurements to estimate the tracked object's position with the intersection point of hyperbolas.

Acoustic (audible and ultrasound) techniques [42,43] utilize angle and range-based principles to estimate the position or even the orientation of the target object. The system proposed in [44] uses a number of broadband ultrasonic transmitters (reference points) in known positions with radio frequency synchronization, which transmit simultaneously using Frequency-Hop Spread Spectrum. Mobile nodes (target objects) utilize a circular sensor array. When they receive at least three signals, they are able to estimate their own position (with ToF) and orientation (with AoA) using a multilateralization algorithm. A big drawback of acoustic positioning is that the speed of sound is temperature dependent, so the temperature of the environment is also needs to be monitored for compensation purposes.

Inertial Measurement Units (IMU) are typically used in pedestrian dead reckoning techniques with other sensor modalities using data fusion [45]. When not combined with other positioning methods, dead reckoning may still provide good results for room level localization in indoors, moreover, the system does not require a beaconing infrastructure, and is low cost. In [46] the relative movement trajectory of the target object is fitted to a map of the environment with which the position estimate becomes absolute and the movement trajectory's drift is also lowered.

Digital cameras are also used in localization solutions, where some kind of feature extraction with image processing is involved to obtain and identify reference points in the captured camera images (e.g., solution [47] using circular feature extraction method [48]). The reference points may be an infrastructural part of the positioning system, e.g., pre-installed markers (Aruco, QR, etc.) or LED beacons [49, 50] in the environment, other solutions use algorithmically extracted feature points of the captured image with Simultaneous Localization And Mapping (SLAM) methods [51] with which not only the positioning of the target object, but also the mapping process of the environment can be achieved at the same time.

Light-based approaches don't necessarily use digital cameras. Laser scanners provide very high accuracy in the mm/cm range, but also at a very high price tag [52]. Localization can also be achieved with photodiodes or phototransistors. In [53] a simple photodiode-based positioning technique was proposed, which utilizes Visible Light Communication (VLC) to transmit additional data (e.g., the reference point's ID, or even its position). The proposed approach uses pre-installed LED beacons (reference points) on the ceiling, at known locations. These beacons transmit their position simultaneously with different subcarriers, to reduce possible interference. A simple photodiode (the target object to be localized) is placed underneath these beacons, facing upwards. The system has some limitations: the movement plane of the photodiode has to be parallel with the plane where the beacons are placed; the distance between the two planes has to be known; the tilting of the sensor is prohibited, or needs to be compensated. The proposed solution uses measured RSS as range-based measurements (TOA principle) and an experimentally measured angle-gain profile for location estimation. In [53] the beacons transmit their positions with a high frequency subcarrier (for 3 beacons: 2.0, 2.5, 3.0 MHz). In [54], and in [55] a similar setup is used, but the beacons transmit their IDs using OFDM, and OFDMA, respectively. In [56] a two-stage neural network is used to learn the environment using the LED beacon's RSS. A more common way of indoor positioning with visible light is to use angle measurements with AoA/ADoA principles [11,17,57,58]. The positioning method proposed in [59] uses a quadrant photodiode (2 by 2 grid sensor arrangement) with a single aperture to measure the angle of the incoming light. In [60] an entire array of photosensitive sensors (in a circular shape) was used to obtain azimuthal angle measurements in 360°.

1.3 Research goals and problem statements

Off-the-shelf cameras present a promising opportunity for the development of indoor positioning systems due to the advancements of camera technology in recent years: their quality is increasing, yet they become more and more affordable and widely available. With the advancements of computer vision and image processing algorithms off-the-shelf cameras can be used to identify and track visible light patterns more accurately. This enables the development of robust positioning solutions that work effectively in complex indoor environments, where other positioning technologies cannot provide good performance or may not be applicable.

My main research goal was to create a robust visible light-based indoor positioning system, that is able to provide real-time position and orientation estimates in the centimeter range, while still can be considered inexpensive.

Optical positioning techniques utilize “anchor” features in images to determine the position and orientation of the camera sensor. The position of these reference points is either estimated on the go, or measured at the installation phase of the system. Image feature extraction methods are used to detect possible anchor points based on their descriptive properties, which is a difficult task: there exists several algorithms tailored to specific needs, but there is no universal solution; it can be computationally expensive; its success rate may depend on the properties of the environment, and may provide several non-reliable, and false detections. Using modulated LED light fixtures as anchor points is beneficial in multiple ways: the detection of such features is not computationally demanding; they can be easily detected in the image as bright spots, and can be identified based on their unique modulation patterns; when operated in the visible light spectra, they may also be part of the environment’s lighting infrastructure. It is also important to note that the modulation signal of the protocol needs to be high enough to achieve flicker-free operation that does not disturb the human eye (>> 200 Hz). However, the sampling rate (or frame rate) of off-the-shelf cameras is usually limited to 30-60 Hz, inevitably causing undersampling. Thus, special communication protocols need to be used that can operate even when the transmitted data signal is undersampled.

Undersampled Frequency-Shift On-Off Keying (UFSOOK) [61] and Undersampled Phase-Shift On-Off Keying (UPSOOK) [62] are two visible light communication (VLC) protocols found in the literature that can be used to transmit data between a light source and a camera sensor. They can also be adjusted to minimize the flickering for the human eye.

The features of UFSOOK and UPSOOK made them promising to be used for beacon identification for the visible light-based positioning system. My first goal was to analyze them and provide detailed explanation how various error sources may affect their operation. (Sections 2-3)

While examining UFSOOK and UPSOOK it turned out that they need time synchronization and precise clock signal generators to operate reliably, which is not feasible in most practical cases, and reduces the beacons’ cost-effectiveness and complexity. Also, their proposed decoding method depends on the sampled light intensity, which makes them impractical for positioning applications where the position of the transmitters and receivers may change over time.

My second goal during my research was to address the trackability and reliable identifiability of the beacons without synchronization by: designing an LED light fixture that is easy to track in the camera image and can be used to transmit beacon IDs for the positioning system; developing a robust communication protocol tailored for the periodical transmission of constant beacon data (e.g., beacon ID, beacon position). (Sections 4-5)

The third topic of my research was directed towards the positioning methods used in the indoor positioning system. Most solutions require at least 6 beacons to operate, due to the degree of freedom in a general, 3-dimensional case. The type of lens used for the camera also has an impact on the expenses of the system: by using a fisheye lens, the field of view of the camera sensor is gradually increased, lowering the beacon count needed for the operation of the positioning system. For most practical applications, the estimation of a 2-dimensional position and orientation is sufficient, which decreases the degree of freedom, and further decreases the minimum number of beacons required by the system. A reliable positioning solution also has to handle outlier measurements in the system, whose occurrence is unavoidable in most cases due to reflections and other measurement error possibilities.

My third goal during my research was to develop robust, fast positioning solutions that can provide 2-dimensional position and orientation estimates in real-time while mitigating the impact of outlier measurement errors in the system. (Sections 6-7)

In Section 1.4 a novel, visible light-based indoor positioning and tracking system is proposed, which utilizes an off-the-shelf global shutter digital camera with fisheye lens as to be localized target object and pre-installed LED light fixtures using VLC as beacons. The motivation behind the system is to merge the good properties of digital camera-based and visible light-based positioning techniques: the good resolution of recent digital cameras is very beneficial for angle measurements for AoA/ADoA-based positioning, moreover, cameras are able to support multiple beacons out of the box (they appear as distinct blobs in the captured image); LED fixtures with visible light communication capabilities may provide reliable, robust and easy-to-detect features in the camera's image stream.

1.4 Proposed localization system's architecture

In this section the proposed indoor positioning system (IPS) is presented, which was also the testbed for the various VLC protocols and positioning methods presented later in the thesis. The design considerations mainly targeted the practical use case in a warehouse, where different kinds of automated guided vehicles (AGVs) and forklifts have to be localized. In most IPS use cases the 2-dimensional position estimate of the target is sufficient, as in this case, since these vehicles usually move on an approximately flat plane with not so many elevations.

The system's infrastructure (see Figure 1.5) consists of pre-installed LED light fixtures, in known positions, which act as reference points.

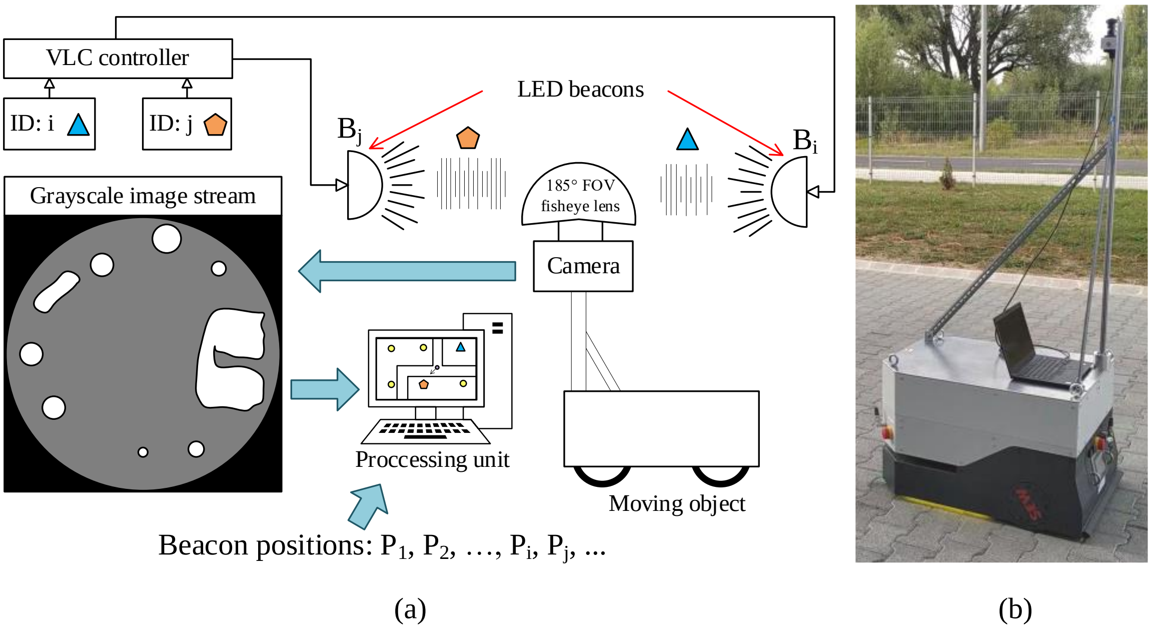


Figure 1.5 (a) Hardware architecture of the proposed indoor positioning system: LED beacons B_k are installed at known positions P_k in the environment and continuously transmit their identification number k using a visible light communication protocol. A digital camera with fisheye lens is attached to the target object, which we would like to localize, and captures a continuous video footage of the environment. A processing unit calculates the real-time position estimate of the moving object based on the camera's image stream and the known location of the beacons. (b) Application example: automated guided vehicle with the attached camera and a processing unit (notebook).

These beacons are controlled according to a visible light communication protocol so that they are able to continuously transmit their own identification number. The receiver part of the system is a digital camera which captures its environment along with these LED beacons and provides a continuous image stream for a processing unit. The processing unit is responsible for the identification of the reference points in the captured image stream, and for the calculation of the real-time position and orientation estimate of the camera based on the position of the identified reference points in the consecutive images and their known real-world location. The pose (i.e., the position and orientation) estimate of the target object is then obtained after the application of a trivial geometric transformation (applying the constant translation and rotation between the camera and the target object) on the calculated camera pose estimate.

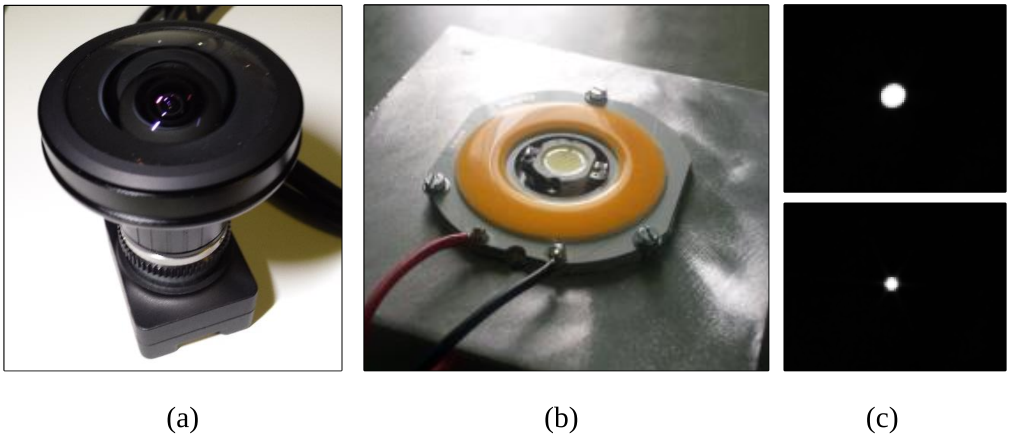
Digital cameras with global shutter are advised to be used in the system, instead of rolling shutter ones, so that every pixel of the sensor captures the incoming light at the same time avoiding the rolling shutter distortion effect. The camera is equipped with a 185° field of view fisheye lens and is mounted above the AGV or forklift, facing upwards, to minimize the occurrence of obstacles between the camera and the beacons providing good LOS availability.

The LED beacons can be located on the ceiling, on the horizon of the camera or somewhere in between. Placing them on the horizon may provide the best angular resolution for angle-based positioning with the fixed upwards looking camera, but other vehicles with the same camera height may interrupt the LOS availability of the beacons. The white light LEDs that were used for beaconing in the system are 10-20 W of power to be easily visible by the camera from higher distances (20-30 meters), and can also be used to illuminate the interior environment being part of the lighting infrastructure.

Note that one may find it tempting to use infrared LEDs in the beacons to minimize possible interferences with other light sources, and to be able to use communication methods with lower frequencies (that may disturbingly flicker with visible light), since the human eye is not that sensitive in that light spectrum. However, high power IR light may cause serious eye injuries if the beaconing infrastructure is designed to be too powerful, and extra care must be taken in that case.

Using the visible light spectrum may also allow the usage of existing light fixtures as passive beacons (auxiliary beacons without VLC) if they provide a consistent, reliable light source without flickering (even for the camera) and without other defects in the camera image. It is also important to use the VLC protocols with such settings that will not cause intense flickering of the light for the human eye. In general, a blinking frequency above 200 Hz can be considered safe [63].

Figure 1.6 shows an image of the digital camera with a 185° fisheye lens attached and an LED beacon that were used in the localization system.



(a) (b) (c)

Figure 1.6 (a) Digital camera with fisheye lens. (b) Trackable LED beacon design. (c) Camera image of the LED beacon when the outer ring is on (upper) or off (lower).

The LED beacon is designed to be easily detectable and trackable by the IPS. As opposed to many VLC methods, the protocols used in this system are not utilizing the measured light intensity changes of a beacon directly, but the changes of its size in consecutive camera images. To achieve the size changing behavior in the captured camera images, the beaconing light fixture consists of two separate high power Multiple Chip-On-Board (MCOB) LEDs: a smaller circular one, and a larger ring one around it. At least one of the LEDs are always on, hence the trackability of the beacon. Section 4 presents the trackable beacon design and the first proposed VLC protocol (TUPSOOK) that was used with it at that time in detail. Using global shutter cameras and size-based decoding also makes it possible to detect the beacons from longer distances, as opposed to the rolling shutter solutions (e.g., Luxapose [64]).

The camera was calibrated using an improved version of the Omnidirectional Camera Calibration (OCamCalib) Matlab toolbox [65], [66] with 16 images of a conventional 7 by 9 calibration chess pattern printed on a A4 sized paper, which is favored by many camera calibration methods due to the ease of use of corner extraction methods. Grayscale cameras provide more accurate measurements than colored cameras [67] with general calibration methods, and also the detection of blobs is more accurate as all the pixels share the same sensitivity to white LED beacons. This is especially important when the beacons are sensed from longer distances, and they are represented on the camera image only by a few pixels.

The functional diagram of the localization system's Processing unit is shown in Figure 1.7. The processing modules that the system is made up of are the following:

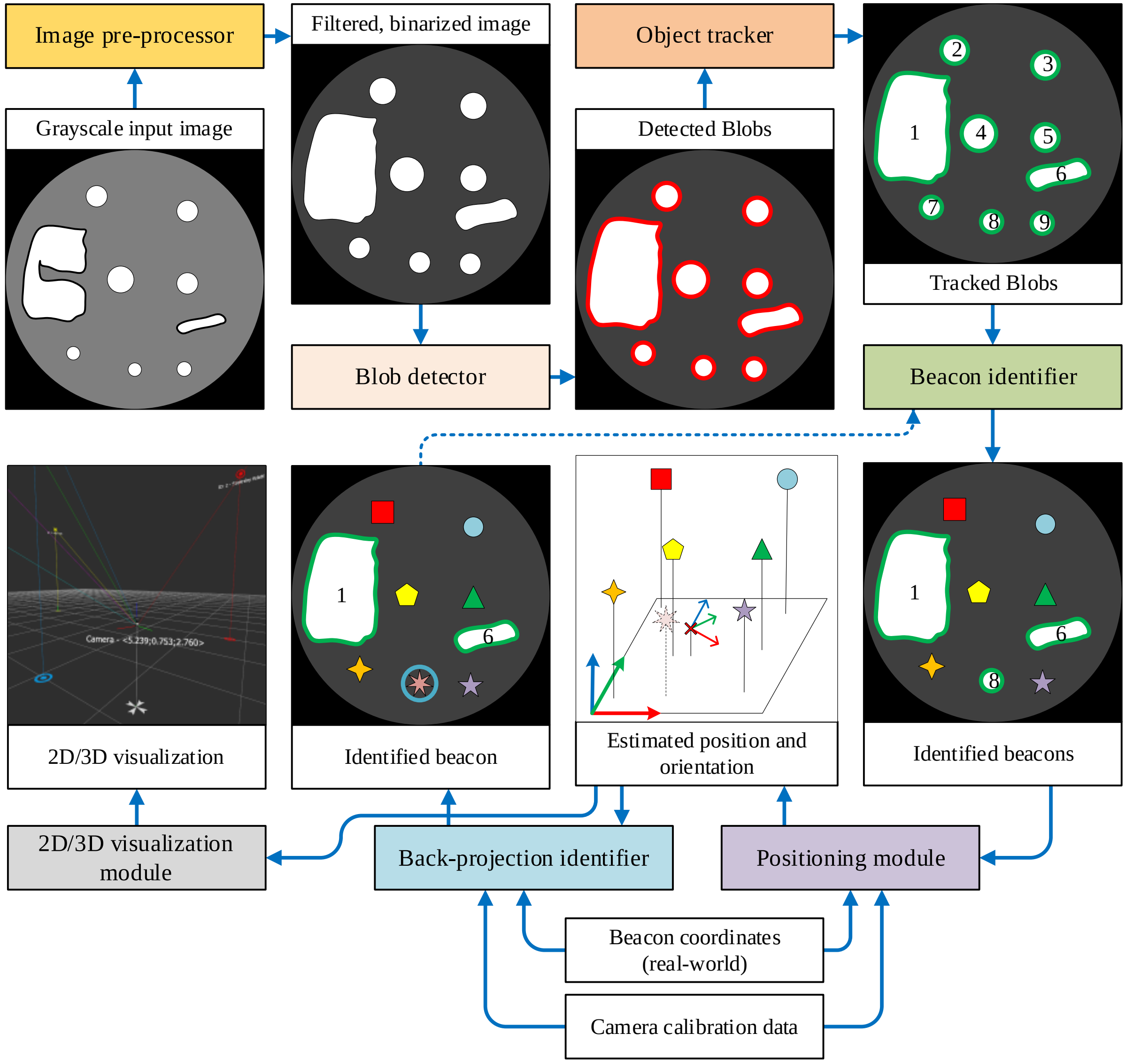


Figure 1.7 Functional diagram of the Processing unit. The forward data flow is represented with solid blue arrows, and feedback with dashed blue arrow. The 2D/3D visualization module is an example output module.

Image pre-processor: The raw input grayscale image frames of the video stream start their journey with image filtration at the Image pre-processor module, which aims to provide a consistent, noise-free, binarized output image. The operation of the module is shown in Figure 1.8. The module houses a Gaussian-filter, an image binarization step, two dilation and one erosion operation. The dilation and erosion operations can even be configured to do a morphological close or open operation (by bypassing the last or first dilation). The kernel size k_n of the operations, the ρ percentage and Q binarization threshold level are experimentally set. Apart from the binarization, every other step can be bypassed if deemed by the user as not needed. The image processing part of the module was implemented with the cross-platform Open-source Computer Vision library (OpenCV) [68].

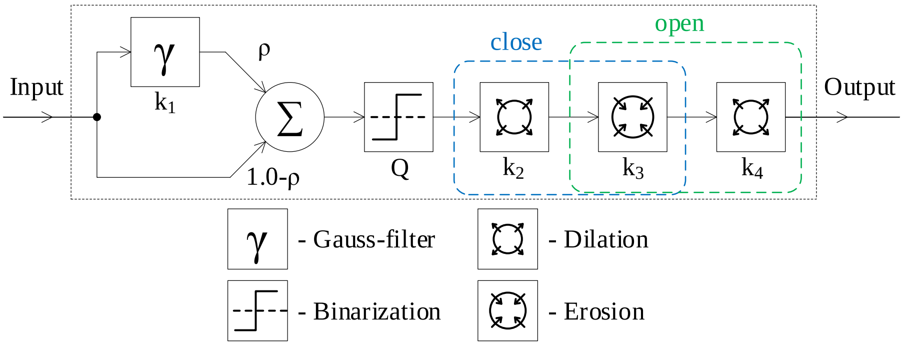


Figure 1.8 Image pre-processor module's operation diagram. The module supports the setting of kernel size k_n of the Gaussian filter and the dilation and erosion operations (bypassed if set to 0), the filter application percentage ρ and the binarization threshold level Q. A morphological close or open operation can also be achieved.

Blob detector: This module analyzes the binarized image to find consistent, white blobs which are possible LED beacons. The center point and area (in pixels) of the detected blobs are also calculated by the module. The blob's center point will be handled as the approximation for the center point of the beacon in the image, which is easy to determine, and introduces a small amount of error if the beaconing is properly designed (e.g., visible, yet small beacons in the image). The areas of the blobs are used by the VLC protocol at the Beacon identifier module. The Blob detector module used in this system was implemented with the *connectedComponentsWithStats* function of the OpenCV library, which, as its name suggests, extracts the connected white areas (potential beacon candidates) from the image along with their statistical information (e.g., midpoint of the connected area, pixelwise size of the area, bounding box, etc.). The speed of this method is remarkably fast (~1.5 msec on a conventional laptop), which makes it possible to use in real-time applications.

Object tracker: To be able to retain the identity of the blobs between the consecutive images, an Object tracker module is incorporated in the system. The operation of the module is shown in Figure 1.9. This module is implemented using an open source multitarget tracker library [69]. The library offers several tracking algorithms, from which the Hungarian method-based one was used with Euclidean distances between the detected blobs' center points in the image and the tracked blobs' latest trajectory elements. The distance, as cost, is minimized with the algorithm creating an optimal matching of the newly detected and already tracked blobs. Each tracked object is backed up with a Kalman-filter using a constant velocity model that smooths the trajectory of the tracked blob and is able to predict its movement for a short period of time if it is not visible in the image. If a tracked object is not visible for too many consecutive image frames it is removed from the tracking system. The threshold is configured experimentally or is based on the properties of the VLC protocol in use. Apart from the trajectory, the sizes of the blobs are also stored for each tracked object in separate buffers which will be used later by the Beacon identifier module to identify the tracked objects as beacons using the beacon database. The processing time of the Object tracker module is less than 1 msec in this application scenario. The use of underexposed, filtered camera images on which mostly only the beacons are visible is highly beneficial.

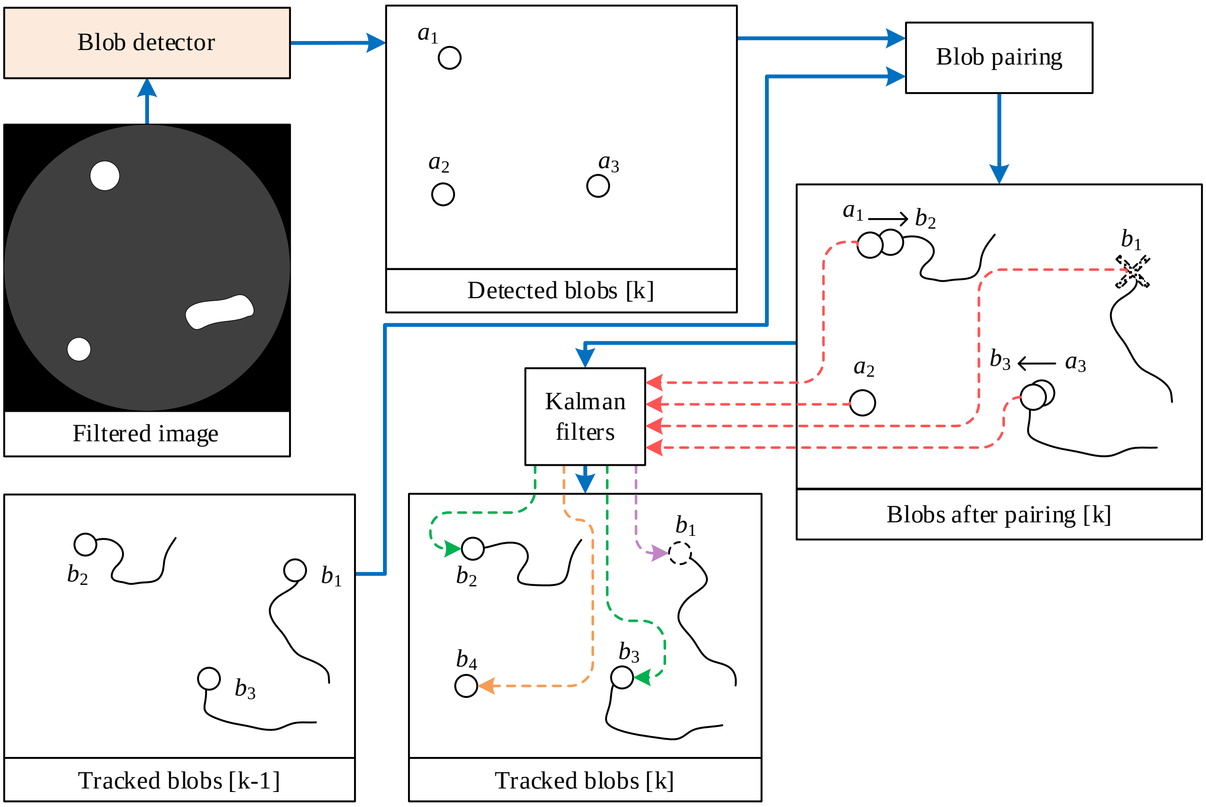


Figure 1.9 Operation diagram of the Object tracker module.

Beacon identifier: This module is responsible for the identification of the beacons. It uses a Visible Light Communication protocol to decode the transmitted identification number of the active beacons and stores the tracking ID and the assigned beacon ID in its database. Based on the selected VLC protocol implementation and beacon design the decoding may happen based on the measured consecutive light intensity changes of the beacon (e.g., the value of the pixel at the center point of a blob), or as in our case, it happens based on the stored consecutive size changes of the blob representing the beacon in the captured image. Two existing VLC protocols (found in the literature) are analyzed in Sections 2-3 in detail, and two proposed VLC protocols are presented in Sections 4-5 in detail.

Positioning module: The positioning module estimates the position and orientation of the target object (camera). First, the tracked and identified beacons' center points in the image are transformed from the 2-dimensional image plane into the camera's 3-dimensional local coordinate system as direction vectors using the camera's calibration data. Perspective-n-Point algorithms [70],[71] are widely used for the generic, purely visual 3D camera pose estimation case, however they usually require at least 4 (typically 6, due to degrees of freedom) reference points. The angles can be measured in 3D, where both the azimuth and elevation of the beacon are measured [17], or in 2D, using only the azimuth values [58]. The azimuth-only estimation requires much less computation [72], and is sufficient for applications where only the 2D location is to be determined and the elevation is irrelevant (e.g., when a fork-lift truck is tracked). The sensor's direction, however, must be known or measured in the azimuth-only method; the simplest solution being when the sensor is facing upwards [58]. The estimation of the sensor position can be made by solving an equation system [73]; or in case of redundant measurements, least squares methods [17], consensus-based approaches [58], exhaustive search methods [17] were proposed. In the proposed system's case, we are only interested in the 2-dimensional position of the camera, which is upwards-facing and is approximately parallel (or the measurements can be compensated with an IMU (e.g., [58]), so that the camera's local coordinate system's up axis is parallel) with the up axis in the world coordinate system. Only the azimuthal angle of the direction vectors is retained and used for positioning. The orientation of the camera is not assumed to be known (only that it is facing upwards), meaning that angle difference measurements between identified beacon pairs from the camera's point of view have to be used in the module with the ADoA positioning principle. I proposed two robust methods to address the positioning task of this module. The proposed methods were tested and used in the system. Sections 6 presents a heuristic, geometry-based approach using inscribed angles and arc intersections. Section-7 presents a RANSAC-based robust positioning approach in detail. Both methods were designed to provide outlier-tolerant estimates.

Back-projection identifier: This module is used in the system to support the usage of passive beacons (light fixtures without VLC) and to speed up the identification of active beacons (requiring only one image frame, as opposed to e.g., 25 consecutive frames). The module requires an already calculated pose (position and orientation) estimate of the camera, the camera's calibration data and the beacon database (real-world coordinates). The direction vectors from the estimated pose to the not yet identified beacons are calculated and projected back into the camera's image coordinate system using the inverse of the transformation that was used to calculate the direction vectors from the image coordinates. When the system is used in a 3-dimensional positioning mode, these back-projected vectors will be points in the image, but in our 2-dimensional case they only contain azimuthal angle information, so they are represented as lines in the camera image, meaning that any blob's center point that is located on that line is a possible beacon-candidate. If the height of the camera and the height of the beacons in the world coordinate system are known, the back-projection method using the 3-dimensional case is also possible.

Output module: The purpose of this module is to transmit the estimated camera position and orientation to other services. In Figure 1.7 the output module is represented by the 2D/3D visualization module that displays the position of the beacons and the estimated position and orientation of the camera along with its trajectory.

2 Analysis of the UFSOOK protocol

The Undersampled Frequency-Shift On-Off Keying (UFSOOK) protocol [61] is a simplex VLC protocol found in the literature that can be used to transmit data between an ordinary camera as the receiver, and controllable LED light fixtures (LED beacons) as transmitters. The protocol's operation is illustrated in Figure 2.1. For the data transmission three different symbols are used: HEADER, MARK, and SPACE symbols, each symbol (or bit) requires 2 samples for decoding.

Data packets are preceded by a Start Frame Delimiter, containing a HEADER symbol and a MARK symbol (see Figure 2.1). The MARK symbol in the Start Frame Delimiter is used to mitigate bad data transmission cases (e.g., caused by phase slippage when the MARK symbol in the Start Frame Delimiter is incorrectly detected as a SPACE). The HEADER symbol of the next packet is also used as a footer for the actual packet, achieving framing. A MARK symbol represents the logical 1 value, containing two consecutive samples with different light intensities (i.e., OFF-ON as in Figure 2.1, or ON-OFF). A SPACE symbol represents the logical 0 value, where two consecutive samples have the same light intensity state (i.e., ON-ON as in Figure 2.1, or OFF-OFF). Each symbol is generated as a square wave with different frequency, which can be chosen to seem flicker-free for the human eye, but not for the camera sensor. Thus, the communication protocol can be used in the environment's lighting infrastructure too.

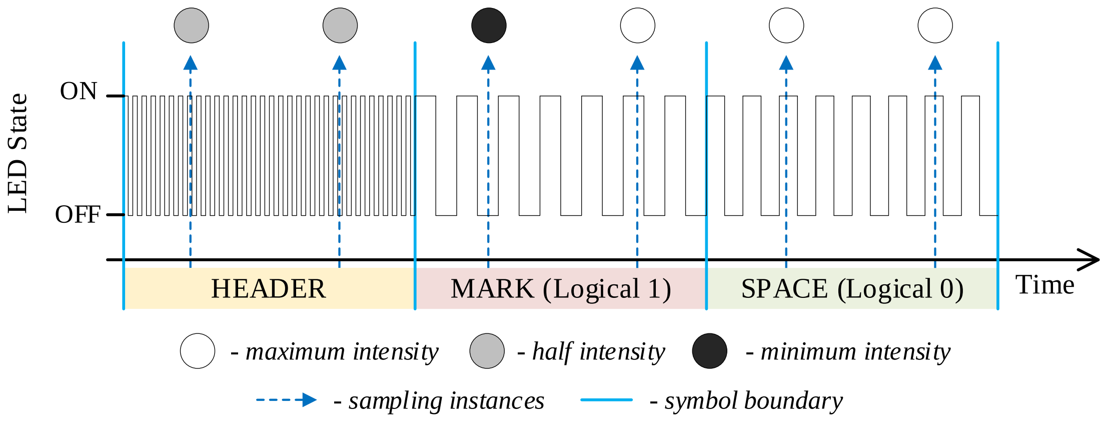


Figure 2.1 The operation of UFSOOK protocol.

The HEADER symbol's time period is chosen to be small with respect to the sensor's exposure time (e.g., $f_{HEADER} > 10$ kHz), thus, the camera senses it as half (average) intensity (see the first two samples with detected half intensity signal values in Figure 2.1). The SPACE symbol's frequency is chosen to be an integer multiple of the camera's sampling frequency:

$$f_{SPACE} = \frac{1}{T_{SPACE}} = nf_{CAM} \tag{2.1}$$

This choice of frequency ensures that consecutive samples are taken at the same phase of the signal, thus, providing the same values (see the last ON-ON sample pair in Figure 2.1).

The MARK symbol's frequency is chosen as follows:

$$f_{MARK} = \frac{1}{T_{MARK}} = (n - 0.5)f_{CAM}, \tag{2.2}$$

which leads to consecutive samples taken 180° out of phase, resulting in opposite signal values (see the second sample pair in Figure 2.1, showing OFF-ON values).

As a typical example, a camera with sampling rate of 30 Hz and $n = 4$ results $f_{SPACE} = 120Hz$ and $f_{MARK} = 105$ Hz. Notice that each symbol contains 2 samples, hence every symbol (i.e., bit) has a transmission time of $2/f_{CAM}$.

2.1 Camera sampling model

In order to be able to perform the mathematical analysis of the protocol's performance, a model on the operation of the camera and the light sources will be used.

We assume the camera has global shutter, i.e., the sampling of each pixel is performed simultaneously. The sampling is performed during the exposure time of the camera sensor. If the exposure time is S and the luminous intensity of the light source is $I(t)$, where t is time, then the detected light intensity d at time instant t_0 is the following:

$$d(t_0) = \alpha \int_{t_0 - S}^{t_0} I(t)dt + n(t_0), \tag{2.3}$$

where α depends on the camera's various properties and settings (e.g., ISO number and aperture size) and $n(t_0)$ is the additive noise component. Notice that (2.3) is valid only when the camera is not saturated, i.e., the value computed in (2.3) is not larger than the maximum value the camera can represent (in case of cameras, sampled values are usually represented by 8-12-bit unsigned integers).

Figure 2.2 (a) illustrates the effect of sampling with an integral camera model. The square-wave light source's intensity signal I is sampled according to (2.3) in an aperture window w with width S. The detected light intensity d, as a function of the sampling time instant, has a trapezoidal shape, where the rising and falling edges have width of S.

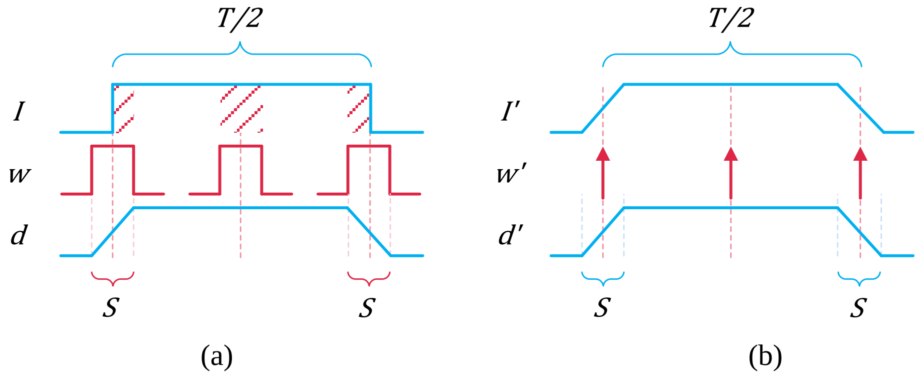


Figure 2.2 Integral sampling and the equivalent model. (a) Integral sampling model. I is the luminous intensity of the light source, w is the sampling window with aperture time S, and d is the detected light intensity as a function of the sampling time instant (which is at the center of the sampling window). (b) Equivalent model. $I' = d$ is the modified light source, w' is impulse sampling, and $d' = d$ is the detected light intensity.

Notice that the sampling time instant is in the center of the aperture window in case of Figure 2.2 to better illustrate the connection.

The operation of the system can be modelled with an equivalent model, illustrated in Figure 2.2 (b): the equivalent light source produces the trapezoidal intensity curve I' (instead of square wave I), as follows:

$$I'(t_0) = \int_{t_0 - S}^{t_0} I(t)dt, \tag{2.4}$$

while the camera performs pulse sampling (instead of the integral sampling of (2.3)):

$$d'(t_0) = \alpha I'(t_0) + n(t_0). \tag{2.5}$$

For every time instant t_0, the two models are equivalent: $d'(t_0) = d(t_0)$.

The camera is assumed to perform sampling with constant frequency (or frame rate) f_{CAM}. We assume that the short-term stability of the camera is good enough, so we can ignore effects of frequency change or jitter.

It is not assumed however, that the light modulator and the camera are synchronized. This would be impractical in most applications; thus, we allow slight differences between the ideal and real frequencies. Let the ideal camera frequency be f_{CAM} and the corresponding ideal sampling period be $T_{CAM} = 1/f_{CAM}$, and let us assume that the modulator operates using this ideal frequency, according to (2.1) and (2.2). If the actual camera frequency is $\tilde{f}_{CAM} = 1/\tilde{T}_{CAM}$ then let us denote the difference between the actual and ideal sampling interval by δ_{CAM}, as follows:

$$\delta_{CAM} = \frac{1}{\tilde{f}_{CAM}} - \frac{1}{f_{CAM}} = \tilde{T}_{CAM} - T_{CAM}, \tag{2.6}$$

and the size of the difference

$$\delta = |\delta_{CAM}|. \tag{2.7}$$

2.2 Error analysis

With respect to the assumptions at the end of Section 2.1 the effect of frequency error (Section 2.2.1) and the effect of noise (Section 2.2.2) on the data transmission will be analyzed in the following.

2.2.1 Frequency error

Let us define the noise-free samples $d_1(t)$ and $d_2(t)$, as follows:

$$\begin{gathered} d_1(t) = \alpha I'(t), \\ d_2(t) = \alpha I'\left(t + \tilde{T}_{CAM}\right), \end{gathered} \tag{2.8}$$

where $d_1(t)$ represents a potential first sample of a symbol, in case the sampling occurs at time t (called primary sampling time), while the next sample, sampled $\tilde{T}_{CAM}$ time later, will be denoted by $d_2(t)$. The decoding graphs, shown in Figure 2.3, contain $d_1(t)$ (solid blue lines) and $d_2(t)$ (dashed red lines). Thus, from the decoding graphs, for each symbol, the value of the first and the corresponding second sample can be read, as a function of the primary sampling time.

In case of the SPACE symbol, according to (2.1), $I(t) = I(t + T_{CAM})$, thus, also $I'(t) = I'(t + T_{CAM})$. If δ_{CAM} error is present according to (2.4) then $I'(t) = I'(t + \tilde{T}_{\mathrm{CAM}} - \delta_{CAM})$, thus, $d_1(t) = d_2(t - \delta_{CAM})$; in this case the two curves of the decoding graph are shifted by δ_{CAM}, as shown in Figure 2.3 (a). For the MARK symbol, however, $I(t) = \,!\,I(t + T_{CAM})$, according to (2.2), where ! denotes the opposite intensity value. If $I(t)$ varies between I_{L} and I_{H} then $I(t)$ can be expressed as $I(t) = I_{\mathrm{L}} + I_{\mathrm{H}} - I(t + T_{CAM})$. Similarly, if $I'(t) = d(t)$ varies between A_{L} and A_{H} then $I'(t) = A_{\mathrm{L}} + A_{\mathrm{H}} - I'(t + T_{CAM})$, thus, $d_1(t) = A_{\mathrm{L}} + A_{\mathrm{H}} - d_2(t - \delta_{CAM})$. Thus, the decoding graph for the MARK symbol contains a trapezoid signal and its shifted inverse, as shown in Figure 2.3 (b) and (c), for different values of δ_{CAM}.

The decoding graph also contains the threshold Q. If $d(t) > Q$ then the detected source is considered to be ON, otherwise it is OFF. Ideally, for SPACE symbols either $d_1(t) > Q$ and $d_2(t) > Q$ (ON-ON) or $d_1(t) \leq Q$ and $d_2(t) \leq Q$ (OFF-OFF); and for MARK symbols the samples are either $d_1(t) > Q$ and $d_2(t) \leq Q$ (ON-OFF) or $d_1(t) \leq Q$ and $d_2(t) > Q$ (OFF-ON). However, as Figure 2.3 shows, there are time intervals, the lengths of which are denoted by λ_A and

λ_B, where these constrains do not hold and thus, the symbol decoding returns false results. Let us call these intervals “dangerous”.

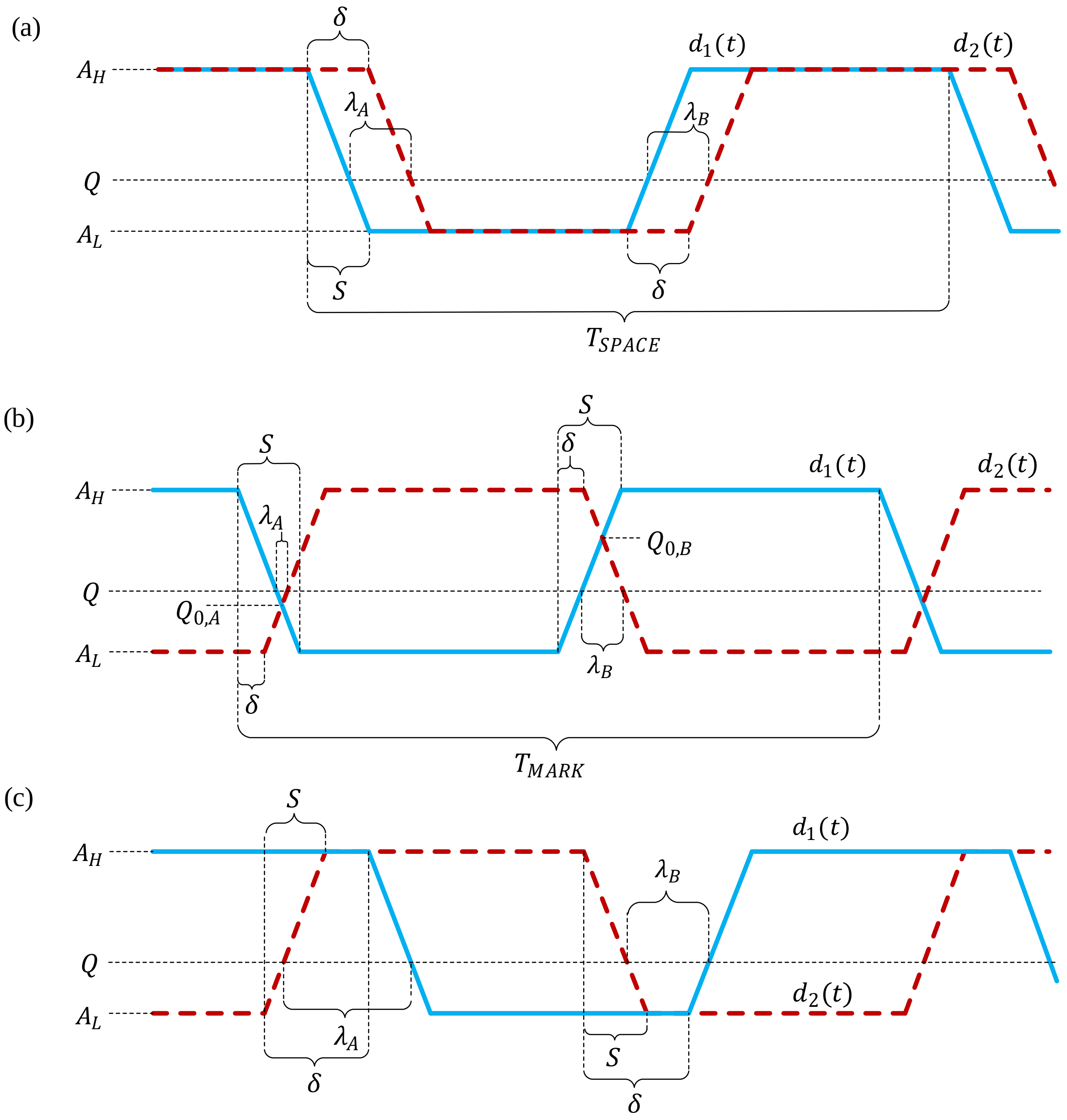


Figure 2.3 The effect of frequency error on the sampling. (a) SPACE symbol, (b) MARK symbol with $\delta < S$, (c) MARK symbol with $\delta \geq S$. The δ_{CAM} is negative in case of (a) and (b), meaning that the second sample is taken a bit earlier than the ideal resulting in that ideal $d_2(t)$ is shifted to the right on the decoding graphs; δ_{CAM} is positive in case of (c) resulting in a shift of the ideal $d_2(t)$ towards the opposite direction.

In the synchronized case (when the frequency error is 0, i.e., $\delta = 0$) the sampling is always performed at the same phase of the blinking signal, thus, the decoding is either always good (the primary sampling time is not in a dangerous interval), or it is always bad (the primary sampling time is inside of a dangerous interval). In the unsynchronized case ($\delta \neq 0$), however, the phase of the sampling time is continuously changing, thus, the primary sampling time sweeps along the timeline. The detection is good, while the primary sampling time is outside of the dangerous intervals; and the detection is faulty, when the primary sampling time is inside of one of the dangerous intervals.

First let us investigate the detection graph for SPACE symbols. As Figure 2.3 (a) clearly shows, the width of the dangerous intervals is $\lambda_A = \lambda_B = \delta$. Since there are two dangerous intervals with cumulative length of 2δ in a blinking period T_{SPACE}, the bit error rate (BER) μ_{SPACE} in case of $\delta \neq 0$ can be estimated as follows:

$$\mu_{SPACE} = \frac{2\delta}{T_{SPACE}} = 2\delta n f_{CAM}. \tag{2.9}$$

Now let us examine the MARK symbol. Here the two cases of $\delta < S$ and $\delta \geq S$ will be handled separately. The case of $\delta < S$ is shown in Figure 2.3 (b). The signal amplitudes at the intersections of $d_1(t)$ and $d_2(t)$ are denoted by $Q_{0,A}$ and $Q_{0,B}$, as shown in Figure 2.3 (b). Using similar triangles, it follows that

$$\frac{Q_{0,A} - A_{\mathrm{L}}}{\frac{S-\delta}{2}} = \frac{A_{\mathrm{H}} - A_{\mathrm{L}}}{S}, \tag{2.10}$$

$$\frac{A_{\mathrm{H}} - Q_{0,B}}{\frac{S-\delta}{2}} = \frac{A_{\mathrm{H}} - A_{\mathrm{L}}}{S}, \tag{2.11}$$

from which the values of $Q_{0,A}$ and $Q_{0,B}$ can be expressed:

$$Q_{0,A} = \frac{A_{\mathrm{H}} + A_{\mathrm{L}}}{2} - \frac{A_{\mathrm{H}} - A_{\mathrm{L}}}{2}\frac{\delta}{S} \ , \tag{2.12}$$

$$Q_{0,B} = \frac{A_{\mathrm{H}} + A_{\mathrm{L}}}{2} + \frac{A_{\mathrm{H}} - A_{\mathrm{L}}}{2}\frac{\delta}{S} \ . \tag{2.13}$$

If $Q_{0,A} \leq Q \leq Q_{0,B}$ (as the illustration of Figure 2.3 (b) shows) then λ_A and λ_B can be calculated, using similar triangles, as follows:

$$\frac{Q - Q_{0,A}}{\lambda_A} = \frac{A_{\mathrm{H}} - A_{\mathrm{L}}}{2S}, \tag{2.14}$$

$$\frac{Q_{0,B} - Q}{\lambda_B} = \frac{A_{\mathrm{H}} - A_{\mathrm{L}}}{2S}, \tag{2.15}$$

from which

$$\lambda_A = \left(Q - Q_{0,A}\right)\frac{2S}{A_{\mathrm{H}} - A_{\mathrm{L}}}, \tag{2.16}$$

$$\lambda_B = \left(Q_{0,B} - Q\right)\frac{2S}{A_{\mathrm{H}} - A_{\mathrm{L}}}. \tag{2.17}$$

Using (2.12)-(2.13) and (2.16)-(2.17) it follows that

$$\lambda_A + \lambda_B = 2\delta. \tag{2.18}$$

Now let us consider the case of $Q < Q_{0,A}$ (not shown in Figure 2.3 (b)). Again, using similar triangles, the following results can be obtained:

$$\lambda_A = \left(Q_{0,A} - Q\right)\frac{2S}{A_{\mathrm{H}} - A_{\mathrm{L}}}, \tag{2.19}$$

$$\lambda_B = \left(Q_{0,B} - Q\right)\frac{2S}{A_\mathrm{H} - A_\mathrm{L}}, \tag{2.20}$$

from which it follows that

$$\lambda_A + \lambda_B = \frac{2S}{A_\mathrm{H} - A_\mathrm{L}}\left(Q_{0,A} + Q_{0,B} - 2Q\right). \tag{2.21}$$

Using (2.12) and (2.13), it follows that

$$\lambda_A + \lambda_B = \frac{2S}{A_\mathrm{H} - A_\mathrm{L}}((A_\mathrm{H} + A_\mathrm{L}) - 2Q). \tag{2.22}$$

Similarly, for the case of $Q > Q_{0,B}$ the following results can be obtained:

$$\lambda_A + \lambda_B = \frac{2S}{A_\mathrm{H} - A_\mathrm{L}}(2Q - (A_H + A_L)) \tag{2.23}$$

The results (2.18), (2.22), and (2.23) for the MARK symbol can be summarized as follows:

$$\lambda_A + \lambda_B = \begin{cases} 2\delta & \text{if } Q_{0,A} \le Q \le Q_{0,B} \\ \dfrac{2S}{A_\mathrm{H} - A_\mathrm{L}}\left|A_\mathrm{H} + A_\mathrm{L} - 2Q\right| & \text{otherwise.} \end{cases} \tag{2.24}$$

Thus, the BER estimate for MARK symbol, for the case $\delta < S$, is the following:

$$\mu_{MARK} = \begin{cases} 2\delta/T_{MARK} & \text{if } Q_{0,A} \le Q \le Q_{0,B} \\ \dfrac{2S}{T_{MARK}}\dfrac{\left|A_\mathrm{H} + A_\mathrm{L} - 2Q\right|}{A_\mathrm{H} - A_\mathrm{L}} & \text{otherwise.} \end{cases} \tag{2.25}$$

The case $\delta \ge S$ is illustrated in Figure 2.3 (c). For this case, with geometrical calculations, similarly to the previous cases, λ_A and λ_B can be derived as follows:

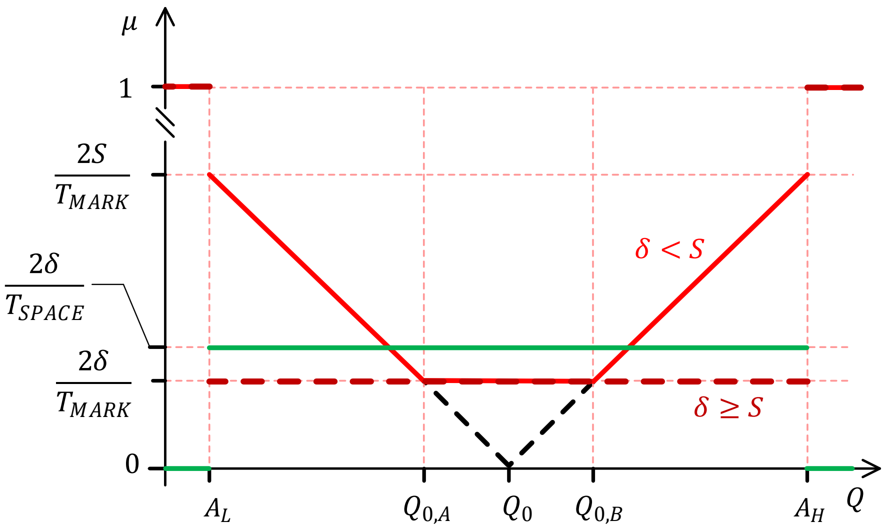
$$\lambda_A = \delta + S\frac{A_L + A_H - 2Q}{A_H - A_L}, \tag{2.26}$$

$$\lambda_B = \delta - S\frac{A_L + A_H - 2Q}{A_H - A_L}. \tag{2.27}$$

Thus, for the case of $\delta \ge S$ the BER is the following:

$$\mu_{MARK} = \frac{\lambda_A + \lambda_B}{T_{MARK}} = \frac{2\delta}{T_{MARK}} = 2\delta(n - 0.5)f_{CAM} \tag{2.28}$$

The BER, as a function of threshold parameter Q, according to (2.9), (2.25), and (2.28), is shown in Figure 2.4.



$$Q_{0,A} = Q_0 - \frac{\delta}{S}\frac{A_H - A_L}{2} \qquad Q_0 = \frac{A_L + A_H}{2} \qquad Q_{0,B} = Q_0 + \frac{\delta}{S}\frac{A_H - A_L}{2}$$

Figure 2.4 The bit error rate, as a function of threshold Q. Green: SPACE symbols, solid red: MARK symbols with $\delta < S$, dashed red: MARK symbols with $\delta \geq S$.

For extreme choices of Q, i.e., $Q < A_{\mathrm{L}}$ or $Q \geq A_{\mathrm{H}}$, everything is detected as SPACE, thus, for $\mu_{MARK} = 100\%$ and $\mu_{SPACE} = 0$. For values of Q between the minimum and maximum detected light intensity, the value of BER μ_{SPACE} depends on δ, n, and f_{CAM}, but does not depend on Q, according to (2.9). Similarly, if $\delta \geq S$, μ_{MARK} does not depend on Q, according to (2.28). When $\delta < S$, for threshold values between $Q_{0,A}$ and $Q_{0,B}$ the BER for MARK symbols is minimal, with value of $2\delta/T_{MARK}$, which is identical to (2.28). For $Q < Q_{0,A}$ or $Q > Q_{0,B}$ the BER value increases as Q approaches the minimum or maximum light intensity:

$$\mu_{MARK}(A_L) = \mu_{MARK}(A_H) = \frac{2S}{T_{MARK}} \tag{2.29}$$

The results show that the BER is proportional with the δ time difference between ideal and real camera sampling interval, the n design parameter, and the f_{CAM} camera frame rate. In practice n must be high enough to provide flicker free operation (i.e., $nf_{CAM} \gg 100$ Hz), but our results also suggest that n should be chosen as small as possible (while satisfying the flicker-free requirements), in order to provide small BER.

The results also indicate that the detection quality for SPACE symbol does not depend on the threshold parameter Q, while for the optimal detection of MARK symbols Q should be chosen between $Q_{0,A}$ and $Q_{0,B}$ (thus, the mean of the signal is a good choice). If this cannot be guaranteed, the BER is proportional with the aperture time S, and inversely proportional with the signal's amplitude $A_{\mathrm{H}} - A_{\mathrm{L}}$, thus, S should be kept small (preferably below δ, when the BER does not depend on the exact value of Q), and low signal amplitudes should be avoided. Thus, the following design rules can be stated in order to provide low BER:

- the frequency offset should be kept small, since the minimum error depends on this value;
- the threshold should be close to the ideal value of $Q_0 = (A_{\mathrm{L}} + A_{\mathrm{H}})/2$;
- small aperture time should be used, to minimize the effect of suboptimal threshold;
- design parameter n should be small, to provide large T_{SPACE} and T_{MARK} (while satisfying the requirements of flicker-free operation), to provide small minimum value for the error;
- the signal amplitude should be as high as possible, to provide a wide range for optimal threshold, and also to provide high signal-to-noise ratio.

2.2.2 Noise

For the noise analysis, we assume that the measurement noise can be modelled as Additive White Gaussian Noise (AWGN). Thus, the noisy detection signals are

$$\check{d}_1(t) = d_1(t) + n_1(t), \tag{2.30}$$

$$\check{d}_2(t) = d_2(t) + n_2(t), \tag{2.31}$$

where $d_1(t)$ and $d_2(t)$ are the noise-free samples, $n_1(t)$ and $n_2(t)$ are the AWGN components with distribution $N(0, \sigma^2)$.

For SPACE symbols detection error occurs when the detected light intensities are different, i.e., in the following two cases:

(a) $\check{d}_1(t) > Q$ and $\check{d}_2(t) \leq Q$ or
(b) $\check{d}_1(t) \leq Q$ and $\check{d}_2(t) > Q$.

Let the Q-function $Qf(x)$ denote the probability that random variable X with distribution $N(0, \sigma^2)$ is higher than x:

$$Qf(x) = P(X > x) = \frac{1}{2} - \frac{1}{2}\operatorname{erf}\left(\frac{x}{\sigma\sqrt{2}}\right). \tag{2.32}$$

If the threshold is Q then the probability that the detection, with primary sampling time t, is false due to case (a) is the following:

$$P_{err-a}(t, Q) = Qf_1(t, Q)\big(1 - Qf_2(t, Q)\big), \tag{2.33}$$

where $Qf_1(t, Q) = \ Qf(Q - d_1(t))$ and $Qf_2(t, Q) = Qf(Q - d_2(t))$. Similarly, the probability of false detection due to case (b) is the following:

$$P_{err-b}(t, Q) = \big(1 - Qf_1(t, Q)\big)Qf_2(t, Q). \tag{2.34}$$

For MARK symbols erroneous detection can happen when the detected light intensities are the same, in the following two cases:

(c) $\check{d}_1(t) > Q$ and $\check{d}_2(t) > Q$ or
(d) $\check{d}_1(t) \leq Q$ and $\check{d}_2(t) \leq Q$.

Given the threshold value Q, the probabilities of false detections, due to case (c) and (d), are the following:

$$P_{err-c}(t, Q) = Qf_1(t, Q)Qf_2(t, Q), \tag{2.35}$$

$$P_{err-d}(t, Q) = \big(1 - Qf_1(t, Q)\big)\big(1 - Qf_2(t, Q)\big). \tag{2.36}$$

Using (2.33) and (2.34), the probability of false detections $P_{err}(t, Q)$ for SPACE symbols can be computed as follows:

$$P_{err,SPACE}(t, Q) = P_{err-a}{+}P_{err-b}. \tag{2.37}$$

Similarly, using (2.35) and (2.36), for MARK symbols the probability of false detections is the following:

$$P_{err,MARK}(t, Q) = P_{err-c}{+}P_{err-d}. \tag{2.38}$$

The BER can be estimated as follows:

$$\mu = \frac{1}{T}\int_0^T P_{err}(t,Q)dt \cong \frac{1}{K}\sum_{k=1}^{K} P_{err}\left(k\frac{T}{K},Q\right), \tag{2.39}$$

where $T = T_{SPACE}$ and $P_{err} = P_{err,SPACE}$ for SPACE symbols and $T = T_{MARK}$ and $P_{err} = P_{err,MARK}$ for MARK symbols. The rightmost expression of (2.39) is a practical discrete approximation using K sample points.

The BER, as a function of Q, is illustrated in Figure 2.5 and Figure 2.6, where the BER was calculated using the discrete approximation of (2.39), for $\delta/T = 0.025$, $S/T = 0.05$, and $FS = A_H - A_L = 20$. The figures show various signal-to-noise levels: the ratio of σ/FS varied from 0.001 to 0.1. The shape of the BER function for SPACE symbols is shown in Figure 2.5.

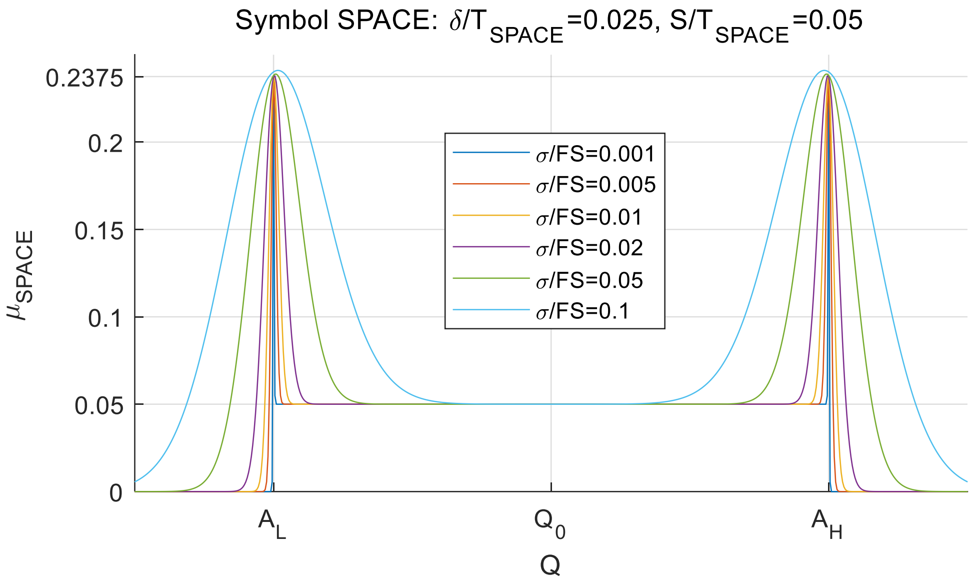


Figure 2.5 The bit error rate for SPACE symbols, as a function of threshold Q, in the presence of additive noise with variance σ.

For small values of Q it converges to the noise-free case, shown by green in Figure 2.4. For values of Q between A_L and A_H, according to (2.9), $\mu_{SPACE} = 2\delta/T_{SPACE} = 0.05$, while outside of this region $\mu_{SPACE} = 0$, which corresponds with the calculated shape of Figure 2.5. Notice that for noiseless case the BER function has a rectangular shape, but in the presence of noise two pulses appear around A_L and A_H, the width of which depend on the noise level (wider pulse for higher noise). The amplitude of the pulse can be derived as follows. Let the threshold be A_L and the noise variance be higher than 0. When either of the detected samples is low (i.e., $d_1 \cong A_L$ or $d_2 \cong A_L$, see Figure 2.3 (a)) then the probability of the false detection is 50%, for arbitrarily small, positive noise variances. For higher values of d_1 and d_2 the probability of false detections is close to 0 for small noise levels. Since the total time of either d_1 or d_2 being close to A_L is $T_{SPACE}/2 - S + \delta$, the BER at A_L in the presence of noise is

$$\mu_{SPACE}(A_L) = 0.5\frac{\frac{T_{SPACE}}{2} - S + \delta}{T_{SPACE}} = 0.25 + \frac{\delta - S}{2T_{SPACE}}. \tag{2.40}$$

The same argument can be repeated for threshold A_H, thus,

$$\mu_{SPACE}(A_H) = 0.25 + \frac{\delta - S}{2T_{SPACE}}. \tag{2.41}$$

In the case shown in Figure 2.5, the value of the BER at A_L and A_H, according to (2.40)-(2.41), is $0.25 - 0.0125 = 0.2375$, which correspond well to the values shown in Figure 2.5. Notice that the effect of noise is especially significant around the smallest and highest detected light intensities A_L and A_H. Around the optimal choice of $Q_0 = (A_L + A_H)/2$, however, the effect of noise is hardly observable.

Figure 2.6 shows the BER functions for MARK symbols. For small noise levels the shape of μ_{MARK} is similar to the function shown in Figure 2.4 with solid red line.

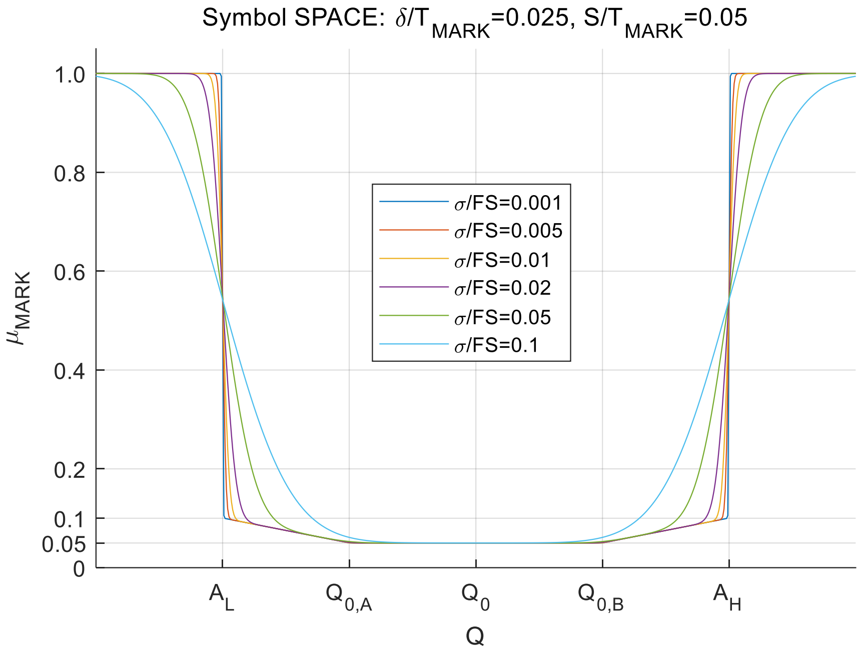


Figure 2.6 The bit error rate for MARK symbols, as a function of threshold Q, in the presence of additive noise with variance σ.

According to (2.12) and (2.13), $Q_0 - Q_{0,A} = Q_{0,B} - \mathrm{Q}_0 = FS \cdot \delta/2S = FS/4$; and according to (2.25), the minimum of the function, around the optimal value, is $2\delta/T_{MARK}$ =0.05, while $\mu_{MARK}(A_\mathrm{L})$= $\mu_{MARK}(A_\mathrm{H}) = 2S/T_{MARK} = 0.1$, and the BER is 100% below A_L and above A_H: these theoretical results are accurately represented by the function shown by deep blue line in Figure 2.6. For higher noise levels the function gets distorted, especially around A_L and A_H. Notice again that the effect of noise is small for threshold values close to the optimum. The theoretical results will be verified by real measurements in the next section.

2.3 Evaluation

The purpose of the measurements is to validate the theoretical BER results derived in Section 2.2. Thus, the two data symbols, SPACE and MARK were separately tested in a well- controlled environment. First the measurement setup and the devices used in the measurement will be introduced, then the measurement results will be presented and evaluated.

2.3.1 Measurement setup

The measurement setup is illustrated in Figure 2.7. The modulated signal was generated by a Microcontroller Unit (MCU) through a MOSFET LED driver. The LEDs were placed in separate closed (black) measurement chambers, in order to exclude external disturbances. In the experiment two separate channels were used to test the behavior of the protocol's MARK and SPACE symbols: Channel 1 was dedicated to MARK symbols, while in Channel 2 SPACE symbols were transmitted. Notice that the purpose of the test was to determine the error sensitivity of the data symbols, thus, only the symbols were transmitted continuously without any data framing (i.e., no HEADER was used). Each symbol's channel was simultaneously observed by the same global shutter camera. The recorded video stream was processed off-line, with optional controlled additive noise.

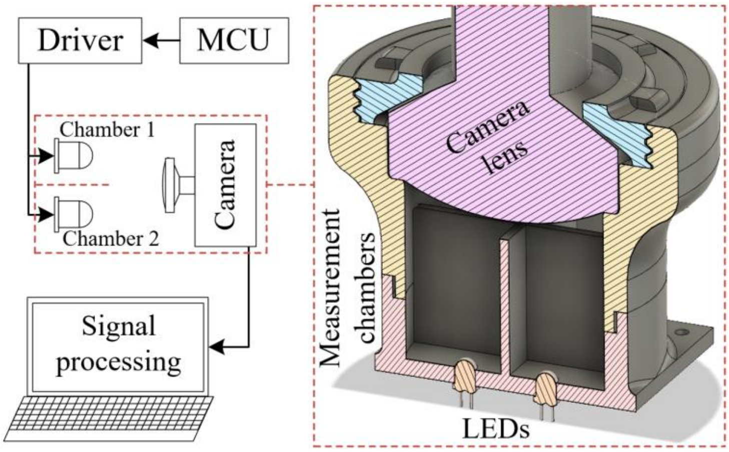


Figure 2.7 Measurement setup: hardware components (left hand side) and the optical measurement chambers

The nominal frame rate of the camera was 30FPS. The design parameter n was set to 4, thus, the frequencies of the SPACE and MARK symbols were ideally 120 Hz and 105 Hz, respectively. Since the exact camera sampling rate slightly differs from the nominal value, to experiment with multiple slip values, the LED frequencies were fine-tuned to provide the required amount of slip. In the experiments slip values were set between 20 µsec and 180 µsec. The detected signal range was between $A_L \cong 20$ and $A_H \cong 150$ with the 8-bit camera. The exposure time of the camera was varied between $S = 100$ µsec and $S = 400$ µsec.

2.3.2 Measurement results

Each symbol channel was processed separately, and the numbers of good and bad detections were determined, as a function of threshold parameter Q. The calculated symbol error rates are presented in the following figures.

Figure 2.8 and Figure 2.9 show the BER results of three experiments with $S^{(1)} = 100\ \mu\text{sec}, S^{(2)} = 200\ \mu\text{sec}$ and $S^{(3)} = 400\ \mu\text{sec}$, when $\delta = 20\ \mu\text{sec}$ was constant. The noise variance was below 1 bit. The light intensities of the LEDs were set so that the minimum and maximum detected amplitudes were approximately the same in each experiment. The representative values $Q_{0,A}$, Q_0, $2\delta/T_{MARK}$, $2\delta/T_{SPACE}$, and $2S^{(i)}/T_{MARK}$ are also shown in the figures (compare with Figure 2.4). For easier comparison, a theoretical BER diagram for MARK symbols is also shown (for better visibility, only for experiment 3). The measurements fit the theoretical results with remarkable accuracy.

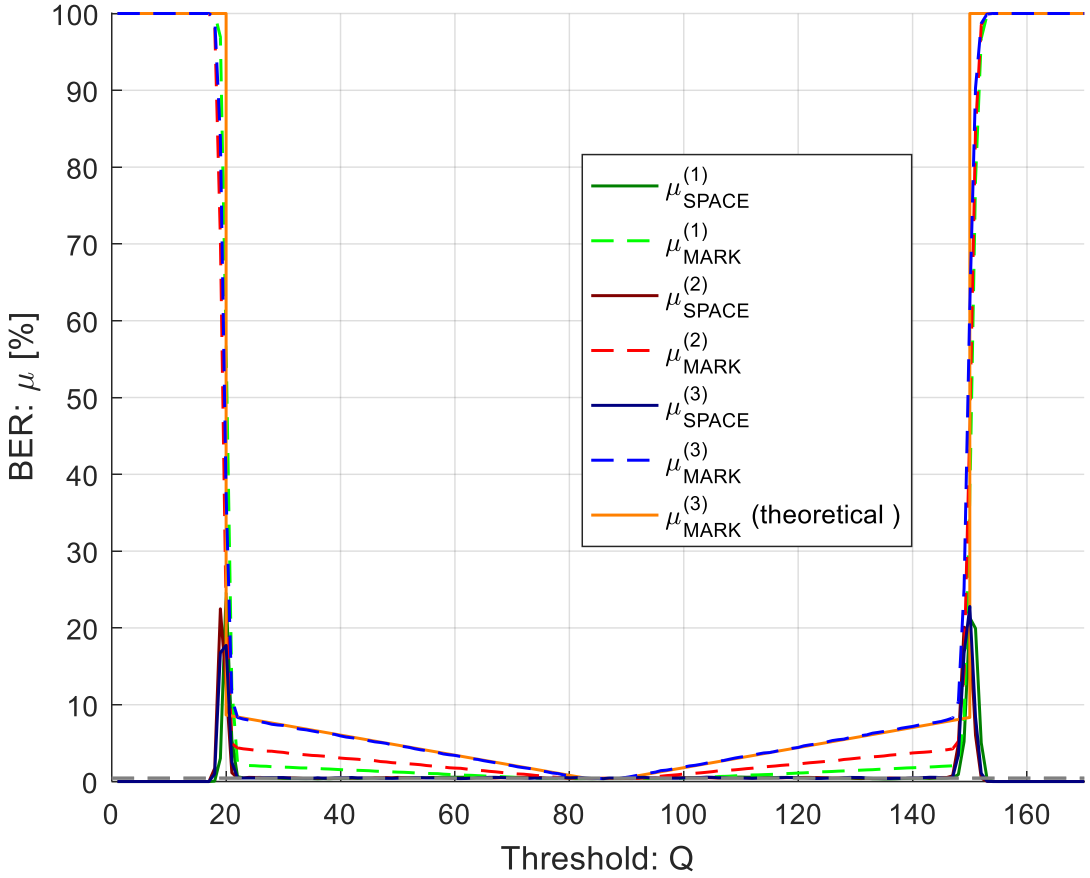


Figure 2.8 Measured BER values for constant δ and three different values of S, as a function of Q. The theoretical BER diagram for MARK symbols in experiment 3 is also shown.

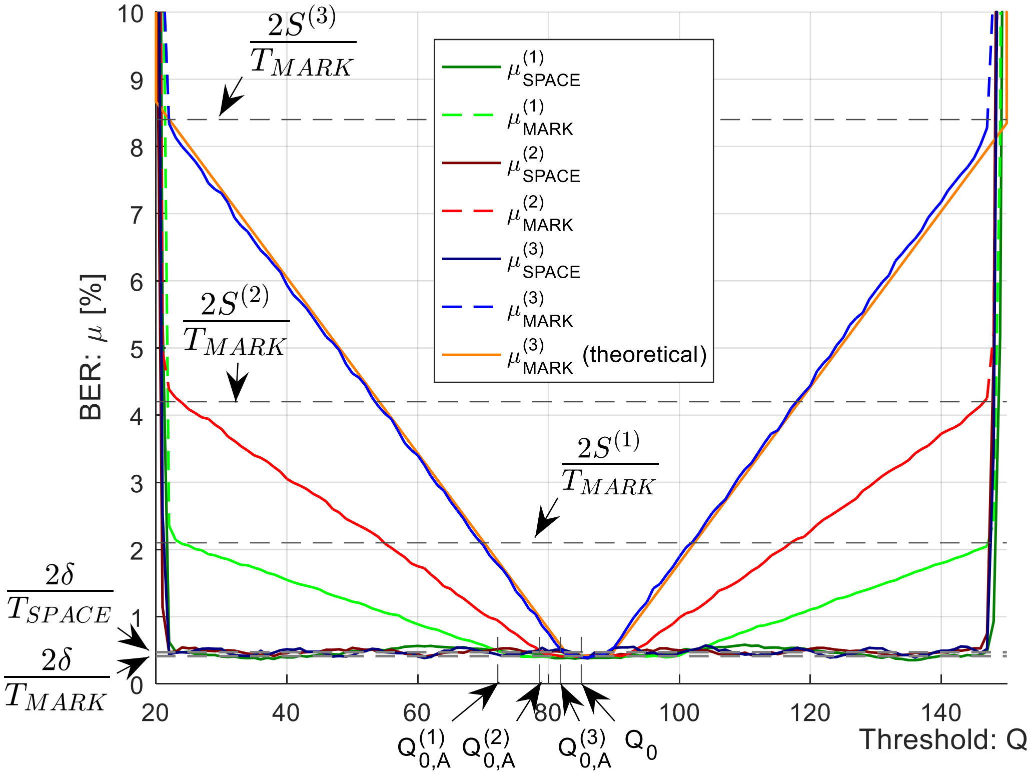


Figure 2.9 Measured BER values for constant δ and three different values of S, as a function of Q (zoomed). The theoretical BER diagram for MARK symbols in experiment 3 is also shown.

Figure 2.10 shows the case of varying slip values of $\delta_1 = 20$ µsec, $\delta_2 = 98$ µsec, and $\delta_3 = 177$ µsec, with constant $S = 200$ µsec. The figure contains the measurement results as well as the theoretical BER functions: orange and grey curves correspond to MARK and SPACE symbols, respectively. As the figure clearly shows, the measurement results fit the theoretical results.

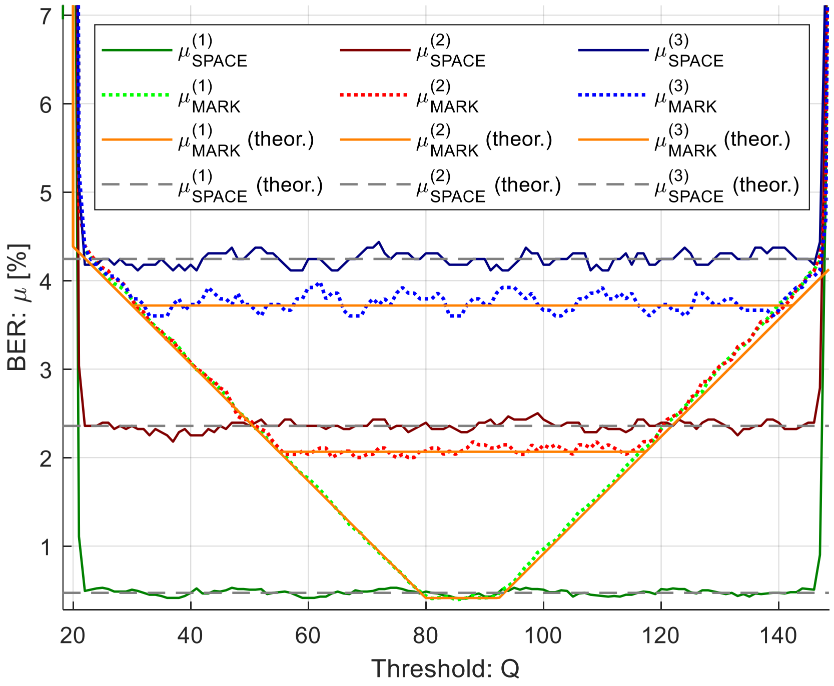


Figure 2.10 Measured BER for constant S and various values of δ, as a function of Q.

The effect of noise is shown in Figure 2.11, for $S = 200$ µsec and $\delta = 98$ µsec. The variance of the additive noise component was set to 1, 5, 10, 15, and 20 LSB (notice that the camera was used in 8-bit mode). Comparing Figure 2.11 with Figure 2.5 and Figure 2.6, the measured BER functions correspond well with the theoretical results.

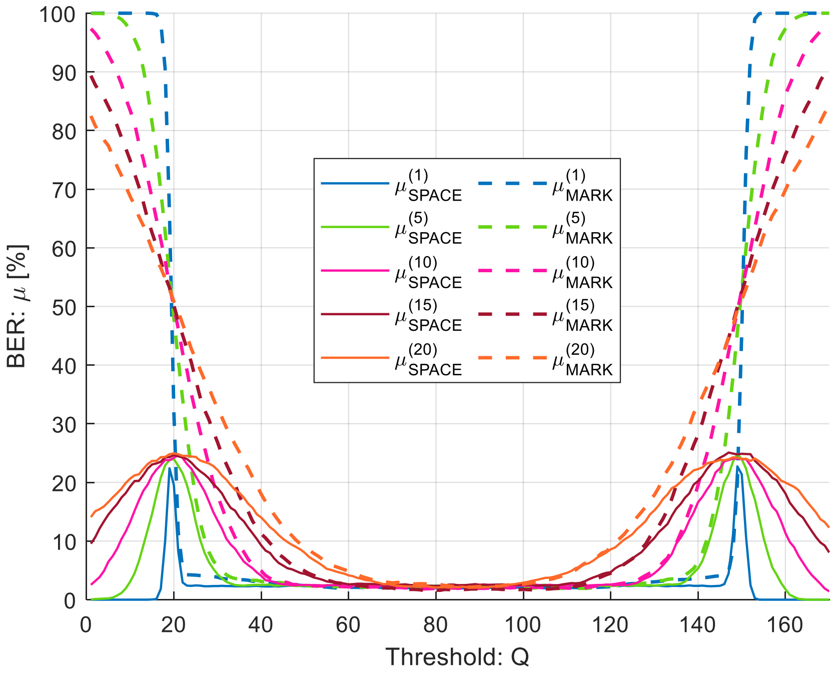


Figure 2.11 Measured and theoretical symbol error rates (zoomed).

The presented measurements are in good agreement with the theoretical results. Thus, the derived theoretical model can be used to design the parameters of the protocol, and also to predict the performance of the protocol.

2.4 Conclusion

In this section I analyzed the effect of frequency error between the transmitter and the receiver, and the effect of noise on the transmission of UFSOOK's data symbols with the proposed equivalent camera operation model, which operates with impulse sampling instead of integral sampling.

I showed that for reliable operation of UFSOOK using a global shutter camera without synchronization between the transmitter and receiver the δ sampling interval offset should be kept small to provide a higher count of good samples, while the S shutter speed should be set even smaller than δ, to avoid the suboptimal thresholding of MARK symbols. The ideal value for the Q threshold is the Q_0 signal amplitude average value, with which we can improve the transmission performance of MARK symbols in case $\delta < S$. While small δ values decrease the BER of the protocol, in practical cases a too small δ value may have a negative impact on the data transmission service when $\delta < S$, since it also extends the width (in sample counts) of dangerous intervals for the symbols, e.g., the transmission of symbols will be good for a day, but then the protocol will be unusable for a few hours.

In practical cases when we would like to tune the value of δ, the camera's sampling frequency may not be fine-tunable (i.e., only pre-defined values are supported by the device). In these cases, instead of adjusting the receiver's frequency the tuning of parameter δ can also be achieved if the transmitter's clock is adjusted, i.e., we can still modify the f_{CAM} ideal sampling frequency instead of the $\tilde{f}_{CAM}$ actual sampling frequency to get the desired δ offset.

The analysis of the effect of noise showed that low signal-to-noise ratio may have a heavy impact on the proper determination of the symbols. In practical cases the camera sensor's noise level is usually low when using small exposure time, low ISO value, and low gain, which is mainly the case when it is used in the positioning system introduced in Section 1.4, but in addition to that, achieving a high signal amplitude between A_L and A_H (but also keeping A_H below the sensor saturation level) is also advised to keep the signal-to-noise ratio as high as possible. I also showed that using Q_0 optimal thresholding value to determine the high/low transmitter states the impact of noise on the symbol transmission can be minimized.

Although setting the protocol's n design parameter to a higher value makes its modulation less flickering for the human eye, it also has a negative impact on the protocol's transmission performance, thus, lower values that already seem flicker-free for the human eye are favorable.

The protocol's data transmission performance also depends on the transmitted data due to the usage of different frequencies for SPACE and MARK symbols. Figure 2.12 shows an example of symbol transmission error regions, where $\sigma = 0$, $\delta = T_{MARK}/80$, and $S = T_{MARK}/20$. The blue and red regions mark the dangerous intervals for SPACE and MARK symbols, respectively, where the corresponding symbols will possibly fail. Each vertical gray line illustrates a time instant when a symbol is received, while its inclusion in a region also depends on the Q threshold level used at the receiver. To successfully transmit a data packet with M bits, we need M+1 consecutive symbols in the white region, where the extra symbol is the MARK symbol of the Start Frame Delimiter. E.g., if we choose threshold Q to be equal with $Q_{0,A}$, the first 17 symbols are error-free. A bad transmission occurs if the transmission of any bit in the data packet happens during a corresponding dangerous interval. In case of Figure 2.12, for logical 0 values, we have to avoid the blue region, and for logical 1 values, the red region. E.g., if we choose the optimal Q_0 as the threshold, and start at index 19, the first bit is correct in both cases (0 and 1 values), but the second bit's transfer will be faulty if its value is 1, as it would be in the red region.

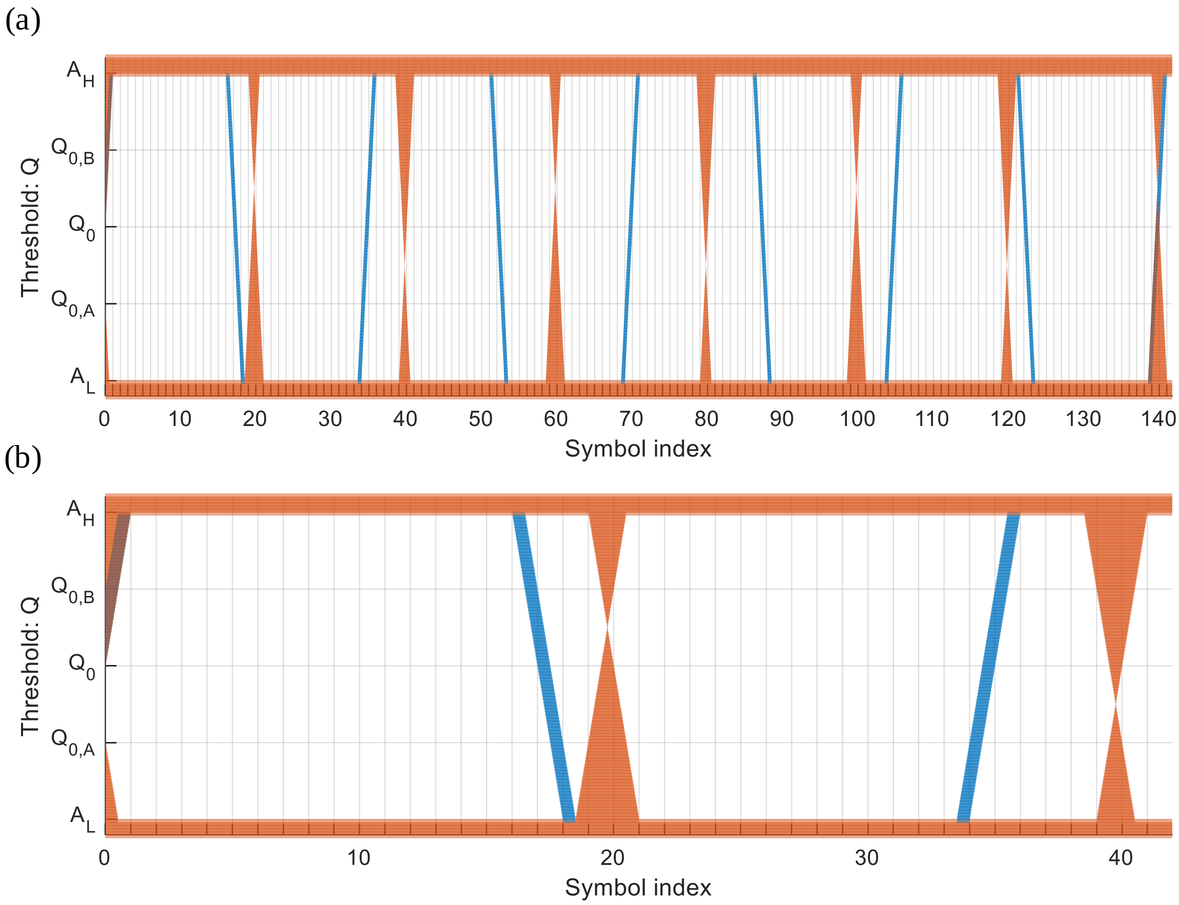


Figure 2.12 Example of data transmission errors. Dangerous intervals for SPACE and MARK symbols are marked with blue and red color, respectively. The frequency error and exposure time are set based on MARK symbol's time period: $\delta = T_{MARK}/80$, $S = T_{MARK}/20$, δ is set to 0 (noise-free). (b) is a zoomed portion of (a).

During my research, the UFSOOK protocol's packet error rate (PER) was not extensively investigated, as the protocol was abandoned relatively early in favor of the upcoming protocol presented in Section 3. However, with the provided equivalent camera operation model and error analysis the PER could be estimated for each data packet with e.g., simulations.

While it is not guaranteed that UFSOOK's data transmission will always be error-free, with the right settings, it can be used as a communication protocol to transfer the beacon IDs in a visible light indoor positioning system from time to time. One of its major drawbacks is that sometimes we have to wait for a little while to be able to correctly identify an LED beacon; the other one is that the position and width of its dangerous intervals depend on the data itself, and their width may occasionally add up, as different frequencies are used for the data symbols. Luckily, if the beacons are tracked in the indoor positioning system, the IPS relies on the VLC protocol only for the first time since their identity is preserved while they are visible for the camera sensor.

In the following section a similar OOK-based VLC protocol will be analyzed that uses phase-shifting instead of frequency-shifting, hence it uses only one frequency for data symbols making the occurrence of dangerous intervals regular. Also, the transmission data rate of the protocol is doubled compared to UFSOOK's.

3 Analysis of the UPSOOK protocol

The Undersampled Phase-Shift On-Off Keying (UPSOOK) protocol [62] is another protocol that can be used to transmit data between a camera and LED beacons, similarly to UFSOOK (analyzed in Section 2). Its operation is illustrated in Figure 3.1. UPSOOK operates similar to UFSOOK, but instead of frequency-shifting, it uses phase-shifting for MARK and SPACE symbols. The protocol also utilizes three symbols: HEADER, MARK, and SPACE, but both MARK and SPACE symbols are square wave signals with duty cycle of 50% and have the same frequency of

$$f_{data} = nf_{CAM}, \tag{3.1}$$

where n is an integer design parameter and f_{CAM} is the ideal camera sampling frequency (in the illustration of Figure 3.1 the parameter n is 3). Equation (3.1) can be rewritten to represent the relationship between the signal period T_{data} and the ideal sampling interval T_{CAM}:

$$nT_{data} = T_{CAM}. \tag{3.2}$$

The phases of MARK and SPACE symbols differ by 180º, as shown in Figure 3.1.

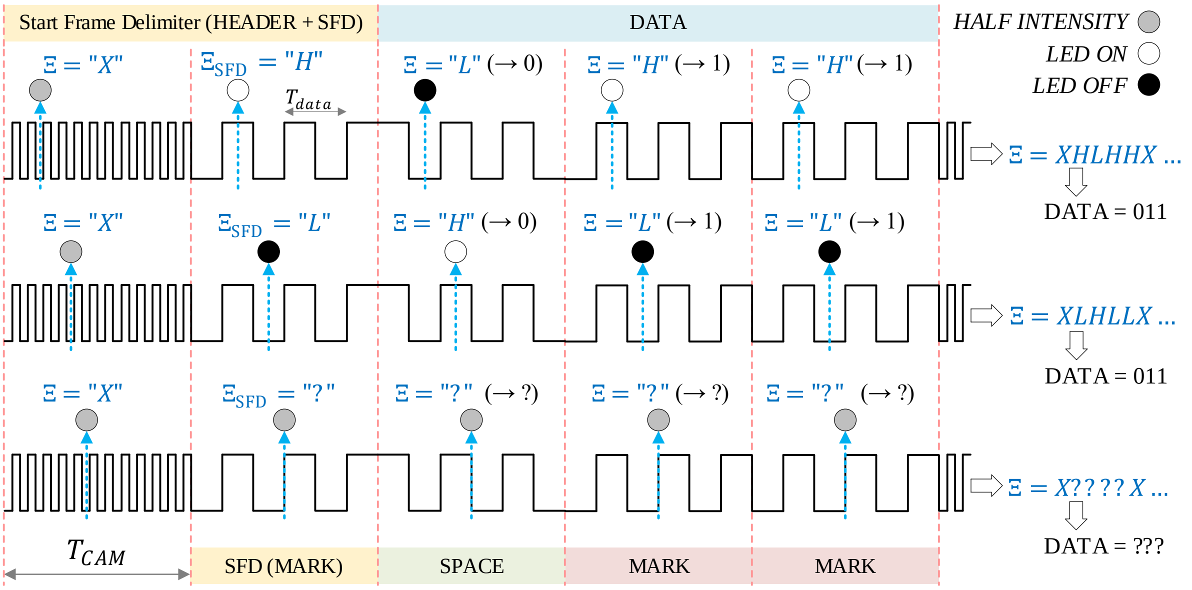


Figure 3.1 The operation of the UPSOOK protocol. The sent packet contains the Start Frame Delimiter (HEADER = X, SFD = MARK = 1), and three data bits of 011. Due to the phase uncertainty, two possible received sample sequence can be produced (1st and 2nd rows), where the ambiguity can be resolved using the SFD. When sampling is done near the edges, the outcome of the decoding is uncertain (3rd row).

Similarly, the HEADER symbol is also a square wave signal with duty cycle of 50%, and with high frequency f_{header}, as in case of UFSOOK. The signal frequencies are determined so that

$$f_{data} \ll \frac{1}{S}, \tag{3.3}$$

and

$$f_{header} \gg \frac{1}{S}, \tag{3.4}$$

where S is the camera exposure time. When (3.3) holds, then the camera senses that the signal level is either high (H) or low (L), while with (3.4) the sensed signal level X has approximately half intensity. Notice that this is true for the ideal case, but in practice, it was found that the level of X approaches the level of H, as the communication distance increases [62]. This makes decoding more difficult and limits the maximum communication distance of UPSOOK (it’s also true for UFSOOK).

The transmitted packet contains the Start Frame Delimiter (a HEADER symbol, followed by a MARK symbol), and then one symbol for each data bit. Data bits are simply coded as follows: 1: MARK, 0: SPACE. See the example of Figure 3.1 for data bit sequence of 011. Notice that although SFD is the abbreviation for the complete Start Frame Delimiter, in our case it is used as the name of the special MARK symbol after the HEADER symbol in the real Start Frame Delimiter.

In case of UPSOOK each symbol has length of T_{CAM}, thus, exactly one sample is taken from each symbol. The received symbols are denoted by Ξ. Since the transmitter and receiver are asynchronous, sampling can happen at any phase of the signal. The phase uncertainty does not affect the X symbol, due to (3.4), but it does affect the SFD and data symbols. Figure 3.1 illustrates the two possible outcomes: in the first case the received SFD is $\Xi_{SFD} =$ H and the received data symbols are $\Xi_{DATA} = LHH$ (upper row of Figure 3.1), while in the second case $\Xi_{SFD} = L$ and the data symbols are $\Xi_{DATA} = HLL$ (middle row of Figure 3.1). Using the SFD, the ambiguity of the data bit Ξ_D is resolved in the following way:

$$D = \begin{cases} 1 & if\ \Xi_{SFD} = \Xi_D \\ 0 & if\ \Xi_{SFD} \neq \Xi_D \end{cases}, \tag{3.5}$$

where D is the bit value corresponding to symbol value Ξ_D. In the example of Figure 3.1, both cases result in the correct bit sequence of 011.

The camera detects light intensity I_s, which is a value between 0 and $2^B - 1$, where B is the depth of the camera sensor (in bits). From I_s the detected symbols are generated by thresholding, as follows:

$$\Xi = \begin{cases} H & if\ I_s \geq Q \\ 0 & if\ I_s < Q \end{cases}, \tag{3.6}$$

where Q is the decision threshold.

Notice that the decoding works only if the sampling is close to the ideal circumstance shown in the first two lines of Figure 3.1. If samples are taken around the signal edges, as shown in the third line of Figure 3.1, erroneous detections can happen.

3.1 Connection between UFSOOK and UPSOOK

Both UFSOOK and UPSOOK utilize three symbols (HEADER, MARK = logical 1, SPACE = logical 0), and are decoded in a similar way, using an intensity threshold to detect ON/OFF states of the transmitter. The equivalent camera model (presented in Section 2.1) can also be used for UPSOOK.

One of the key differences between the two protocols is in the determination of the symbols. The meaning of primary (reference) and secondary LED states are exchanged, with respect to the MARK symbols and SPACE symbols. In UFSOOK, a MARK symbol is determined if the secondary sample represents the opposite LED state than the LED state in the primary sample. And a SPACE symbol is determined if the LED states are similar both in the primary sample, and in the secondary sample. In UPSOOK however, a MARK symbol is determined when the reference (SFD) and the data sample are similar, and a SPACE symbol is determined if they represent the opposite LED state.

The other key difference lies in the timing of the samples. In UFSOOK, each symbol has its own reference sample (the first sample of a symbol), immediately followed by the data’s sample (the second sample of a symbol). In UPSOOK however, all of the data samples are compared to a dedicated reference sample (SFD) preceding the data part, meaning that the inevitable time offset in practical unsynchronized use cases, caused by the frequency error, will be its multiple based on the index of the data bit.

The results of the UFSOOK analysis (presented in Section 2) can also be used for UPSOOK with respect to the differences mentioned above: the equations for the calculation of bit error rates should be exchanged (MARK ⇔ SPACE); the δ parameter should be multiplied with the index of the examined data bit. The latter implies that in UPSOOK, the bits will have different error rates inside a packet based on their position, and the bits at the end of the packet are more susceptible to errors.

3.2 Error analysis

As discussed in Section 2.4 the calculation of the packet error rate (PER) of the UFSOOK protocol is a rather complex task, as the protocol uses different transmission frequencies for its symbols. However, in UPSOOK, only one frequency is used for data transmission, which implies that the dangerous regions, where the transmission may possibly fail, are aligned. The goal of this error analysis is to estimate the worst-case BER and PER of the protocol to obtain the guaranteed transmission performance.

The first two rows of Figure 3.1 show an ideal situation when the sampling is done near the middle of the light and dark pulses. However, this cannot be guaranteed; if the transmitter and receiver are not synchronized, the sampling may be performed near the edges of the blinking signal, as shown in the 3rd row of Figure 3.1.

The outcome of the detection depends on various factors: the decision threshold, noise, jitter, frequency error. The possible error sources and their effect will be discussed in detail using the equivalent camera operation model presented in Section 2.1.

3.2.1 Threshold

If $I(t)$ is a rectangular signal with duty cycle of 50%, and there are no additional error sources, and no sensor saturation, then according to (2.4) and (2.5), $d'(t)$ is a symmetric trapezoid signal with rising and falling times equal to S, as shown in Figure 3.2 (a). Notice that the symbol coded by signal $I(t)$ can be reconstructed from $d'(t)$ by using the ideal threshold

$$Q_0 = \frac{A_H - A_L}{2}. \tag{3.7}$$

Using Q_0, the time function of the received symbol is the exact replica of that of the transmitted symbol, delayed by $S/2$, as shown in Figure 3.2 (a). Notice that the sampling time instant is at the end of the aperture window in this case, according to (2.4).

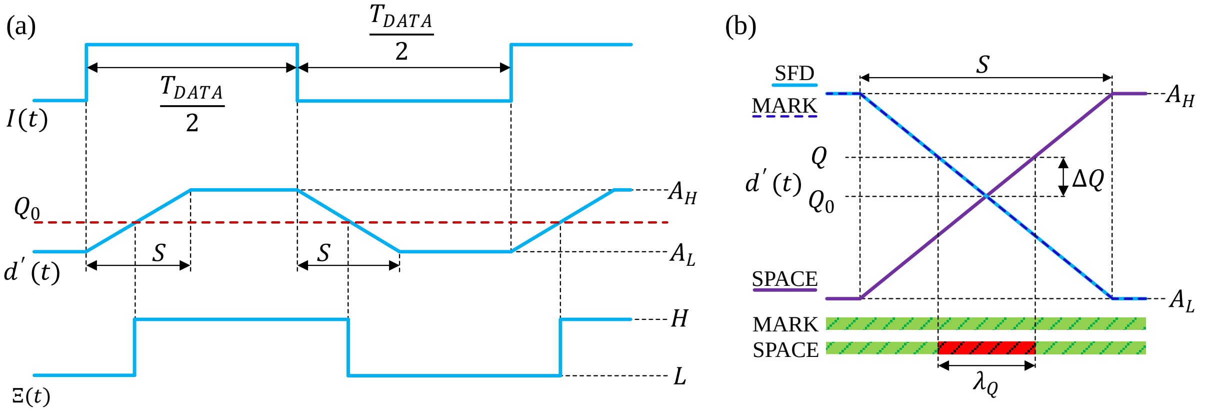


Figure 3.2 Decoding of symbols using thresholding. (a) Using ideal threshold Q_0. (b) The effect of non-ideal threshold: samples taken in the unsafe interval (red) provide incorrect symbols, while samples in the safe regions (green) provide correct symbols.

The illustration of Figure 3.2 (b) shows multiple signal segments of $d'(t)$ around signal changes, so that the shown signal segments are kT_{CAM} apart, where $k = 1,2,3, \ldots$ is integer and refers to the bit index. If (3.1) holds then all rising edges and all falling edges are precisely aligned, as shown in Figure 3.2 (b). The horizontal axis represents possible sampling instants and at the same time successive sampling instants kT_{CAM} time later. Using the figure, the values of the samples can be determined, given the sampling instant and the threshold.

As an example, Figure 3.2 (b) shows a SFD (around a falling edge, shown by a solid light blue line), a MARK symbol (around a falling edge, dashed dark blue line), and a SPACE symbol (around a rising edge, solid purple line). The ideal threshold Q_0, and the real threshold Q are also shown.

Let us suppose that the data symbol to be decoded is SPACE. In this case the samples, taken from the SFD and from the data symbol, should be different for correct detection. This is true for time instants both on the left-hand side and the right-hand side of Figure 3.2 (b). However, if samples are collected in the unsafe time interval, shown by red in Figure 3.2 (b), then both samples are considered L, resulting in incorrect detection. In Figure 3.2 (b) a falling SFD edge and a rising data edge is shown; notice that similar unsafe intervals exist on the opposite edges, too.

If the detected signal amplitude is A_L and A_H for Low (L) and High (H) states, respectively, and the difference between the actual and ideal thresholds is $\Delta Q = Q - Q_0$ then the worst-case width of the unsafe interval can be computed, using similar triangles of Figure 3.2 (b), as follows:

$$\lambda_Q = \frac{2|\Delta Q|}{A_H - A_L} S \tag{3.8}$$

If the data symbol is MARK and no other disturbances are present, then the sample of the SFD and the sample of the data symbol will be the same (see the solid light blue and dashed dark blue lines in Figure 3.2 (b)); in this case the value of Q has no effect on the decoding, i.e., for MARK symbols there is no unsafe interval. Since in a message both MARK and SPACE symbols may be present, in worst case (3.8) provides the width of the unsafe interval.

3.2.2 Noise

The sensed signal $d'(t)$ may contain additive noise, due to possible external disturbances and camera sensor noise. The effect of noise is illustrated in Figure 3.3, where the maximum noise amplitude is denoted by A_n.

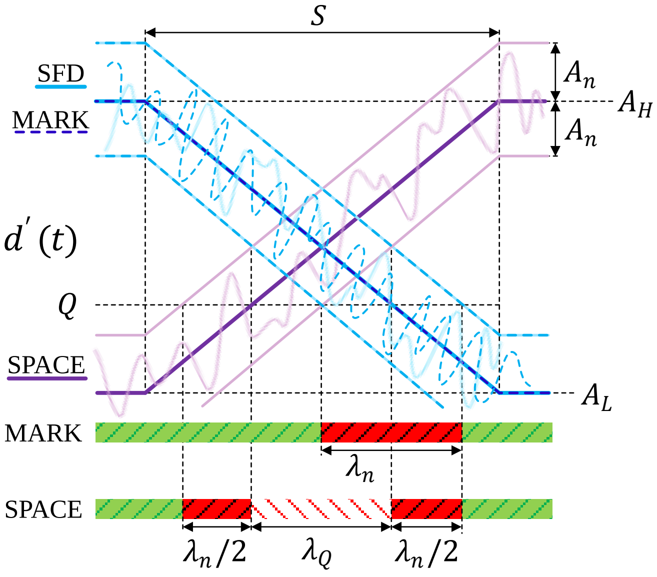


Figure 3.3 The effect of noise to symbol decoding. Samples taken in the red unsafe intervals may provide incorrectly decoded symbols, while samples in the safe regions (green) provide correct detections.

For MARK symbols the width of unsafe interval, shown by a red line (for the case of no saturation), using notations of Figure 3.3, is the following:

$$\lambda_n = \frac{2A_n}{A_H - A_L} S \tag{3.9}$$

According to the results of Section 3.2.1, for SPACE symbols there may exist an unsafe interval, due to inaccurate thresholding, denoted by a striped white-red interval in Figure 3.3. The additive noise widens this interval on both sides, as shown by red intervals in the figure. The width of the red intervals on both sides are equal to $\lambda_n/2$, where λ_n is defined in (3.9). Thus, the total increase of the unsafe interval, due to noise, equals to λ_n for both MARK and SPACE symbols. The usage of maximum noise amplitude is sufficient for the worst-case scenario.

3.2.3 Jitter

Both the transmitter and the camera may have jitter. The effect of the cumulative jitter is illustrated in Figure 3.4.

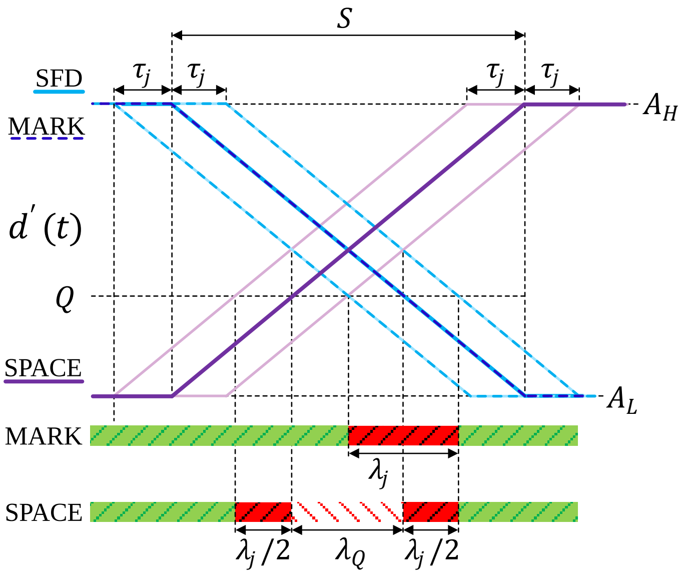


Figure 3.4 The effect of jitter to symbol decoding. Samples taken in the red unsafe intervals may provide incorrectly decoded symbols, while samples in the safe regions (green) provide correct detections.

Let us use the SFD as reference; in the presence of jitter, the edges of SPACE and MARK symbols may arrive earlier or later than in the ideal case, thus, the sample taken in the unsafe interval may or may not be correct. If the maximum jitter is τ_j then the width of the resulting unsafe interval for MARK symbols is the following:

$$\lambda_j = 2\tau_j \, . \tag{3.10}$$

For SPACE symbols, very similarly to the case of noise, jitters enlarge the unsafe interval of inaccurate thresholding. The total growth of the unsafe interval is equal to λ_j of (3.10), as shown in Figure 3.4. Thus, for both MARK and SPACE symbols the effect of jitter is characterized by (3.10).

3.2.4 Frequency error

Ideally, the camera's sampling interval T_{CAM} is an integer multiple of the blinking period length T_{data}, according to (3.2). The validity of (3.2) can be ensured if the transmitter and receiver are synchronized. In practical cases, where no time synchronization is applied, there always is a small frequency error between the transmitter and the camera, i.e., instead of (3.2), similarly to (2.6), the following holds:

$$\tilde{T}_{CAM} = nT_{data} + \delta_{CAM} \,, \tag{3.11}$$

where δ_{CAM} can be considered as the error of the camera's sampling interval when the transmission frequency is accurate (see (2.6) and (2.7) in Section 2.1). If δ (i.e., the size of δ_{CAM}) is not zero, the phase of the sampling instants is shifted at each sample, the effect of which is shown in Figure 3.5 for the kth data bit.

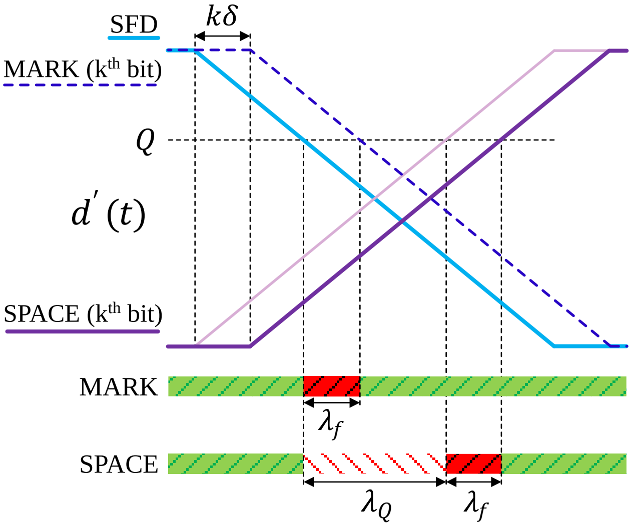


Figure 3.5 Effect of frequency error. SFD: solid light blue, SPACE: solid purple, MARK: dashed dark blue lines. Samples taken in the red unsafe intervals provide incorrectly decoded symbols, while samples in the safe regions (green) provide correct detections. The light purple line shows the SPACE symbol with no frequency error.

Notice that during decoding each sample is compared to the SFD, according to (3.5). Thus, the reference is the SFD sample, and each of the successive sampling times of the data bits are shifted by δ, the total delay being $k\delta$ at the kth data bit, as shown in Figure 3.5. Thus, compared to the reference SFD, the first bit is sampled with a δ time offset; the delay of the second bit is 2δ, etc., generally the kth bit is sampled with a $k\delta$ time offset. Thus, the unsafe interval for the kth bit, due to frequency error, for both the SPACE and MARK symbols is the following:

$$\lambda_f = k\delta \tag{3.12}$$

3.3 Performance properties

In worst case, the cumulative effect of the above error sources can be represented as a combined unsafe interval with width of λ, around the edges of the transmitted signal, as shown in Figure 3.6. In worst case the width of the combined unsafe interval is the following for MARK and SPACE symbols in case of a non-saturated camera:

$$\lambda_{MARK} = \lambda_n + \lambda_j + \lambda_f \,, \tag{3.13}$$

$$\lambda_{SPACE} = \lambda_Q + \lambda_n + \lambda_j + \lambda_f \,. \tag{3.14}$$

For general (not symbol-based) worst case estimation (3.14) should be used (as it is more sensitive to error sources).

When the frequency synchronization between the transmitter and the camera is perfect, according to (3.2), the sampling is always performed at the same phase of the transmitted signal, thus, the sampling is either always good (green arrows of Figure 3.6 (a)), or always possibly bad (red arrows of Figure 3.6 (a)). In practical cases there always is a small frequency error between the transmitter and the camera, i.e., (3.11) holds.

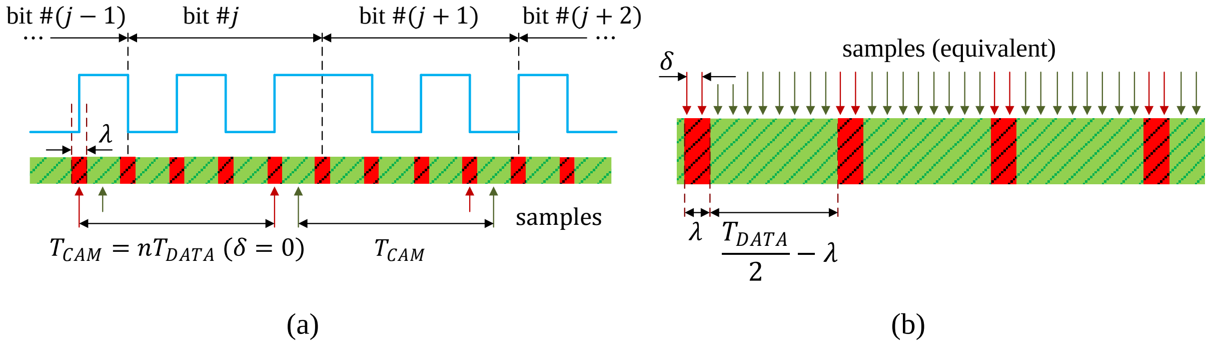


Figure 3.6 (a) Unsafe intervals with width of λ, as a cumulative results of various error sources. Sampling is made with $\delta = 0$. Red arrows: samples taken in unsafe intervals, dark green arrows: samples taken in safe intervals. (b) The equivalent sampling of the signal with equivalent time interval of $\delta > 0$. Samples are taken from safe and unsafe intervals as well. The unsafe and safe intervals are red and green, respectively.

If δ is not zero, the phase of the sampling instants is shifted at each sample, as shown in Figure 3.6 (b). When $\delta_{CAM} > 0$, the equivalent samples are taken from left to right, while in case of $\delta_{CAM} < 0$, from right to left. In Figure 3.6 (b) the safe and unsafe intervals are shown by green and red colors, respectively, and the equivalent sampling instants are denoted by colored arrows. The interpretation of the figure is the following: If a sample is taken at a particular phase of the signal (e.g., the leftmost red arrow of Figure 3.6 (b)) then the next sample will be taken δ time to the right (second red arrow in the middle of the unsafe interval in Figure 3.6 (b)), the next sample is taken further δ time to the right (third, green arrow at the beginning of the safe interval in Figure 3.6 (b)), etc. The samples traverse along the alternating safe and unsafe intervals; thus, the sampling is performed in the safe interval for a while, then in the unsafe interval, then again in the safe interval, etc.

The average number of samples in the unsafe region is

$$N_{unsafe} = \frac{\lambda}{\delta}, \tag{3.15}$$

while in the safe region the average number of samples is

$$N_{safe} = \frac{\frac{T_{data}}{2} - \lambda}{\delta} = \frac{T_{data}}{2\delta} - \frac{\lambda}{\delta} = \frac{T_{data} - 2\lambda}{2\delta}. \tag{3.16}$$

Thus, the transmission is potentially broken periodically with period

$$T_P = \tilde{T}_{CAM}\left(N_{unsafe} + N_{safe}\right) = \tilde{T}_{CAM}\frac{T_{data}}{2\delta}, \tag{3.17}$$

and the length of the potentially bad interval is

$$T_{unsafe} = \tilde{T}_{CAM}N_{unsafe}. \tag{3.18}$$

While the sampling is done in the unsafe interval, the protocol is likely to provide incorrect detections, thus, the protocol may not be responsive for time interval T_{unsafe}. Thus, the responsiveness in worst case is characterized by T_{unsafe}.

To calculate the worst-case PER for a particular data transmission, we can calculate the λ value for each bit using (3.13) and (3.14) based on the bits' value and position in the packet. Then, the maximum value of λ in the packet is sufficient to calculate the PER estimate with the above equations for a worst-case scenario, since even a single bit error can cause a bad packet, and the position of the unsafe intervals are aligned. To calculate the absolute worst case PER for the entire protocol, we may use the last bit (index) and assume, that it is a SPACE symbol (logical 0).

To estimate the worst-case BER for MARK and SPACE symbols in UPSOOK, we only need to divide (3.13) or (3.14) with $T_{data}/2$, similarly as in the case of UFSOOK. To estimate the average BER, the results of Section 2 can also be used for this protocol, (e.g., with the estimation method (2.39) presented in Section 2.2.2, which supports the noise, suboptimal threshold, and frequency error components) with exchanged rule set for the symbols (MARK ⬄ SPACE), common symbol time T_{data}, and proper phase offset. Notice, that due to UPSOOK's encoding-decoding method, each bit will have a different error rate if frequency error is present in the system, and δ must be adjusted accordingly with the usage of $k\delta$, where $k = 1,2,3, \ldots$ is the index of the examined bit.

3.4 Evaluation

The responsiveness of the UPSOOK protocol was tested in laboratory settings in ideal circumstances. In the test setup 9-bit packets were transmitted repeatedly. Altogether 45000 bits (5000 packets) were sent, using $T_{data} = 1/(n * 30)$ sec and $\tilde{T}_{CAM} = 1/30 + \delta$ sec, where $n = 4$, $\delta = 0.2$ µsec and $S = 250$ µsec. The detected signal amplitude was between $A_L = 7$ and $A_H = 109$. The standard deviation of the measurement noise was $\sigma_{noise} = 0.6$ LSB.

To illustrate the behavior of UPSOOK, a test was run with various values of Q. The transmission status of the protocol for $Q = 50$ is shown in Figure 3.7.

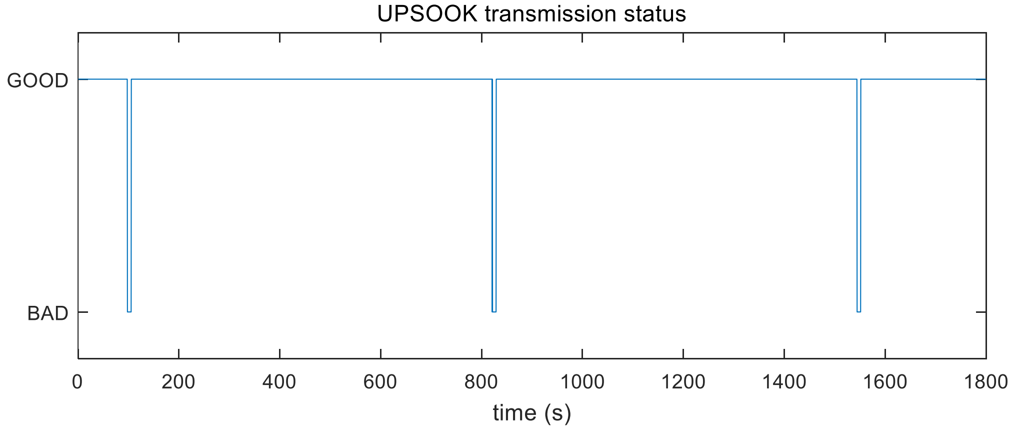


Figure 3.7 Measured transmission status of the UPSOOK protocol, with $S = 250$ µsec, $A_L = 7$, $A_H = 109$, $Q = 50$ and $\delta = 0.2$ µsec.

As shown in the record, the protocol fails periodically, where $T_P \cong 12$ min, and the width of the unsafe intervals was $T_{unsafe} \cong 7$ sec. During these unsafe intervals the system is not responsive: in worst case the detection of a beacon is delayed by ~7 seconds.

Table 3.1 contains measured and theoretical values for T_P and T_{unsafe}, for $Q = 20, 50$ and 58. The theoretical values were calculated for the entire protocol's worst-case scenario. Let us examine the theoretical values for $Q = 50$. The results show that in worst case, we should expect that the protocol will fail in every 694 s, and its transmission is erroneous for 8.5 s. The measured values show a better case: the occurrence of transmission errors was less common, and their width was also less (with 1.5 s). The measured and theoretical values show good correspondence, validating the theoretical results, which can be used to calculate the guaranteed operation performance of the protocol.

Q	Theoretical		Measured	
	T_P	T_{unsafe}	T_P	T_{unsafe}
20	694 sec	33.5 sec	729 sec	30 sec
50	694 sec	8.5 sec	718 sec	7 sec
58	694 sec	1.8 sec	721 sec	1 sec

Table 3.1 Theoretical results and measured performance properties of the UPSOOK protocol

3.5 Conclusion

I presented the in-depth analysis of how and when do the various error sources affect the performance properties of the UPSOOK visible light communication protocol: improper thresholding, noise, jitter, and frequency error between the transmitter and the receiver. I showed that these error sources all enlarge the width of the periodically repeating unsafe intervals, where the communication possibly becomes faulty. With these results one can pre-calculate the guaranteed data transmission rate even before the system is put into operation, and adjust its parameters to achieve the best possible outcome.

The key benefits of utilizing UPSOOK over UFSOOK are that it can transmit data twice as fast, and also the distribution of unsafe intervals is more uniform since the protocol uses only one frequency for both SPACE and MARK symbols. The design rules of UPSOOK are mainly the same as UFSOOK's: a high signal amplitude is favorable, so that it provides a wide range for optimal thresholding, while also keeping a high signal-to-noise ratio; the design parameter n should be kept minimal, to provide large safe intervals, and should only be increased to the point where the flickering is not disturbing for the human eye anymore; the smallest possible exposure time, with which the transmitters can still be sensed with a good signal amplitude, is favorable for the communication to decrease the width of the signal transition parts, in practice, setting a small gain is still better than having a large exposure time; using stable clocks both for the transmitter, and for the receiver minimizes the effect of jitter, which does not depend on any other parameters; the δ parameter should be set according to the requirements of the actual use-case, for example, in case of the indoor positioning system introduced in Section 1.4, the δ parameter was fine-tuned, so that after several minutes of reliable operation the delay for the next successful beacon ID transmission (i.e., the length of the unsafe interval) was under a second.

Technically, both UFSOOK's (see Section 2) and UPSOOK's decoding depends on the received brightness level of a single, blinking light source modulated according to the protocol's encoding and transmitted data (in case of the aforementioned indoor positioning system, only the beacon IDs are transmitted, repeatedly). Due to the fact that these protocols contain dark (low) transmitter states, the trackability of the transmitters is poor. The trackability property of the LED beacons will be investigated in the next section (Section 4), which plays a crucial role for the IPS presented in Section 1.4.

4 Novel TUPSOOK VLC protocol

When using VLC protocols for camera communication with rolling shutter cameras the transmitted code appears in the form of fringes in the camera's image frames [70]. The usage of rolling shutter cameras also makes it possible to decode the transmitted data from just one image frame, as the sampling is done per row (or column). However, the size of the LED beacon needs to be sufficiently large to encompass a whole packet within the image frame, significantly restricting the practical usability distance.

Global shutter cameras, on the other hand, sample the entire image simultaneously: the LED will blink in the consecutive camera image frames instead of becoming fringy. Also, the decoding of the VLC protocols mentioned in Sections 2-3 is based on the sensed brightness levels of the transmitters. which in case of the indoor positioning system presented in Section 1.4 is hard to implement. When the target camera is static, the image of the blinking LED beacon is at the same position of the detected image stream, making it possible to decode the data. However, in case of a moving camera (or beacon), the image of the LED is also moving in the picture. In order to be able to detect the transmitted code, the position of the LED beacon must be known in each image frame. This requirement will be referred to as trackability requirement. Figure 4.1 shows the modulation scheme of various VLC protocols.

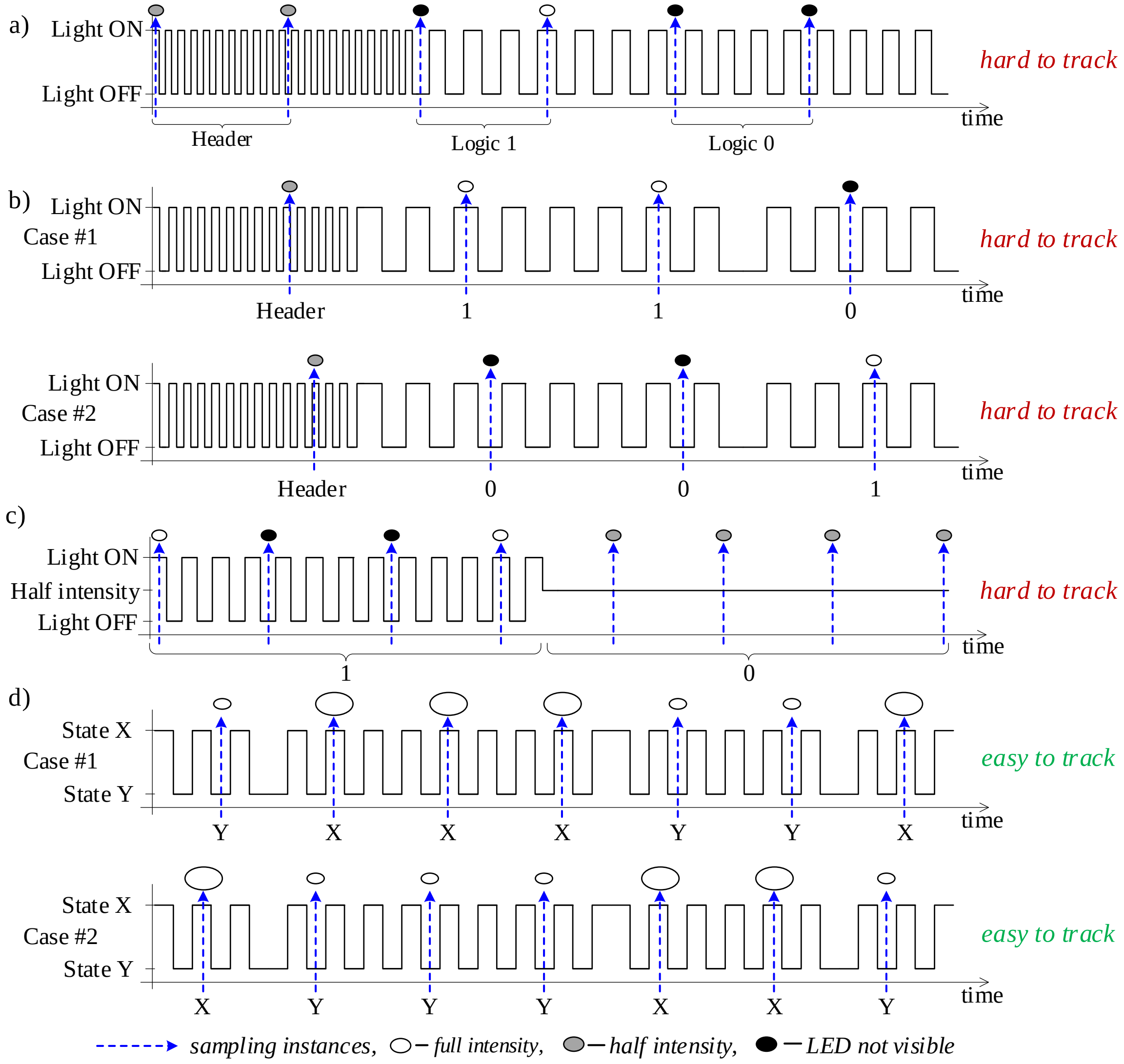


Figure 4.1 Various VLC modulation schemes for undersampled camera communication, (a) UFSOOK, (b) UPSOOK, (c) LookUp, (d) TUPSOOK

Since the LED is blinking, the image may contain various detections, depending on the utilized coding scheme and the actual code. For illustration purposes let us consider transmitted code segment 1000. For UFSOOK, the detected sequence may be the following (see coding in Figure 4.1 (a)): (OFF-ON)-(OFF-OFF)-(OFF-OFF)-(OFF-OFF), thus the beacon may not be visible for 6 consecutive frames (in parentheses detections corresponding to one bit are shown). In case of UPSOOK, it may happen that the detection is the following (see Figure 4.1 (b), case #2): (ON)-(OFF)-(OFF)-(OFF). In this case the beacon is not visible in 3 consecutive frames. The LookUp [58] coding for bit 1 allows several similar detections before the required change: in the illustration of Figure 4.1 (c)) the detection for bit 1 is (ON-OFF-OFF-ON), where the beacon is not visible in 2 consecutive frames. In practice, this number can occasionally be even higher.

All of these modulation schemes (Figure 4.1 (a)-(c)) can produce (in worst case) several consecutive frames with blank (dark) image, where the position of the LED cannot be measured. If the target makes a significant move during these frames, the tracking of the LED and the decoding of the transmitted code may become impossible. The more frames the beacon is not visible in, the slower motion can safely be tracked. Thus, for simultaneous tracking and decoding of beacons with a moving camera the optimal choice is when the beacon is visible in all images.

It is worth noting, that there exist other VLC solutions that make an LED beacon trackable in consecutive image frames. In [74] Color-Shift Keying (CSK) was used to increase the transmission data rate. The usage of color coding may also make an LED beacon trackable, for example, when the LED consistently emits a dedicated color (e.g., green), and the encoding is performed using other colors (e.g., red and blue). I dismissed the idea of investigating the usage of CSK for my indoor positioning system for the following reasons: the LED beacons would not only blink but also emit different colors based on their transmitted data, making the beacons impractical for illuminating an environment; the transmission distance would be lower due to the Bayer pattern of most color image sensors (although this problem may be solved by using a Foveon-type color image sensor, where a physical pixel contains all three color components [75]); the intensity of the LED's color components would need to be fine-tuned per image sensor so that all color components can be sensed in the same range (with normalized intensity).

In the following a modified UPSOOK-based modulation scheme will be introduced, which provides good trackability.

4.1 Protocol design

The key elements of the Trackable UPSOOK (TUPSOOK) are the following (see Figure 4.1 (d) for illustration and Figure 4.2 for detailed explanation):

- The original ON-OFF key (LED ON and LED OFF) is replaced by symbols X and Y, where the size of X and Y states is different in the image (instead of the difference in sensed brightness level), which is achieved with a special transmitter (LED light fixture) design. This way the LED can always be visible in the image, providing ideal trackability. The price of the modified keys is the somewhat decreased communication range, in which the size differences can still be detected.
- For increased robustness, bits are Manchester-coded, using pairs of states to code each bit: bit 1 is coded as X-Y, while bit 0 is coded as Y-X. The coding allows the detection of incorrect detections at a price of decreased communication bandwidth: the maximum data transmission rate is $f_{CAM}/2$, ignoring the header, similarly to UFSOOK.
- The original header (high frequency blinking, sensed as a half-intensity signal) is replaced by a special series of symbols, which cannot occur among the data symbol series (Y-X-X-X-Y, 2.5 bits long). The header plays both the role of frame delimiter and MARK symbol. The main advantage of the proposed header is that it avoids the utilization of a special symbol, the detection of which may depend on environmental factors. See e.g., the measurement results in [62], showing that the HEADER and ON symbols become very similar as the distance increases between the transmitter and the receiver. The price of increased robustness is the somewhat increased frame length: while in UPSOOK the header

and mark symbols occupy 2 symbols, in TUPSOOK the length of the combined header is 5 symbols.

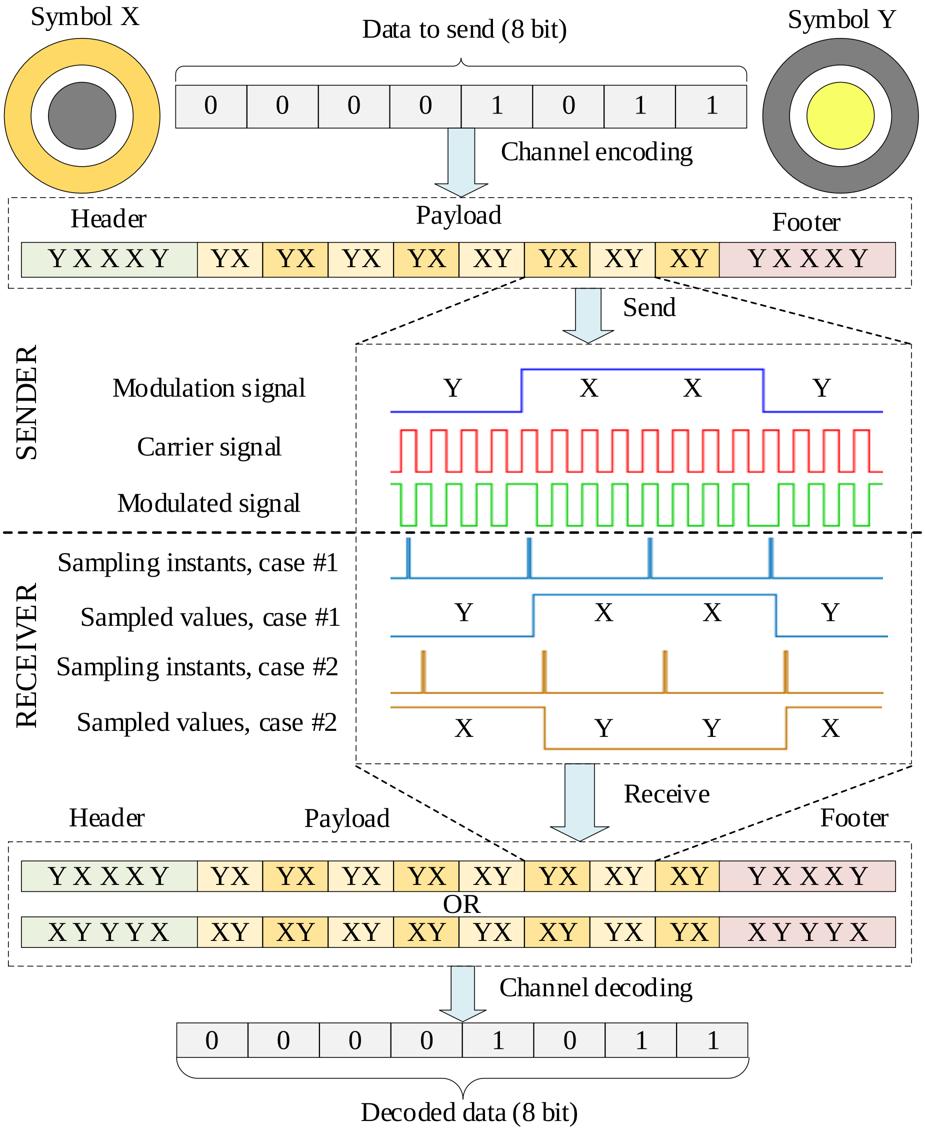


Figure 4.2 Coding and decoding of TUPSOOK

The proposed transmitter can be seen in Figure 4.3. Symbol X is represented by the outer ring ON and the inner ring OFF, while symbol Y is coded as the outer ring OFF and the inner ring ON.

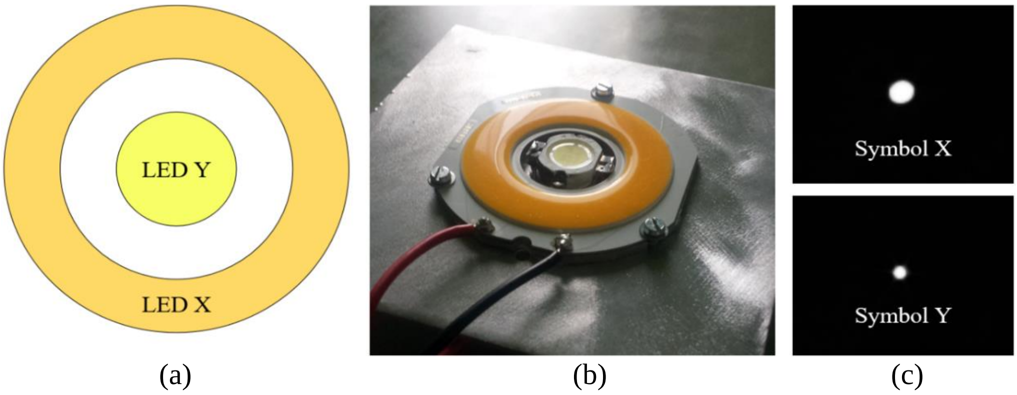


Figure 4.3 LED transmitter (a) beacon containing an outer ring LED (symbol X) and an inner dot LED (symbol Y), (b) photo of the beacon, (c) detected symbols on the camera image

Figure 4.3 also shows the detected symbols on the camera image: it is clearly visible that symbol X is larger than symbol Y. Notice the important fact that the beacon is visible on each camera frame, regardless of the transmitted symbol.

The coding and decoding process is illustrated in Figure 4.2. First the data is Manchester-coded: bit 1 is represented by XY, while bit 0 is coded by YX symbol pairs. The payload is surrounded by the header/footer segments, coded by symbol series YXXXY. (Notice that a footer segment is also the header of the next payload segment, the name footer is used only for convenience.) The channel coding is performed by phase modulation: symbol X shifts the carrier by phase 0, while symbol Y shifts the carrier by or 180º.

The sampling is performed by the camera, detecting bright blobs in the image which are possible LED beacons. On each image symbols X and Y are detected, as will be described in Section 4.2. When a header is found, the payload and the footer is also stored in a buffer. Similar to UPSOOK, the sampling may reproduce either the transmitted symbol series (case #1) or its inverted form (case #2), depending on the actual relative phase of the receiving camera and the transmitting beacon. The ambiguity is resolved using the received header: in case series YXXXY is received, the payload is decoded using XY→1 and YX→0, while for received header XYYYX the decoding is made using XY→0 and YX→1.

Error checking is performed as follows: the received packet is marked faulty if

- the header and footer are different or
- the header/footer pattern is not YXXXY or XYYYX or
- any data bit, i.e., symbol pair in the data region, is XX or YY.

The error detection scheme provided by the above channel coding replaces the repeat coding used e.g., in [77].

4.2 Simultaneous beacon tracking and detection

The tracking and decoding infrastructure is illustrated in Figure 4.4. On each camera frame, after pre-processing and conditioning it, blobs are detected, the center position and size of which are measured. Notice that blobs, corresponding to beacons, are visible on each image frame, due to the applied coding and beacon design. The tracking is performed as follows:

The t*racked blob set* at time instant $k-1$ is $B_{k-1} = \{b_1, b_2, \dots, b_{N_{k-1}}\}$, based on the acquired image samples up to time instant $k-1$. Tracked blob b_i contains the following set of information:

- Kalman filter parameters, describing the observed motion of the blob;
- the last size s_{k-1} and the current size s_k of the blob,
- number of consecutive frames N_{INVIS} where the blob was not seen,
- the last Ξ_{MSG} decoded symbols $\{SIM_k, SIM_{k-1}, \dots, SIM_{k-k_{MSG}+1}\}$, where k_{MSG} is the total number of symbols a message contains (including the header, the payload, and the footer), and
- the detected ID of the corresponding beacon (if any).

At time instant k, a set of d*etected blobs* $A_k = a_1, a_2, \dots, a_{N_k}$ are detected. From B_{k-1} and A_k the new tracked set B_k is generated, as follows:

- For each tracked blob b_i the predicted position $\hat{p}_i$ is calculated using the Kalman filter. Notice that for each detected blob a separate Kalman filter is operated, using a constant speed model.
- Using the predicted position $\hat{p}_i$ of tracked blob b_i, and the measured position p_j of detection a_j, a matching is calculated (see block blob pairing in Figure 4.4): for each b_i at most one a_j is assigned, and for each a_j at most one b_i is assigned such that the sum of distances $\sum|p_j - \hat{p}_i|$ be minimal, with the constraint that all pairwise distances

are below a limit: $|p_j - \hat{p}_i| < d_{max}$. Thus, for each tracked blob at most one detected blob (presumably the one, which is closest to it) is assigned, as the latest track element.

- Some tracked blobs b_i have assigned detected blob s (see e.g., tracked blob b_2 and detected blob a_1 in Figure 4.4). For these tracks $N_{INVIS}(i) = 0$ is set.

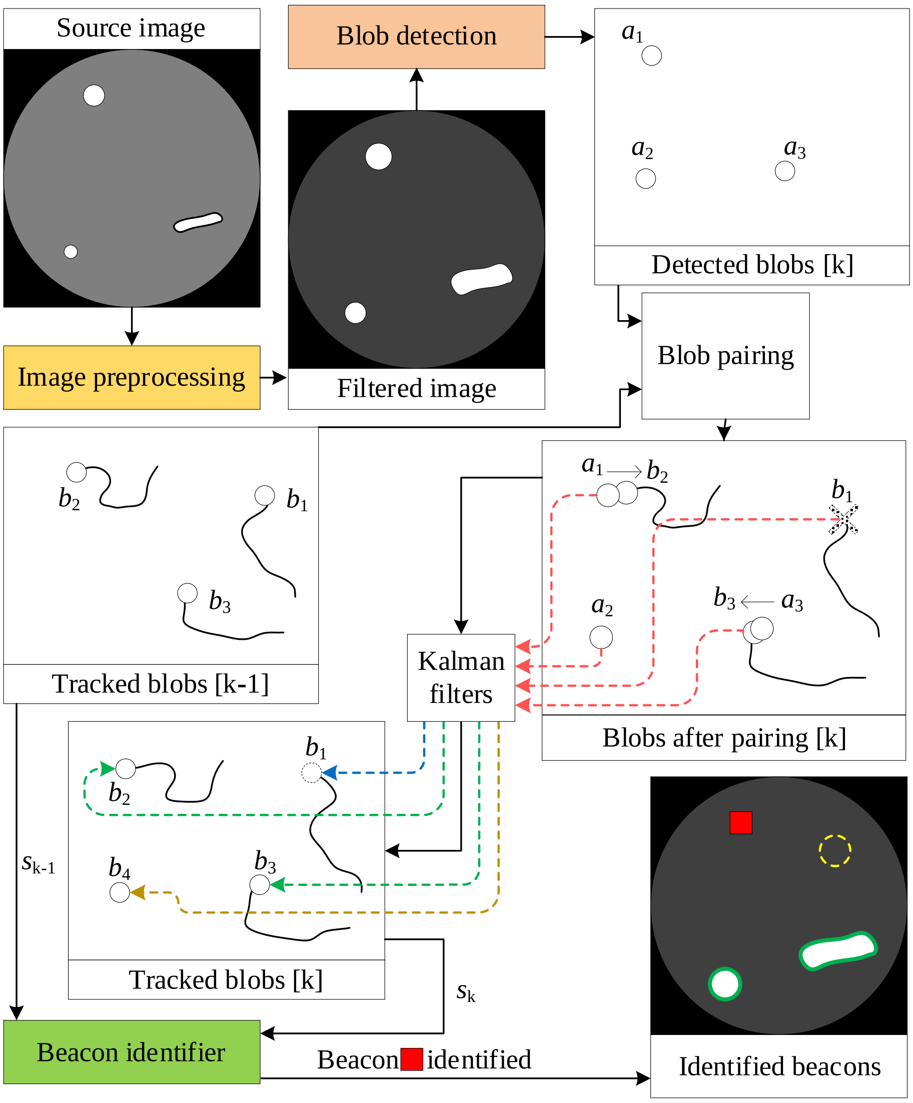


Figure 4.4 Simultaneous tracking and symbol detection

- Some tracked blobs b_j may not have a detected blob assigned (e.g., a tracked beacon is now covered by an obstacle, see e.g., tracked blob b_1 in Figure 4.4). For these blobs $N_{INVIS}(j) = N_{INVIS}(i) + 1$.
- Some detected beacons a_k may not have an assigned track (e.g., a previously unseen beacon appeared, see detected a_2 in Figure 4.4). For these detections a new tracked blob is created (b_4 in Figure 4.4).
- Tracked blobs with high $N_{INVIS} > N_{INVIS\_MAX}$ are purged.

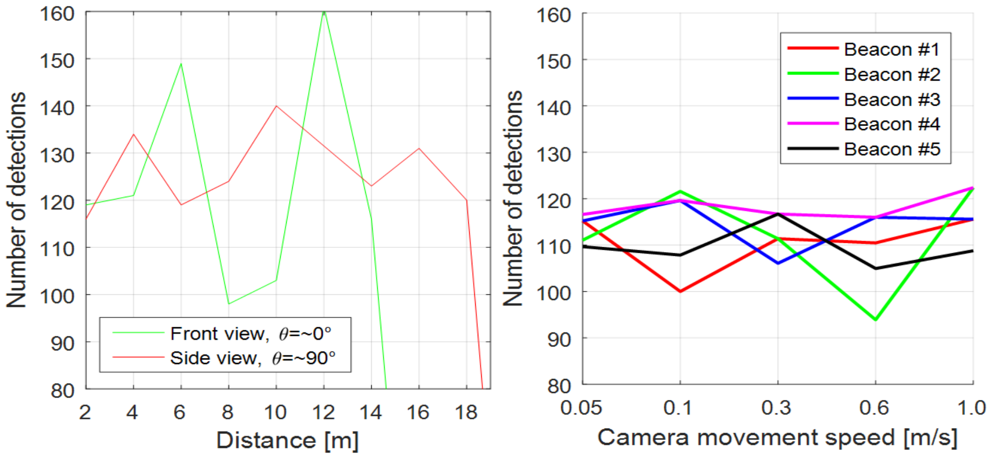
The symbol detection is performed for each frame, for tracked blob b_i blob sizes s_{k-1} and s_k are compared:

- if $s_{k-1} > \gamma s_k$ then $SIM_k = Y$, where γ is a constant parameter
- else if $\gamma s_{k-1} < s_k$ then $SIM_k = X$,
- otherwise $SIM_k = SIM_{k-1}$.

4.3 Evaluation

The maximum communication distance was tested in two scenarios, using a global shutter camera equipped with fisheye lens, with $f_{CAM} = 80Hz$, $f_{data} = 640Hz$, and with parameter $\gamma = 1.25$. In scenario 1 the camera was facing towards the LED beacon (i.e., the LED image appeared at the center of the picture), while in scenario 2 the LED was placed on the horizon of the camera (i.e., the LED appeared at the edge of the picture). The number of successfully received messages per minute are shown in Figure 4.5 (a). In both cases $14m$ was safely covered. Notice that TUPSOOK has smaller communication distance than techniques using purely ON-OFF techniques (e.g., LookUp can provide 50m communication range).

The trackability of TUPSOOK was also tested with 5 different beacons. The camera was moving with different speeds, and the number of received messages per minute is shown in Figure 4.5 (b). As the results show, the system is able to provide simultaneous tracking and detection for speeds as large as 1m/s, without significant change in the system performance.



(a) (b)

Figure 4.5 Number of successful detections per minute, (a) as a function of distance, (b) as a function of camera speed

Notice that the number of detections is not consistent (as shown in Figure 4.5), which is mainly caused by the constantly changing phase between the receiver and transmitter as discussed during the analysis of frequency error for UPSOOK in Section 3.2.4. The beacon is detectable from a longer distance when placed on the horizon of the camera, due to the better azimuthal angular resolution (e.g., if the beacon can be seen at a 2° angle by the camera, the beacon's image contains more pixels along the horizon of the camera).

4.4 Conclusion

With the proposed beacon design and symbol encoding the transmitter will always be visible in line-of-sight case, thus TUPSOOK fulfills the trackability requirement as opposed to the other mentioned solutions when used with global shutter cameras. The tracking of moving transmitters makes it possible to decode the transmitted data, even when the receiver or the transmitter is moving.

In case of the indoor positioning system introduced in Section 1.4, each beacon transmits its own unique identification number continuously. Since tracking enables the retention of the identity of moving objects in the camera's image frames, the VLC protocol is crucial mainly for the initial detection. Therefore, the only drawback arising from the unsafe intervals of the protocols is their potential to delay the detection of the beacons.

In the next section a novel, robust version of UPSOOK will be introduced, which eliminates the existence of unsafe intervals on a wide range of system parameters.

5 Novel RUPSOOK VLC protocol

The visible light communication protocols introduced in Sections 2-4 all suffer from frequency errors, causing phase slippage, due to the practical case of unsynchronized transmitters and receivers, as proved in Section 2.2.1 and in Section 3.2.4. Even if we would try to fine-tune the frequencies to have a small δ frequency error, that would not only cause the enlargement of safe transmission segments, but would also enlarge the unsafe transmission segments (e.g., the data transmission would be flawless for a whole day, but then would be unusable for a few hours). On the other hand, if the δ frequency error is intentionally made larger, it results in smaller, but more frequent transmission errors. This problem urged the development of Robust UPSOOK (RUPSOOK) that can operate in an error-free manner on a wide range of system parameters, and can be used to reliably transmit data (e.g., beacon IDs for the indoor positioning system).

The protocol uses only two symbols, which are single ON (H) and OFF (L) states of the transmitter, each symbol having width of T_B, as shown in Figure 5.1. First the data bits to be transmitted are Manchester-coded, i.e., each bit is represented by two symbols as follows:

$$\begin{aligned} 0&:\quad LH \\ 1&:\quad HL \end{aligned} \tag{5.1}$$

The Manchester coded bits can be considered as phase-shift keying signals, similarly to UPSOOK's MARK and SPACE symbols. A key difference is that, according to (3.2), each UPSOOK symbol contains $n > 1$ periods (for practical cases to make flickering unnoticeable for the human eye), while here only one period is utilized, as shown in Figure 5.1.

At the beginning of the packet the header is transmitted, which contains $N_{HEAD} = 5$ symbols of LHHHL. Due to the Manchester coding, the three consecutive H symbols can never be present in the data part, thus, the header can be separated from the data (see details of decoding later). The data segment, following the header, contains B bits, Manchester encoded.

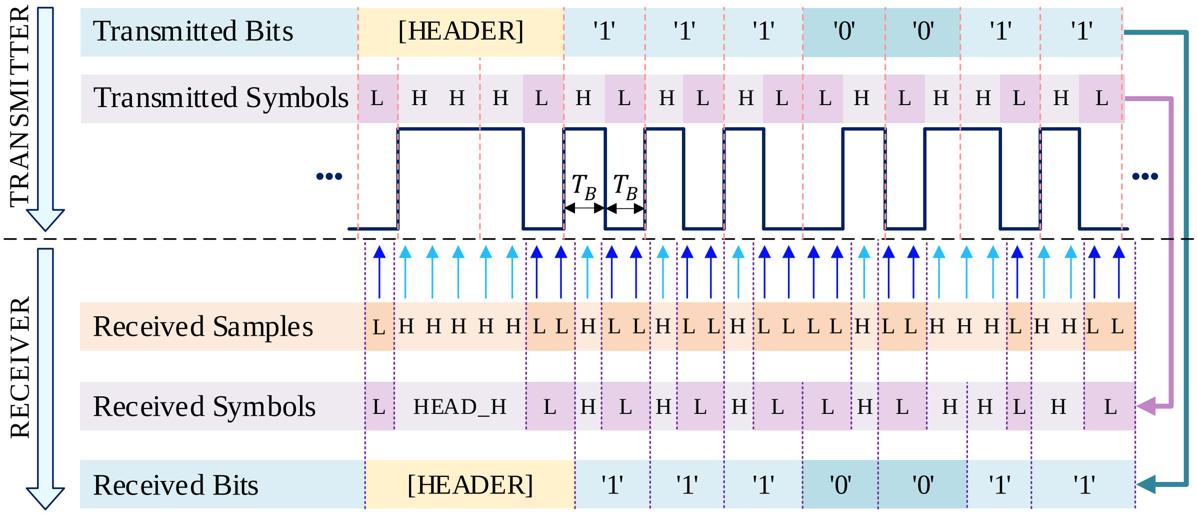


Figure 5.1 The operation of the Robust UPSOOK protocol.

The packet contains M symbols, where

$$M = N_{HEAD} + 2B. \tag{5.2}$$

The transmitter repeats the packet of M symbols continuously; thus, the header also serves as a footer. The receiver is configured to have an intentional frequency shift between the transmitter and receiver; the ideal sampling interval T_{CAM} of the receiver is set so that T_{CAM} is slightly longer than the packet time ($T_B M$):

$$T_{CAM} = T_B(M + c), \tag{5.3}$$

where $0 < c < 1$. With this setting, the continuously transmitted packet will be sampled so that the phase of the samples changes continuously, providing equivalent sampling intervals of

$$\delta_E = cT_B \tag{5.4}$$

between consecutive samples, as illustrated in Figure 5.2. If δ_E were zero, each instance of the packet would be sampled at the same phase. Since $\delta_E > 0$, instance $k + 1$ is sampled at a different phase, virtually δ_E time later than instance k. The equivalent sampling of the packet is shown in the lower part of Figure 5.2 and in the receiver part of Figure 5.1.

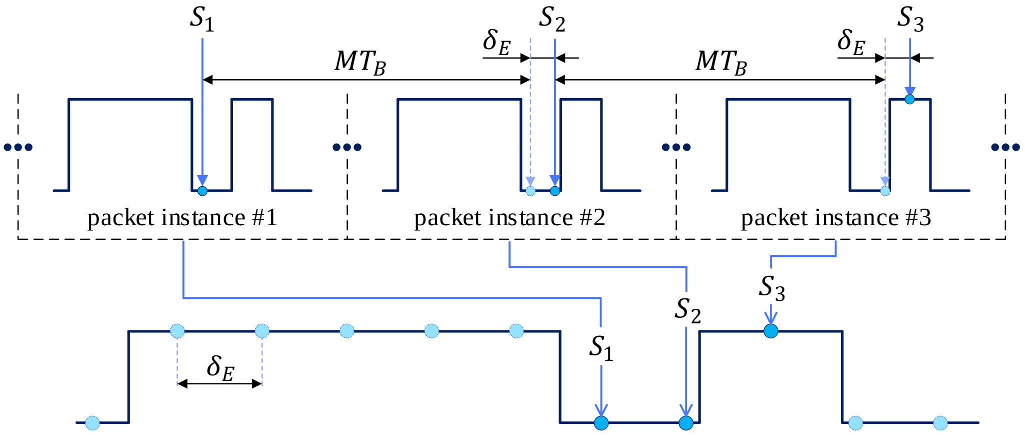


Figure 5.2 The sampling at the receiver side of Robust UPSOOK protocol, and its equivalent representation. The equivalent sampling interval is δ_E. The sampling instances are denoted with solid blue arrows.

5.1 Robust parameter design

The value of c should be chosen so that the following three requirements are fulfilled:

R1 from each symbol one or two samples are taken;
R2 from two consecutive symbols three or four samples are taken;
R3 from three consecutive symbols at least five samples are taken.

The above requirements allow the decoding of the symbols from the received samples, as shown in the receiver part of Figure 5.1. First the number of the same consecutive samples is determined. In Figure 5.1 the illustration shows one L, five H, two L, one H, etc. samples in row *Received Samples*. Using R1-R3, the symbols are decoded as follows:

- One or two of samples Z are converted to a symbol Z, where Z is H or L.
- Three or four of samples Z are converted to symbols ZZ, where Z is H or L.
- At least five samples of H are converted to HEAD_H.

The result of symbol decoding is illustrated in Figure 5.1, row *Received Symbols*. The received symbols are then converted to bits, using the following rules:

- An L followed by HEAD_H, followed by L is the HEADER.
- After the HEADER, B symbol pairs are decoded using (5.1).

The decoded bits are shown in Figure 5.1 in row *Received Bits*.

The value of constant c must be determined so that the protocol be tolerant towards the error sources. Most of the error sources, described in Section 3.2 for UPSOOK, are relevant for RUPSOOK as well: the effects of thresholding, noise, jitter, and saturation [S4] on the sampling are the same. The frequency error, which was characterized by the slip parameter δ_{CAM}, is now a design parameter, δ_E. Thus, the widths of the unsafe regions are now estimated by (3.14) with $\lambda_f = 0$.

In the protocol one, two, or three of the same symbols can follow each other (one or two in the data segment, three in the header). Since sampling around the edges is unsafe, again safe and unsafe regions are observed, as shown in Figure 5.3: the symbols' width is T_B, and the unsafe regions' width is λ.

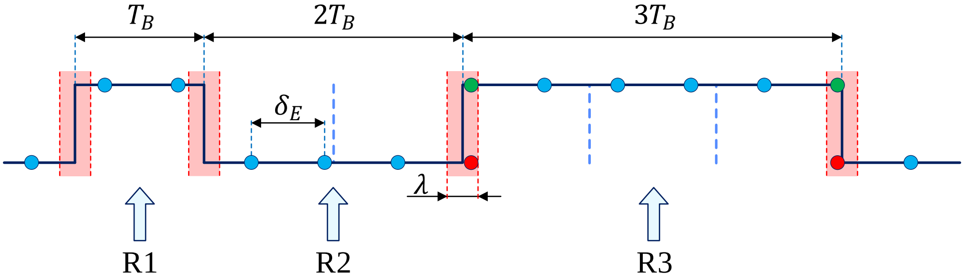


Figure 5.3 Unsafe regions (red rectangles) in a transmitted signal, showing the equivalent sampling of the signal (blue dots). The samples in an unsafe region may result in an incorrectly detected value (red dots), or in a correctly detected value (green dot). The example also illustrates requirements R1, R2, and R3.

Notice that the unsafe regions may virtually increase or decrease the width of a symbol in worst case by λ. E.g., in Figure 5.3 if the sample taken in the third unsafe interval (middle of the figure) is L, then the sensed length of two symbols L becomes longer and at the same time the length of the three symbols of H seems shorter. If the sample in the unsafe interval is H, then the effect will be the opposite.

Taking this effect into consideration, R1 can be expressed as follows (for illustration see the first symbol H in Figure 5.3):

$$T_B - \lambda > \delta_E \tag{5.5}$$

$$T_B + \lambda < 2\delta_E \tag{5.6}$$

Similarly, for R2 (see the two consecutive symbols L in Figure 5.3):

$$2T_B - \lambda > 3\delta_E \tag{5.7}$$

$$2T_B + \lambda < 4\delta_E \tag{5.8}$$

Finally, R3 results in the following constraint (see the three consecutive symbols H in Figure 5.3):

$$3T_B - \lambda > 5\delta_E \tag{5.9}$$

From (5.5), (5.7), and (5.9) it follows:

$$\delta_E < \min\left(T_B - \lambda, \frac{2}{3}T_B - \frac{1}{3}\lambda, \frac{3}{5}T_B - \frac{1}{5}\lambda\right) \tag{5.10}$$

Since in a meaningful case $\lambda < \frac{T_B}{2}$, (5.10) can be simplified to

$$\delta_E < \frac{3}{5}T_B - \frac{1}{5}\lambda. \tag{5.11}$$

From (5.6) and (5.8) the following constraint follows:

$$\delta_E > \max\left(\frac{T_B}{2} + \frac{\lambda}{2}, \frac{T_B}{2} + \frac{\lambda}{4}\right) = \frac{T_B}{2} + \frac{\lambda}{2}. \tag{5.12}$$

Let us use the following notation:

$$\lambda = a_\lambda T_B. \tag{5.13}$$

Using (5.4), (5.11), (5.12), and (5.13), the possible region for constant c is the following:

$$\frac{1}{2}(1 + a_\lambda) < c < \frac{3}{5}\left(1 - \frac{a_\lambda}{3}\right). \tag{5.14}$$

The solution exist if

$$a_\lambda < \frac{1}{7}. \tag{5.15}$$

From (5.13) and (5.15) it follows that the protocol can operate only if the unsafe interval λ is smaller than $T_B/7$. From (39) a region is given for possible values of c, where the width of the region is

$$c_{max} - c_{min} = \frac{1}{10} - \frac{7}{10} a_\lambda. \tag{5.16}$$

The value of c should be chosen as the middle of the interval:

$$c_{opt} = 0.55 + 0.15 a_\lambda \tag{5.17}$$

5.2 Adaptive thresholding

In practical cases the amplitudes A_L and A_H may change in time, which causes the change of ideal thresholding parameter Q_0. In the following a windowing-based adaptive estimator for Q_0 will be introduced. Let the sampled intensity value at time instant k be $I_S(k)$. Let us use a time window with length W. The minimum and maximum amplitudes are estimated as follows:

$$I_{MAX}(k) = \max\big(I_S(k - W + 1), I_S(k - W + 2), \dots, I_S(k)\big), \tag{5.18}$$

$$I_{MIN}(k) = \min\big(I_S(k - W + 1), I_S(k - W + 2), \dots, I_S(k)\big), \tag{5.19}$$

and the estimate of Q_0 is the following:

$$Q_0(k) = \frac{I_{MAX}(k) + I_{MIN}(k)}{2}. \tag{5.20}$$

The window length W must be chosen so that the window contains at least one sample from a symbol L and at least one sample from a symbol H, at any part of the signal, even in the header. From (5.4) it follows that $1/c$ expresses the average sample count per symbol. The header contains the largest possible consecutive symbols (LHHHL), thus to satisfy the requirement for the window length at least 4 symbols should be sampled, meaning $W > 4/c \approx 8$.

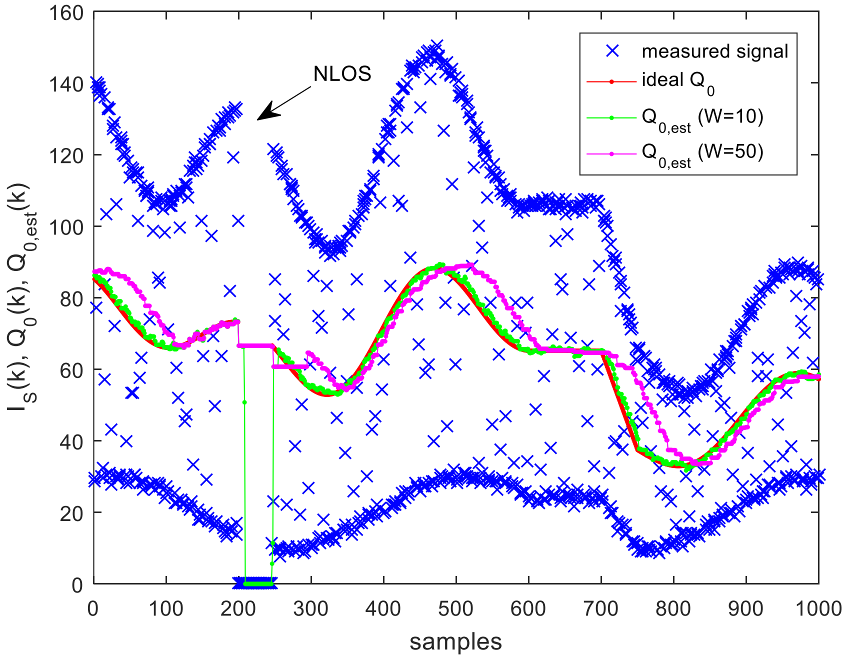


Figure 5.4 Adaptive estimation of threshold Q_0.

The operation of the adaptive estimator is illustrated in Figure 5.4. The varying input signal intensity, denoted by blue crosses, is synthetized with $B = 8$ bits and camera aperture time of $S = 500$ µsec. The ideal Q_0 value is calculated using (3.7), and plotted in red. The signal contains additive noise (with standard deviation of 1 LSB). In the signal there is a section where the line of sight is blocked (between samples 200 and 240), and a section where the amplitude is constant (between samples 600 and 700). In the experiment two windows were used with $W = 10$ and $W = 50$.

Both estimators estimate well the constant region. The smaller window (shown in green) allows faster adaptation when the signal changes, thus, the estimated threshold follows closely the ideal threshold. The larger window (shown in magenta) produces slower adaptation, the delay between the ideal and estimated values is apparent.

When a non-line-of-sight (NLOS) situation occurs, the estimated threshold decreases to $A_H/2$ within a few samples (visible for both windows), and falls to zero not later than W samples after the start of the NLOS (visible for $W = 10$). In both cases the estimate quickly recovers when the line of sight is restored.

According to the experiment, the smaller window size provides faster and more accurate estimate, thus, in practice a good choice is a window with size slightly above the minimum vale, e.g., around 10.

Notice that the adaptive mechanism only sets the threshold value and thus, has no effect on the latency of the protocol.

5.3 Evaluation

In this section the performance of the proposed RUPSOOK protocol is evaluated using simulations and real measurements.

For evaluation purposes, instead of c, a more practical parameter, called Samples Per Bit (SPB), will be used. Since the length of one bit is $2T_B$, the number of samples per bit is, using (5.4), can be expressed as follows:

$$SPB = \frac{2T_B}{\delta_E} = \frac{2}{c}. \tag{5.21}$$

Notice that according to (5.17), the ideal value of c is around $0.55 \dots 0.56$, which corresponds with ideal samples per bit value of $SPB \cong 3.6$. Table 5.1 shows the main technical parameters, used in the simulations and measurements.

Parameter	**Simulations**	**Laboratory I.**	**Laboratory II.**	**Long-distance**
T_{CAM} (sec)	1/30	1/30	1/30	1/30
S (µsec)	100	100	15	100
$f\text{-}number$	$n.a.$	$f/8$	$f/2.8$	$f/2$
$focal\ length$ (mm)	$n.a.$	2.7	2.7	6
B(bits)	5 + header	9 + header	9 + header	20 + header
SPB (c)	2.5…5.0 (0.4…0.8)	2.75…4.25 (0.47...0.73)	2.75…4.75 (0.47...0.73)	3.65 (0.55)
T_B (msec)	$\cong 2.381$	$\cong 1.515$	$\cong 1.515$	$\cong 0.7575$
Q (LSB)	$0 \dots 255$	$0 \dots 255$	$0 \dots 255$	$0..40$
$A_L;\ A_H$ (LSB)	$40;\ 160$	$42;\ 155$	$43;\ 154$	0; 28-36 (varied)
std_{noise} (LSB)	5 (controlled)	0.8 (L), 2.31 (H)	0.9 (L), 2.7 (H)	0 (L), 1.7 (H)
P_{LED} (W)	$n.a.$	0.0015	0.0015	4.9
$total\ \#bits\ sent$	1000/experiment	1800/experiment	1800/experiment	38280

Table 5.1 Technical parameters of the measurements.

5.3.1 Simulations

The behavior of the proposed protocol was analyzed in a well-controlled simulation environment. The sampled signal was simulated using an ideal square-wave blinking signal and a camera model according to (2.5). The resulting signal was similar to the trapezoidal signal shown in Figure 3.2 (a), with $T_B = T_{DATA}/2$, with tunable parameters T_B and S. By varying the sampling rate, various SPB values were set, and the variation of the sampling frequency allowed the simulation of jitter and frequency error as well. With additive nose the effect of noise was modelled.

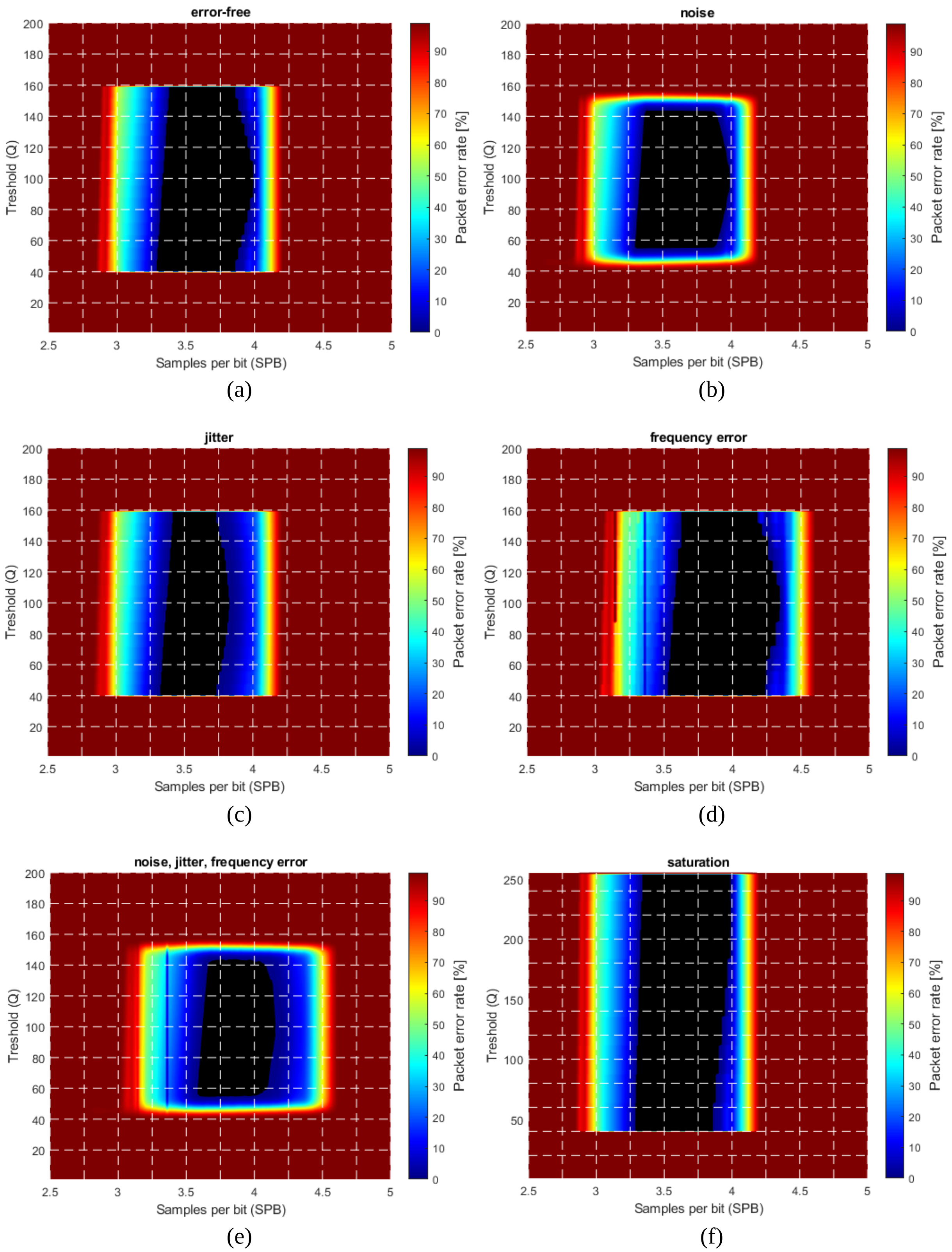
The camera frequency was set to 30 Hz, the simulated aperture time was 100 µs, the received signal amplitude was set between 40 and 160. The threshold Q was varied from 0 to 255, and the transmitter frequency was tuned to provide samples per bit (SPB) between 2.5 and 5. In each simulation the same transmission sequence with 200 packets were used and the Packet Error Rate (PER) was calculated as follows:

$$PER = \frac{number\ of\ bad\ packets}{number\ of\ all\ packets}. \tag{5.22}$$

In Figure 5.5 the PER is shown as a function of the threshold Q and samples per bit SPB. Figure 5.5 (a) shows the ideal case, where no error source is present. The central part, shown by black, is error free ($PER = 0$). The error-free region is located around $SPB = 3.6$, corresponding well with the theoretical results. As it was expected, the protocol operates well with thresholds between the minimum and maximum

Figure 5.5 (b) shows the effect of additive noise with variance $\sigma = 5$ LSB. The error-free region decreased by approximately 3σ LSB around the threshold values corresponding to the minimum and maximum signal amplitude (i.e., the top and bottom of the error-free region).

For better visibility, in the simulations a high jitter value in the range of ± 100 µs, was used, with uniform distribution. The effect is shown in Figure 5.5 (c): the shrinking of the error-free region on both the left-hand side and right-hand side is clearly observable.



(a) (b)

(c) (d)

(e) (f)

Figure 5.5 Simulated PER in the presence of various error sources. (a) ideal case, (b) noise with $\sigma = 5$ LSB, (c) ± 100 µsec jitter, (d) camera frequency error of with $\Delta f_{CAM} = -0.1$ Hz, (e) combined noise, jitter, and frequency error, (f) saturation.

The frequency error was modelled as a constant bias in the camera frequency: the camera frequency was set to 29.9 Hz. The effect is visible in Figure 5.5 (d), where the original error-free region is shifted to the right. The explanation is the following: RUPSOOK uses a constant frequency difference between the camera and the transmitter, according to (5.3) and (5.4). This frequency difference is modified by the constant frequency error of the camera, thus, the designed SPB value is different from the nominal value, resulting in a shift of the error-free region. If the sampling of camera is faster/slower than expected then the real SPB is higher/lower than the nominal, causing shift of the error-free region to the left/right. The theoretical relationship between the nominal SPB, the real SPB', and the frequency error Δf is the following:

$$SPB' \cong \frac{SPB}{1 + SPB\frac{1}{2T_B}\frac{\Delta f_{cam}}{f_{cam}^2}} \cong SPB - SPB^2\frac{1}{2T_B}\frac{\Delta f_{cam}}{f_{cam}^2}. \tag{5.23}$$

As an example, if the nominal SPB is 3.5 and the frequency error is 0.1 Hz then the real SPB' is 3.2, thus, the values corresponding to $SPB = 3.5$ in Figure 5.5 (d) correspond to values $SPB = 3.2$ in Figure 5.5 (a), producing the shift visible. Notice that according to (5.23) the value of the shift depends on the SPB: the larger the SPB the larger the shift.

The effect of multiple error sources, including noise, jitter, and frequency error is shown in Figure 5.5 (e). The levels of the disturbances are equal to those of the individual sources presented in Figure 5.5 (a)-(d). The combined effect, containing shrinking and shifting of the error-free region, is obvious.

The effect of saturation is illustrated in Figure 5.5 (f). In the example the signal values changed between 40 and 400, saturated at 255. Without saturation (up to amplitude 400) the figure would be an enlarged version of the saturation-free case of Figure 5.5 (a), but due to saturation, it is truncated at the saturation level of 255. Thus, Figure 5.5 (f) is the stretched version of the lower half of Figure 5.5 (a), caused by sensor saturation.

5.3.2 Measurement setup

During the tests two measurement setups were utilized: a well-controlled laboratory environment and a long-distance outdoor setup. In the laboratory setup the camera and the low-power transmitters were built into an enclosure blocking external light, where the minimum and maximum light intensity, along with the noise level can be controlled. The photo of the equipment is shown in Figure 5.6. In the long-distance setup power LEDs were used, as transmitters. The map of the premises and the photo of the equipment are shown in Figure 5.7.

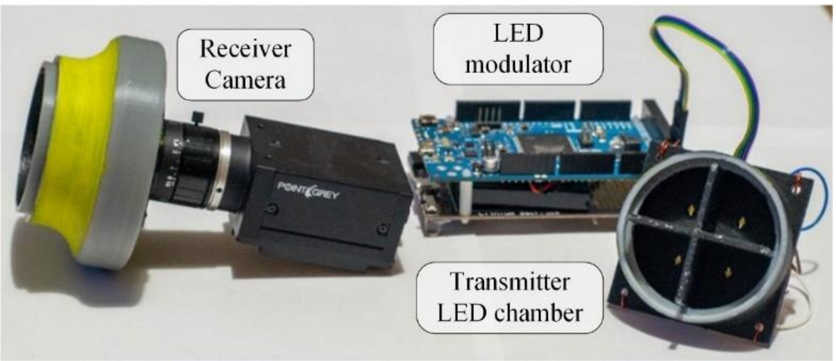


Figure 5.6 The setup for laboratory measurements.

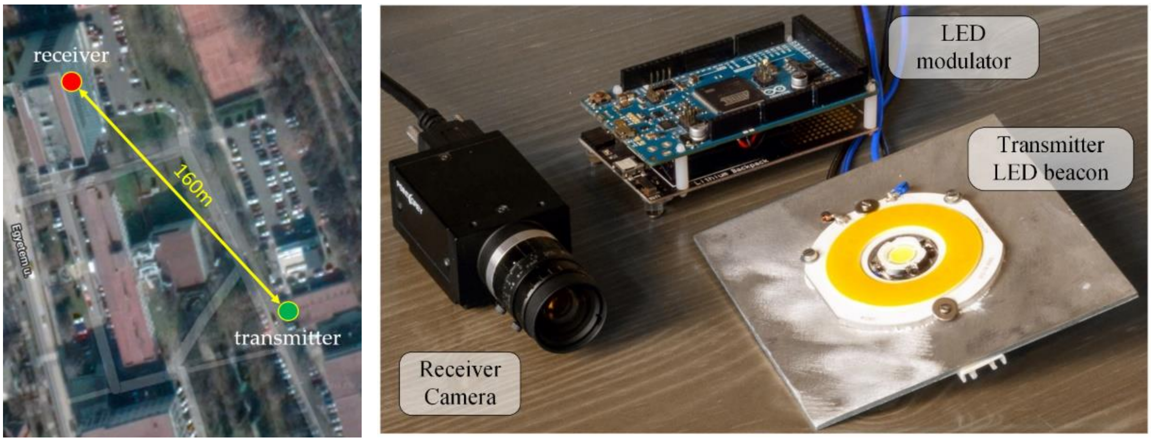


Figure 5.7 Long distance measurement setup and the photo of the equipment.

The main technical parameters of the measurements are shown in Table 5.1. Notice that in Table 5.1 the value of T_{CAM} is the nominal value, but the camera provided slightly different sampling period; thus, parameter T_B was tuned while measuring the actual value of parameter SPB. The measurement method for SPB is described in the next Section.

5.3.3 Measurement of the SPB value

For the tests the exact SPB value need to be determined. Parameter T_B of the transmitter can be measured easily with high precision, using a time/frequency meter, but for most cameras the measurement of the exact sampling frequency is troublesome. Thus, the value of δ is difficult to obtain. Instead, a measurement setup was used to directly measure the SPB, as shown in Figure 5.8.

The transmitter is the RUPSOOK transmitter under test, but during the SPB measurement the packets do not contain header and the same symbols (e.g., bit 1's HL) are transmitted, resulting a continuous alternating sequence of HLHLHL… The received samples are binarized by a comparator. The measurement is performed using two counters: one counter contains the number of received bits N_b, while the other counts the number of received frames (or samples) N_f. After each received full bit the ratio of the samples and bits is calculated, providing an estimate for the SPB value:

$$SPB = \frac{N_f}{N_b} \pm \frac{1}{N_b}. \tag{5.24}$$

Figure 5.8 shows the result after the reception of 4 full bits, containing 27 samples. Note that the counters should be cleared when the transmitter frequency is changed. Using the real-time feedback of SPB values, the required SPB can be easily set by tuning the transmitter's clock frequency.

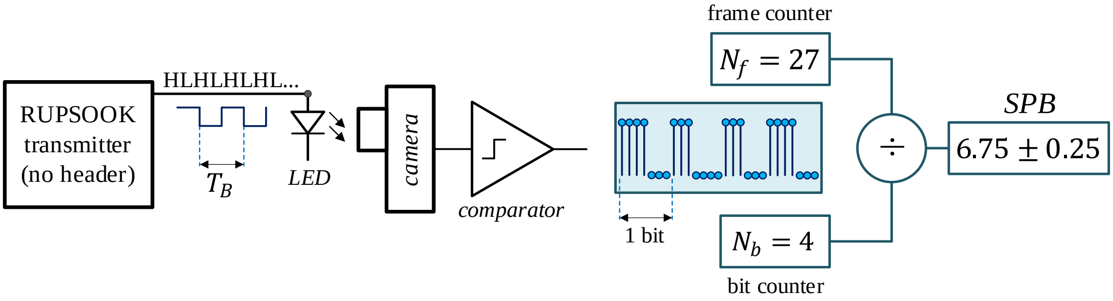


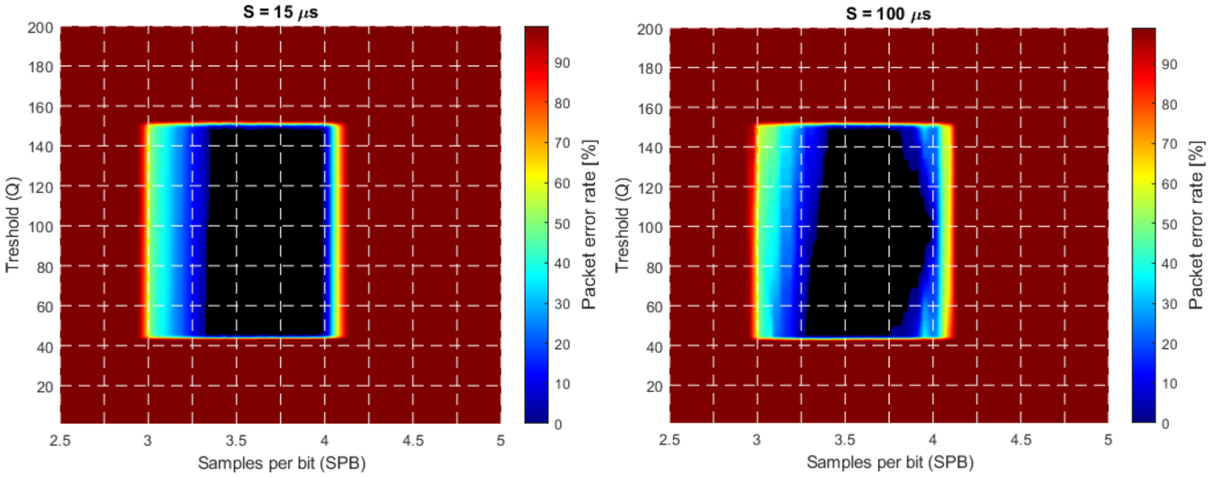
Figure 5.8 Measurement setup to determine the SPB value.

Notice that the speed of measurement depends on the actual SPB value and the required precision. E.g., for $SPB = 3$ each bit contains 3 samples, thus, samples corresponding to one bit are collected in $0.1sec$, using $f_S = 30\,\text{Hz}$. In order to provide measurement accuracy of ± 0.05, according to (5.22), $N_b = 20$ is necessary, thus, the measurement time is $20 \cdot 0.1\text{ sec} = 2\text{ sec}$.

5.3.4 Laboratory measurements

The laboratory measurements contained two setups, using aperture times of 100 μsec and 15 μsec. The technical parameters are listed for both setups in Table 5.1. In each test different SPB values were set, ranging from 2.75 to 4.25. The Packet Error Rate (PER), as a function of parameters SPB and Q, was calculated.

Figure 5.9 shows the PER as a function of SPB and Q, for both setups. The shape and size of the error-free regions resemble to the those of simulation results. In case of smaller aperture time the tilting on the left- and right-hand side of the error-free region is narrower. The robust operation is apparent: the PER is zero for a wide range of parameters SPB and Q, as shown by the black area in the center.



(a) (b)

Figure 5.9 The packet error rate as a function of parameters SPB and Q.
(a) $S = 15$ μsec, (b) $S = 100$ μsec.

Notice that the robust operation region provides guaranteed response time. E.g., in case of $SPB = 3.5$ the decoding of 11-bit long packet requires 39 samples. Since in worst case the reception begins slightly after the start of the first header symbol, almost two packets must be received for the first detection. In this case, considering the utilized camera frequency, the worst-case guaranteed response time will be 2.6 sec.

5.3.5 Long-distance measurement

A long-distance measurement was also conducted to check the robustness of RUPSOOK in extreme conditions. The camera was placed in a 9th-floor window, while the transmitter was deployed at the side of a nearby road (see Figure 5.7). The distance between the transmitter and the camera was 160m. In this experiment no additional noise was added to the measurement. The noise was observable only at state H (with standard deviation of 1.7 LSB), state L was always zero. A sample packet can be seen in Figure 5.10. Notice that most of the samples are around zero and 30 but there are numerous samples in between: these were taken in unsafe intervals.

The PER is shown in Figure 5.11, as a function of parameter Q. The performance of the protocol was perfect (zero error) with parameters $1 \leq Q \leq 20$, for the 38280 transmitted bits: 1740 packets were transmitted and successfully received. The header was 2 bits long in this experiment (instead of the 2,5 bits shown in Figure 5.1) with HHHL symbols (the last symbol of the packet in the data region was L, so the first L symbol of the header was omitted).

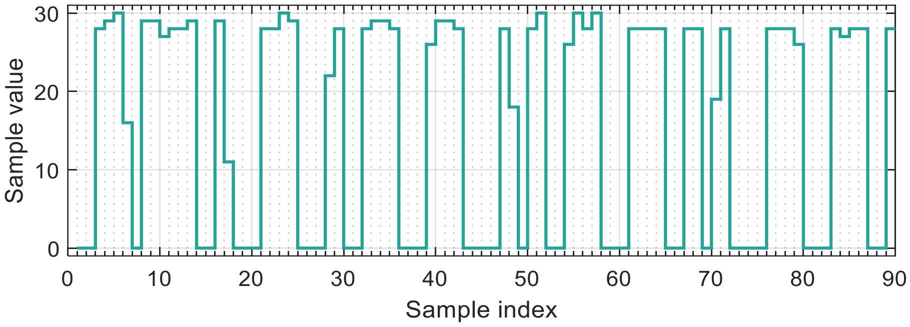


Figure 5.10 Example samples from the long-distance measurements.

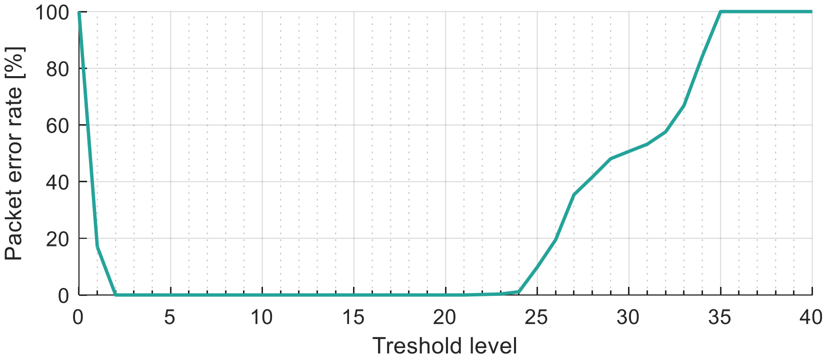


Figure 5.11 The packet error rate of the long-distance measurement, as a function of parameter Q.

5.4 Comparison and conclusion

The main performance properties of UFSOOK [61], UPSOOK [62], the Undersampled 64-PAM [78], and the proposed RUPSOOK protocol are listed in Table 5.2.

	UFSOOK	UPSOOK	UPAM-64	RUPSOOK
Bits / Sample	0.5	1	8	0.28
Covered distance	medium	medium	low	high
Theoretical *PER*	> 0	> 0	> 0	$= 0$
QoS	no	no	no	yes
Worst-case response time	$O(1/\delta)$	$O(1/\delta)$	$O(1/\delta)$	$O(1)$

Table 5.2 Performance properties of UFSOOK [61], UPSOOK [62], Undersampled 64-PAM [78], and RUPSOOK

The highest spectral efficiency belongs to U-64-PAM, where the PAM coding allows the coding of multiple (8) bits into one symbol. In this respect RUPSOOK has the lowest performance with only 0.28bits/sample. Protocols with higher bandwidth, using various color coding and parallel channels, were also proposed (e.g., [77], [79]), but for the target application field these are not applicable, due to their limited communication range. The covered distance is low in case of U-64-PAM, due to the sensitive coding. Both UFSOOK and UPSOOK performs better, due to the OOK coding, but the observed sensitivity of the header to distance reduces the possible distance coverage [62]. RUPSOOK's better performance is due to its header structure, where there is no need for a different symbol. The theoretical PER is zero for RUPSOOK, with correct settings. The wide range of correct settings makes the protocol robust. The only correct setting for the other protocols is the tight synchronization, which is not practical. Thus, RUPSOOK is able to guarantee Quality of Service (QoS) for the response time, while the other protocols are not. The worst-case response time is constant for RUPSOOK, while this time is inversely proportional with the frequency error δ for other protocols.

When combined with the size change-based data encoding-decoding method and beacon design introduced in Section 4, RUPSOOK can also satisfy the trackability requirement. This combined solution was superior to any other solution found in the literature at the time that could be used to transfer beacon identification numbers in the visible light-based indoor positioning system introduced in Section 1.4.

6 Novel HIAL ADoA positioning method

Sections 2-5 dealt with the topic of VLC protocols that can be used in the IPS system presented in Section 1.4 to transmit an LED beacon's identification number, so that the system may recognize the beacons, and use them as anchor (reference) points for positioning purposes.

In this section, the Heuristic Inscribed Angle-based Localization (HIAL) method will be presented, which can be used to estimate the position and orientation of a camera in the indoor positioning system. The camera is using global shutter and is equipped with fisheye lens. A key requirement for the method is that the camera has to face upwards, so that the normal vector of the camera is parallel with the z axis of the real-world coordinate system, or its tilt needs to be compensated. The method uses angle difference measurements (from the camera's point of view) of tracked and identified beacons in the IPS, which position is known in the real-world coordinate system.

If the viewing angle between two beacon points (also known as Angle Difference of Arrival) P_i and P_j is $\vartheta_{i,j}$, measured at point C (i.e., $\sphericalangle P_i C P_j = \vartheta_{ij}$), then the possible position of point C is on a circularly symmetric surface, which is constructed as the rotation of a circular arc around chord P_i and P_j, corresponding to viewing angle $\vartheta_{i,j}$ [17]. Such surfaces are illustrated in Figure 6.1. If multiple pairs of beacons are available, multiple surfaces can be constructed, the intersection of which determines the location of C (camera), as was used e.g., in [17]. In HIAL, however, it is not the viewing angle $\vartheta_{i,j}$ in 3D, what is measured, but instead the 2D viewing angle $\alpha_{i,j} = \sphericalangle P'_i C' P'_j$, where P'_i, P'_j, and C' are the orthogonal projections of P_i, P_j, and C, respectively, to plane x-y, as shown in Figure 6.1 (a). This can be done due to the abovementioned requirement about the camera's normal vector. Using multiple pairs of beacons, first the C' position of the camera on the x-y plane is calculated, as the intersections of circular arcs, followed by the calculation of the height and azimuth of the camera.

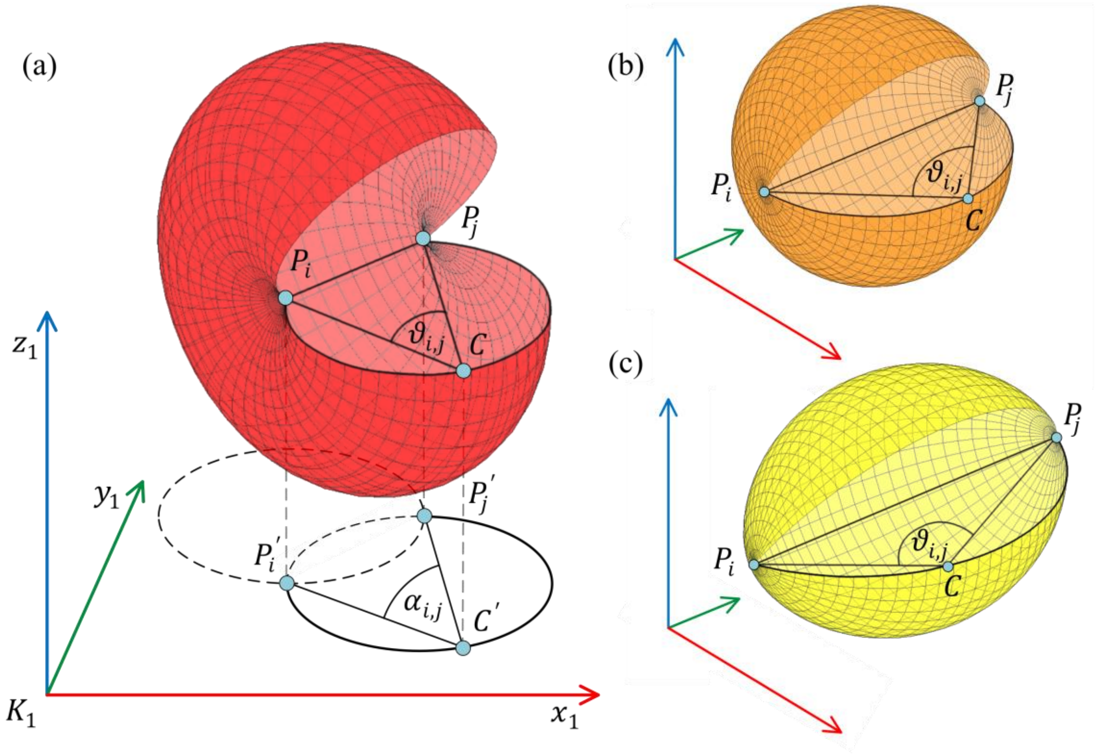


Figure 6.1 The surfaces show the possible location of the camera if the $\vartheta_{i,j}$ angle difference of arrival from two beacons P_i and P_j are measured: (a) Apple surface of a spindle torus, when $\vartheta_{i,j} < 90°$, also showing the measured 2D viewing angle $\alpha_{i,j}$; (b) Sphere, when $\vartheta_{i,j} = 90°$; (c) Lemon surface of a spindle torus, when $\vartheta_{i,j} > 90°$.

The utilized coordinate systems and their relations are shown in Figure 6.2. The real-world coordinate system is K_1 (shown by solid axes), in which the unknown camera coordinates are represented by vector $\boldsymbol{c} = (x_c, y_c, z_c)$ and the known position of the j-th reference point (LED beacon) is $P_j = (x_j, y_j, z_j)$. Camera coordinate system K_2 (shown by dashed axes) is a translated and rotated version of K_1, where the rotation is performed around axis z by unknown angle φ, and the translation is characterized by the unknown vector $\boldsymbol{c}$, as shown in Figure 6.2. Notice that z_1 and z_2 are parallel, since the camera is facing upwards (or is being compensated to do so). The position of beacon j in K_2 is $\boldsymbol{p}_{\boldsymbol{j}}^{(\boldsymbol{2})} = (x_{p,j}{}^{(2)}, y_{p,j}{}^{(2)}, z_{p,j}{}^{(2)})$. The fisheye camera shows the beacon in a heavily distorted image in coordinate system K_3, from which, using the camera calibration data, unit vector $\boldsymbol{v_j} = (x_{v,j}{}^{(2)}, y_{v,j}{}^{(2)}, z_{v,j}{}^{(2)})$ can be calculated, which has approximately the same direction (will be discussed in Section 6.3) as $\boldsymbol{p}_{\boldsymbol{j}}^{(\boldsymbol{2})}$, as shown in Figure 6.2 (see e.g., [80] for detailed discussion). The orthographic projection of $\boldsymbol{v_j}$ to the x-y plane is $\boldsymbol{v'_j} = (x_{v,j}{}^{(2)}, y_{v,j}{}^{(2)})$, from which $\alpha_j = \mathrm{atan2}\left(y_j{}^{(2)}, x_j{}^{(2)}\right)$ can be calculated, where α_j is the angle of $\boldsymbol{v'_j}$ in K_2, as shown in Figure 6.2.

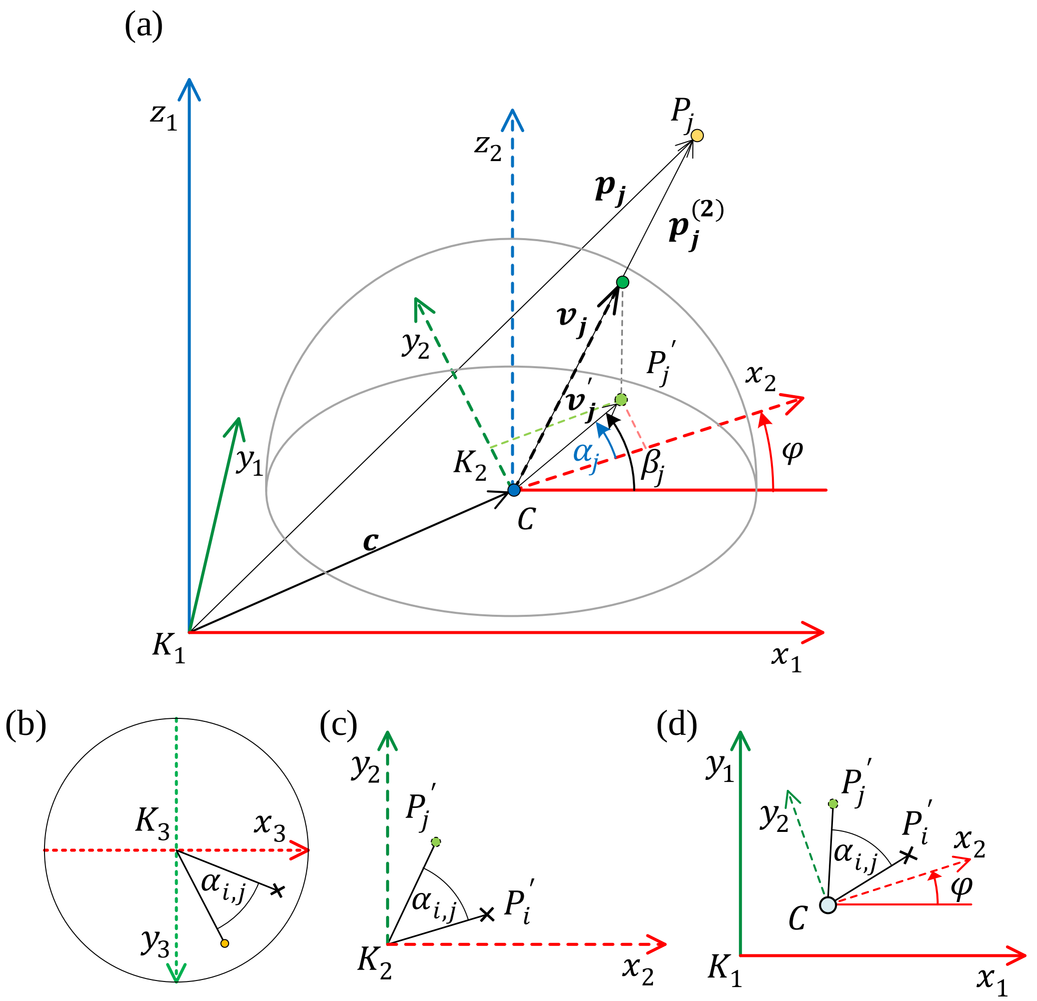


Figure 6.2 Coordinate systems used in positioning. K_1: world, K_2: camera calibrated, K_3: camera raw (fisheye). The unknown camera location and orientation in K_1 are C and φ, respectively. The coordinate systems (only x-y plane) are illustrated on the right-hand side.

When a beacon pair i and j is detected, the angle difference of arrival $\alpha_{i,j} = \alpha_j - \alpha_i$ is calculated from angles α_i. and α_j. Notice that α_i and α_j are measured in K_2, but the viewing angle α_{ij} is the same in K_1 and K_2, as illustrated in Figure 6.1 (a). Using the inscribed angle theorem, for fixed points P_i' and P_j', the possible location of C', for which $\sphericalangle P_i' C' P_j' = \alpha_{i,j}$ is an arc of a circle, as shown in Figure 6.3 (a).

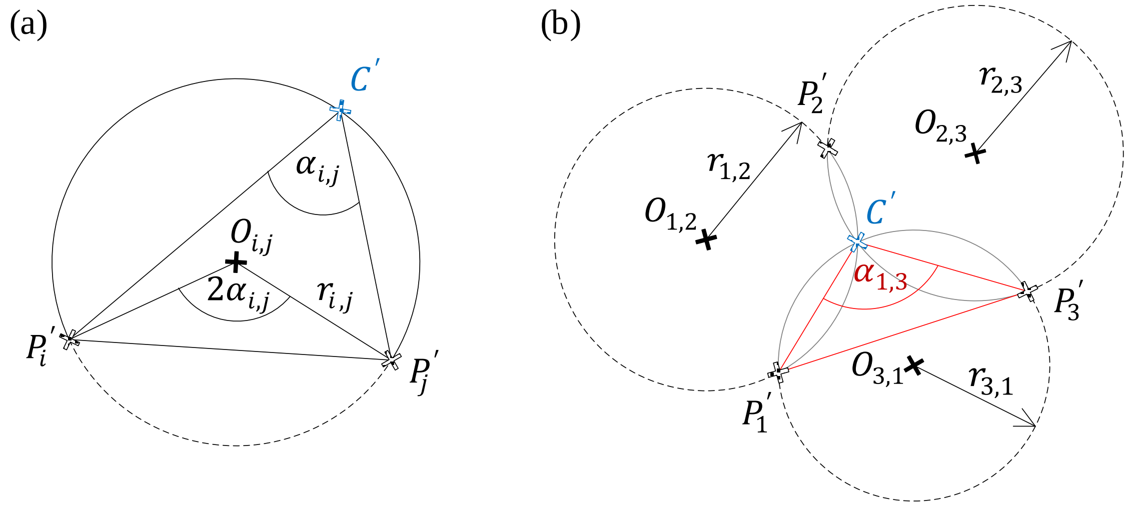


Figure 6.3 (a) The possible location of C' is on an arc of circle, when viewing angle $\sphericalangle P_i' C' P_j'$ is α_{ij}. (b) The calculation of position C', as the intersection of three arcs of circle.

Let us denote the arc, defined by viewing angle α_{ij} and two points P_i' and P_j', by the triplet $[\alpha_{ij}, P_i', P_j']$. The center $O_{i,j}$ and radius $r_{i,j}$ of each arc of circle can be calculated using e.g., [58].

Having three anchor points P_1, P_2, and P_3, three pairs of anchors and thus three corresponding arcs $[\alpha_{1,2}, P_1', P_2']$, $[\alpha_{2,3}, P_2', P_3']$, $[\alpha_{3,1}, P_3', P_1']$ can be calculated, the intersection of which is the location of C', as shown in Figure 6.3 (b).

The calculation of C' is straightforward when three beacons are available, according to Figure 6.3 (b). When multiple beacons are visible, however, more than three pairs can be selected and thus more arcs can be calculated. This allows the creation of a more robust, fault-tolerant estimate.

Theoretically, all arcs should intersect in one point, but in reality, due to measurement errors, several intersections may be present. Errors can occur due to small measurement inaccuracies, e.g., inaccurate beacon location in the database or small detection errors. These sources can be modelled as small measurement noise and they result in the scattering of intersections around the true location. Due to the nature of the detection, however, outliers are not uncommon, e.g., due to reflections a mirror image of a beacon is detected, which results a completely bad arc and several intersections far from the true location. Thus, simple statistical approaches (e.g., least squares method) do not work when bad detections can be present. To avoid large bias caused by outliers, a fast heuristic scoring of the intersections is proposed, which helps the selection of a group of possibly valid intersections, which do not contain outliers.

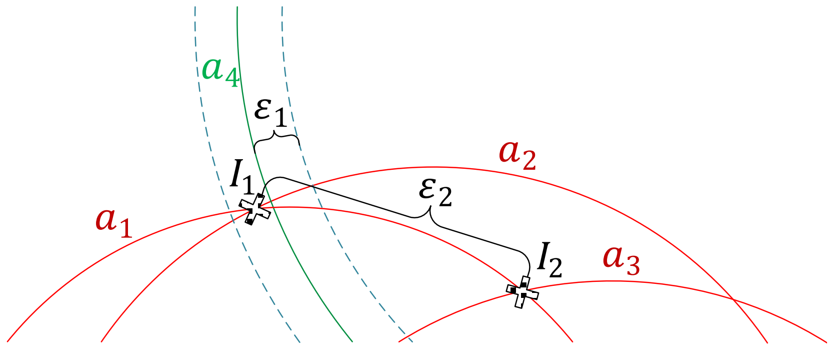


Figure 6.4 Scoring of arc intersections with parameters ε_1 and ε_2

The scoring process is illustrated in Figure 6.4. Every pair of detected beacons is used to create an arc of circle, and every pair of arcs creates an intersection point. The arcs in Figure 6.4 are denoted by a_1, a_2, a_3, a_4, while the intersections are I_1 and I_2. Intersection I_1 was computed from a_1 and a_2, while I_2 is the intersection of a_1 and a_3.

An intersection gets a score if

- it is close to an arc, which was not used in the computation of the intersection (a point is considered close to an arc if their distance is smaller than ε_1, where ε_1 is a predefined constant; in Figure 6.4, I_1 is close to a_4, that was not used to create I_1, thus I_1 gets plus one score); or
- it is close to another intersection (two intersections are close to each other if their distance is smaller than ε_2, where ε_2 is a predefined constant; in Figure 6.4, I_1 is close to I_2 thus both I_1 and 2 get plus one score).

In the illustration of Figure 6.4 intersections I_1 and I_2 have 2 and 1 scores, respectively. Notice that the higher the score of a point the higher number of measurements it is consistent with. Thus, the location estimate $\hat{C}$ is the intersection I_{max} with the highest score. If multiple intersections share the same highest score, their mean position is used for the location estimate.

Once $\hat{C}$ is known in K_1, the calculation of the camera orientation (azimuth) and height is straightforward, as follows. Let P_j be a beacon which was used to create one of the arcs that resulted I_{max} (i.e., P_j is possibly a beacon that was correctly detected). First, vector $C'P_j'$ is calculated. Let us denote its angle by β_j, as shown in Figure 6.2. The unknown camera orientation is calculated as $\hat{\varphi} = \beta_j - \alpha_j$.

Notice that $\boldsymbol{p}_j^{(2)} = u\boldsymbol{v}_j$, where scalar u is unknown. Since $C'P_j' = u\boldsymbol{v}_j'$, u can be calculated as $u = \|C'P_j'\| / \|\boldsymbol{v}_j'\|$. Thus, the z coordinate of beacon P_j in K_2 is $z_{p,j}^{(2)} = uz_{v,j}^{(2)}$. Finally, the height of the camera is calculated as $z_c = z_j - z_{p,j}^{(2)}$.

6.1 Error analysis

The possible error sources were analyzed with Monte Carlo simulations. For easier comparison, the test configuration in the simulations reflects the real measurement setup (see Section 6.2)

The tests were performed in a room with size of approx. 6 m x 6 m, where 17 test positions were defined in the central 3 m x 3 m area, as shown in Figure 6.5. In the room 6 beacons were installed, thus different configurations with different number of beacons could be tested. The beacon positions are denoted by $B_1, B_2, \dots, B_6$ in Figure 6.5. The 17 test positions are shown by blue crosses. The z coordinates of the beacons and the test points were 3.04 m and 0.02 m, respectively (not shown in Figure 6.5).

Notice that in the simulations beacon detection and tracking capabilities were not tested: the detected beacon positions were calculated from the ideal physical model and additional errors were added, as will be described in each case.

In the tests the following error sources were analyzed:

- Camera orientation error: the camera is supposed to look upwards. This simulated error source reflects real situations where the camera orientation is not accurate.
- Detection error: the beacon's position is detected in the image with possible error. This error source describes e.g., errors due to inaccurate camera compensation, quantization, or inaccuracies of the preprocessing algorithms.
- Reference position error: beacons are assumed to be in known positions. This error source simulates measurement inaccuracies of the reference positions.
- Outliers: real detections may contain large errors (e.g., due to reflections or incorrect beacon identification). This simulated error source injects large measurement errors to analyze the fault tolerance capabilities of the positioning algorithm.

In the tests the proposed algorithm using n beacons will be denoted by HIAL-n (HIAL-3, HIAL-4, HIAL-5, and HIAL-6). The applied beacons setups were the following:

- HIAL-3: beacons 1, 3, and 4
- HIAL-4: beacons 1, 2, 3, and 4
- HIAL-5: beacons 1, 2, 3, 4, and 5
- HIAL-6: all 6 beacons

For reference, the widely used Maximum Likelihood Perspective-n-Point (MLPnP) algorithm was used (always with the minimum required 6 beacons, and without covariance information feedback) [76].

The 3D error is defined as the distance between the true location and the location estimate. In several applications the 2D error is more significant: here only the x and y coordinates are used for error calculation.

6.1.1 Effect of camera orientation error

The camera is supposed to look upwards, but the camera orientation in reality may not be accurate. In the simulation the tilt of the camera was changed by 1-5° (in steps of 1°) from the ideal (upwards) direction. The azimuthal orientation of the camera was swept around from 0 to 360°. For each non-ideal camera position the beacon detections were calculated, ignoring other possible error sources, and the localization algorithm was run for each case. The results are plotted in Figure 6.5 with red color. According to the results, for a constant amount of tilt error (e.g., 1°) the localization error is almost constant: the error trajectory is approximately circular (the smallest circles correspond to 1°, the largest ones to 5°).

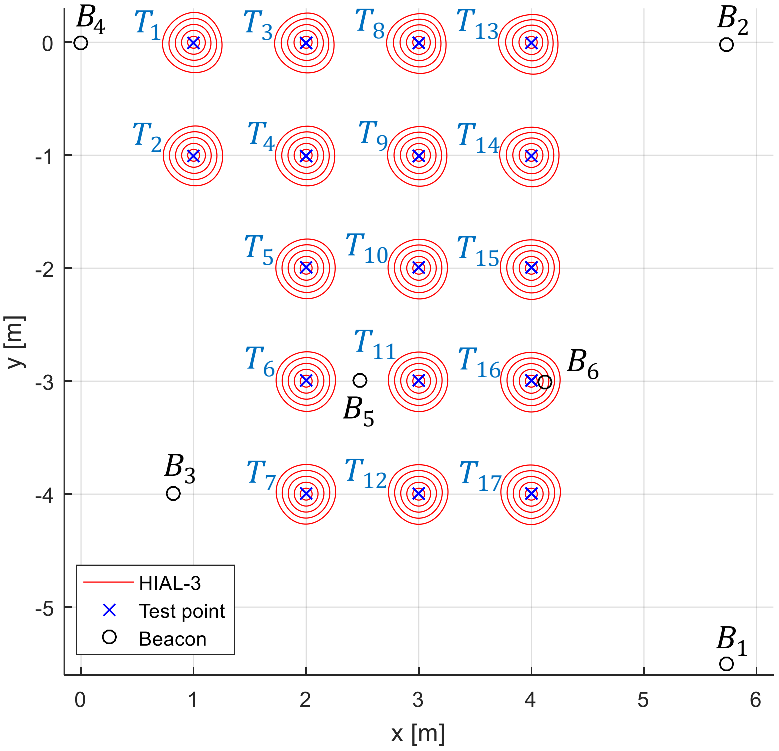


Figure 6.5 Test environment and the result of camera angle perturbation analysis in 2D.

The detailed results are listed in Table A.1, both for 2D and 3D cases. It can be observed that the localization error is approximately proportional with the orientation error. In the test environment 1° of orientation error resulted in approximately 5 cm of position error. The number of beacons had no significant effect on the localization error. Notice that this test is not meaningful for reference algorithm MLPnP, since it is able to handle any camera orientation.

6.1.2 Effect of detection error

Due to the non-ideal physical properties of the camera and the inaccuracies of the applied processing algorithms the detection of the beacons may be inaccurate. In the tests ideal detections were simulated for all 17 test points. In the next step the simulated detection images were corrupted by various amount of error (i.e., the center of the detected object was moved by some pixels, in random directions). The localization algorithms were run on each corrupted image.

The tests were performed using $n = 1, 2, 3$ pixels of error, and for each error value 800 independent simulations were run. Figure 6.6 illustrates the 2D error for HIAL-6 and MLPnP, for $n = 2$. The error greatly depends on the location, e.g., in T_{11} for HIAL-6 and T_{13} for MLPnP large errors (10-20 cm) can be observed, possibly due to geometric dilution of precision.

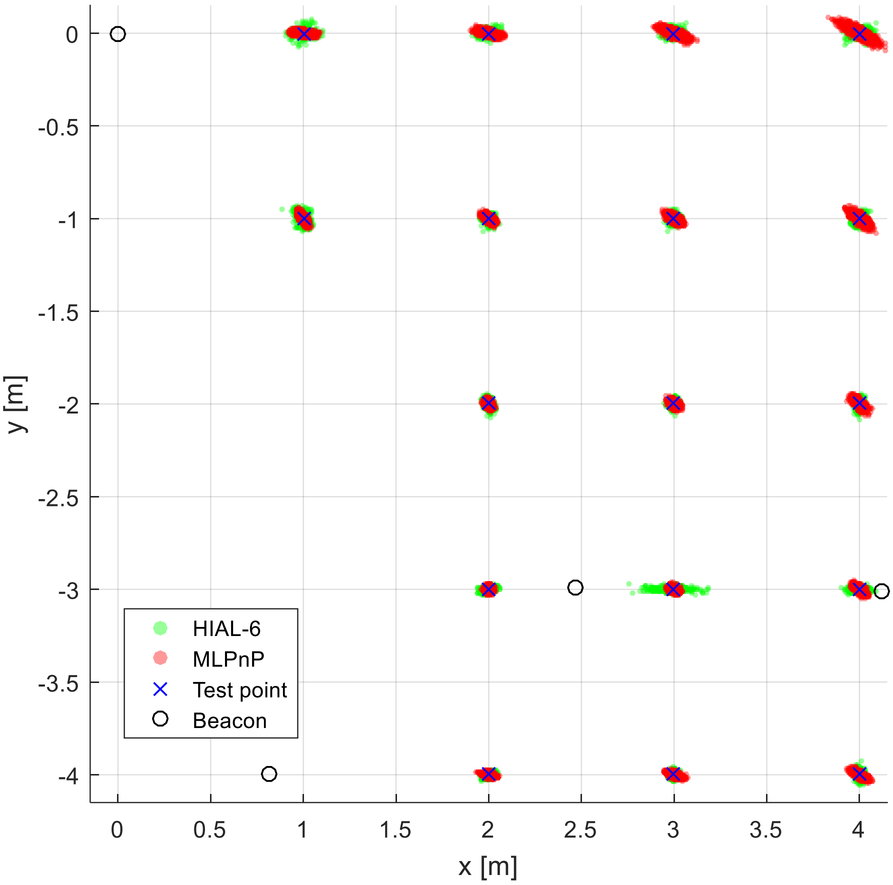


Figure 6.6 Effect of detection error on the 2D localization accuracy for MLPnP and HIAL-6, with $n = 2$ pixels.

The detailed measurement results are summarized in Table A.2 (due to lack of space HIAL-5 was omitted). Again, the localization error is approximately proportional with the detection error, 1 pixel of detection error resulting in approx. 1-1.5 cm of localization error. The number of beacons has moderate effect on the localization error: more beacons usually provide more accurate results.

6.1.3 Effect of reference position error

To analyze the effect of inaccurate reference positions, the following test was performed: for each test points the ideal detections were calculated (with no error). Then the reference positions were all perturbed, by adding a vector with random direction and length d to the true positions, with $d = 1, 2, 3, 4$ cm. The localization algorithm was run using the true detections and the perturbed reference locations. The resulted location error is illustrated in Figure 6.7 with $d = 4$ cm, using MLPnP and HIAL-4.

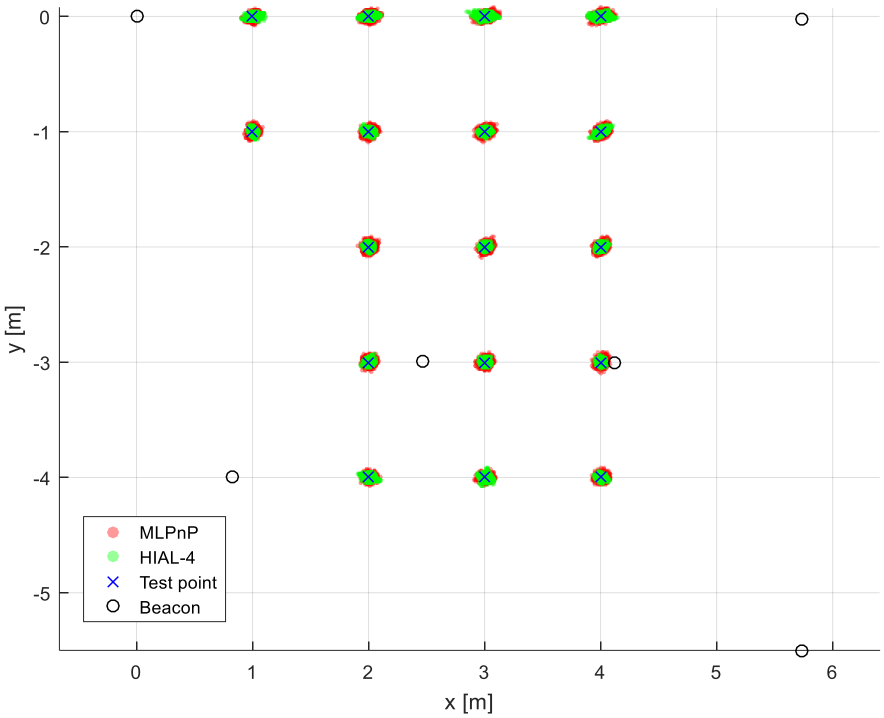


Figure 6.7 Effect of reference position error on the 2D localization accuracy for algorithms MLPnP and HIAL-4, with $d = 4$ cm.

The detailed results are listed in Table A.3. Clearly, increasing amount of error in the reference positions caused proportionally increasing localization error, in the test case 1 cm uncertainty in the reference position caused approximately 1-1.5 cm of localization error. The number of beacons had no significant effect on the accuracy. Notice that in general the MLPnP algorithm was less sensitive to reference position errors: the mean and maximum values tend to be lower in this case. This is possibly due to the applied position estimation method: while MLPnP approximates the maximum likelihood solution in 3D, the HIAL algorithms compute the 2D solution first, and extend it to 3D in a separate step (see main part of Section 6).

6.1.4 Effect of outliers

False detections can happen e.g., when a reflected beacon is observed. False detections can be anywhere in the detected image; thus, they can be modelled as random outliers. The test was performed in the following way.

Since the reference algorithm MLPnP requires at least 6 beacons, for fair comparison altogether 7 beacons were utilized, as follows: the locations of the original 6 beacons $B_1, B_2, \dots, B_6$ were always correctly detected, while beacon B_7 (see Figure 6.8 for its location) was always incorrectly detected.

The detections for the test were simulated as follows: for all test points the ideal detection was computed for all beacons (no other error source was utilized) and then the detected position of B_7 was modified randomly within ±(50-100) pixels, using uniform distribution. The localization algorithms were run using the modified detection and the ideal beacon positions. For each test point 800 independent experiments were carried out.

The tested HIAL algorithms utilized the original beacon configurations plus always used beacon B_7. E.g., now HIAL-5 utilized 4 good beacons (as in earlier HIAL-4 configuration) and B_7. Accordingly, in the tests HIAL-4 … HIAL-6 were tested, along with MLPnP, which utilized all seven beacons.

The behavior of the algorithms MLPnP and HIAL-4 is illustrated in Figure 6.8. As the plotted results clearly show, MLPnP is very sensitive to large measurement errors: even if 6 of 7 detections were correct, the one outlier caused large bias (~0,5 m) of the location estimate. Algorithm HIAL-

4, however, always provided perfect results (since the other simulated detections were free of error, the localization error was always practically zero, except for small rounding errors).

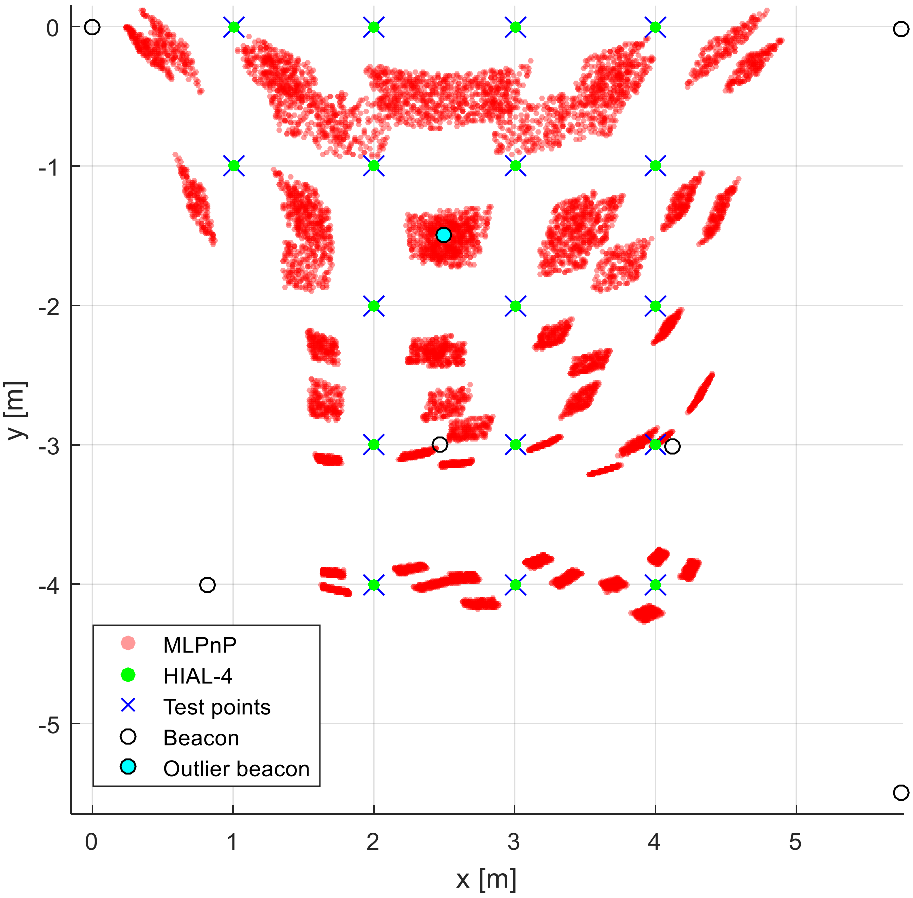


Figure 6.8 Effect of one outlier detection on the 2D localization accuracy, for algorithms MLPnP and HIAL-4.

The test results for all test cases are shown in Table A.4. While MLPnP was clearly very sensitive to the outlier, all HIAL configurations provided perfect localization in all test cases (with practically zero error). Thus, the proposed HIAL algorithm is able to filter out bad detections and provide fault tolerant behavior, thanks to the approach utilized in data fusion: only consistent measurements (determined by parameters ε_1 and ε_2) are used in the estimation process and outliers are automatically neglected, while MLPnP needs to be used in RANSAC-based outlier rejection schemes.

6.2 Evaluation

Real measurements were performed in a room of size 6.5 m x 6 m, with height of approx. 3.1 m. The 6 beacons were deployed on the ceiling. The beacon locations in 2D are shown in Figure 6.5, and the z coordinates were 3.04 m for all beacons. The 17 reference positions for the static tests were measured on the floor, as shown in Figure 6.5. The positions were measured with a laser distance meter with estimated accuracy of 1 cm. Parameter ε_1 was set to 2 cm, parameter ε_2 was set to 6 cm.

6.2.1 Static accuracy

To test static accuracy of the algorithms 17 test points were used (see Figure 6.5). The camera was placed over each test point and at every point 800 measurements were performed. For HIAL-3 and the reference algorithm MLPnP, the results are illustrated in Figure 6.9. Both algorithms provided error in the range of a few centimeters. At the center test points the error is as low as 10-20 mm, while at some test points at the perimeter the error was systematically higher (60-80 mm) for both the HIAL and the reference MLPnP algorithms. The reason of the higher error is possibly the inaccurate beacon positioning, which has more significant impact at certain location, due to geometric dilution of precision.

Notice the slight variance of the estimated object positions even though the camera was not moving. This effect is due to the internal operation of the object detection: since the beacon is blinking, the size and shape of the detected object is changing in time, and thus the detected object's center position may slightly change. This effect is smoothed by the Kalman-filters of the Object Tracker block, but still there is a small variation in the detected object position, resulting slight variation in the estimated object position.

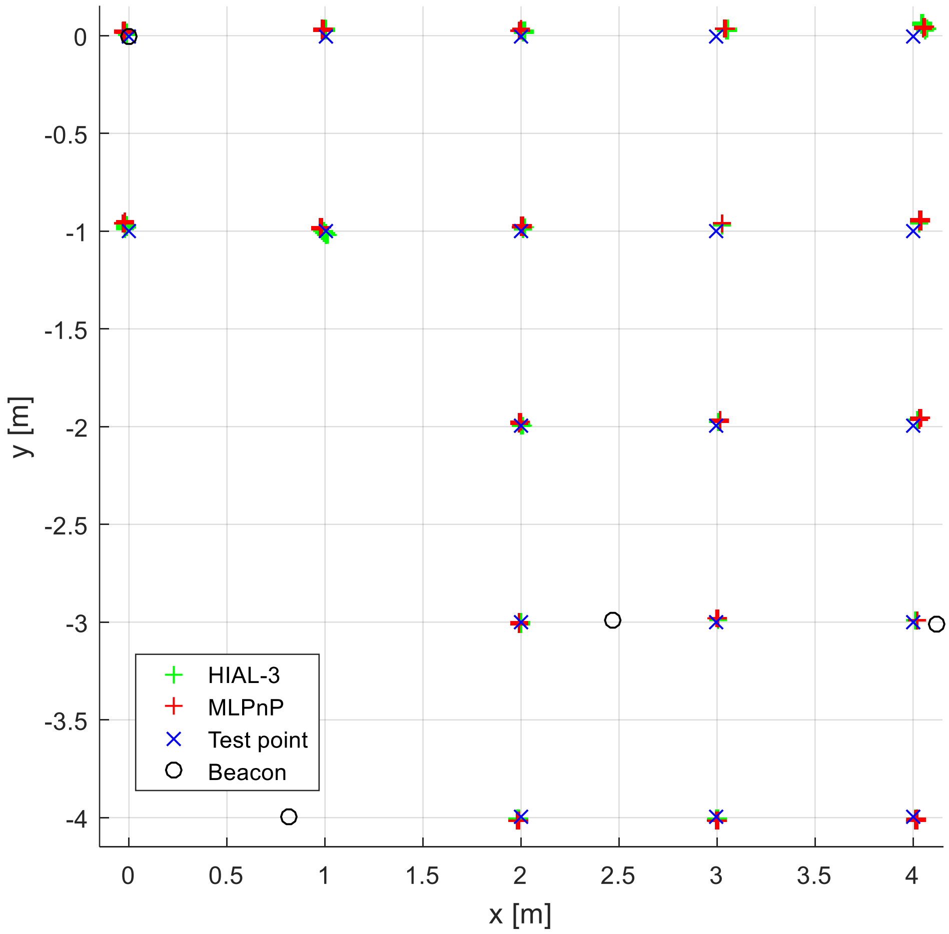


Figure 6.9 3D localization accuracy for MLPnP and HIAL-3, using real measurements

The detailed 3D test results are listed in Table A.5. The average accuracy of the system is around 3-4 cm. According to the results, the HIAL algorithm provides accuracy in the same range as the reference algorithm MLPnP, but with fewer (at least 3) beacons. When the number of beacons is higher than 3, the accuracy of HIAL is better than that of MLPnP. Notice that MLPnP is able to operate with arbitrary camera orientation, while HIAL requires that the camera faces upwards.

6.2.2 Dynamic accuracy

To test the accuracy of the system (see Section 1.4) while the tracked object is moving, another test setup was constructed. The camera was moved along a circular path, the radius of which was 1 m and its center was at test point T_{10}. The true trajectory is shown in blue in Figure 6.10. During the test the speed of the object varied between $0.1\ \mathrm{m/s}$ and $0.6\ \mathrm{m/s}$. The detected trajectories, according to MLPnP and HIAL-3 are plotted in red and green, respectively. There is a visible distortion along the path, which is again probably due to the inaccurate beacon positioning. The positioning errors along the track correspond well with that of the static case.

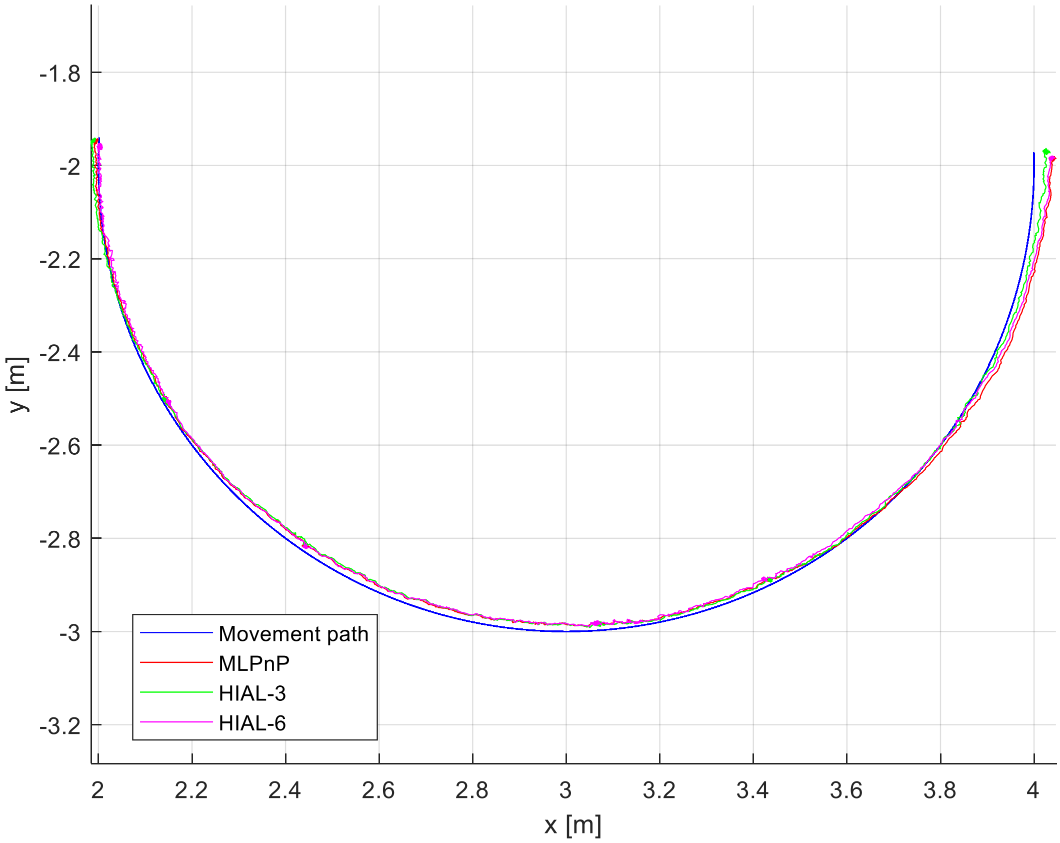


Figure 6.10 Tracking results of a moving object. The insert shows the 3D error of the HIAL and MLPnP algorithms.

The detailed statistical results are shown in Table A.6. As the results show, increased number of beacons provided somewhat better results; the mean error being around 2 cm. This result is slightly better than that of the static case, the reason of which is that the dynamic track was placed in the middle of the room, where the static error was also smaller.

To test the dependence of the location accuracy as a function of camera speed, the following test was performed. The camera was moved along a linear trajectory with constant speed. Five different speed values were used, and the measured 3D tracking errors are included in Table A.7. Interestingly, the increase of speed slightly increased the accuracy, especially in case of lower beacon numbers. This is probably due to the smoothing effect of the built-in Kalman filters: higher speed provides more efficient filtering effect against the random variance, present due to object detection (see Section 6.2.1). Notice that the proposed system is capable of tracking objects in real-time with high accuracy, even when the object is moving as fast as 1 m/s (which is a typical speed of forklift trucks).

6.3 Comparison and conclusion

The proposed method is compared to other VLC-based localization methods in Table 6.1. The compared methods include the exhaustive search (MEX) and least squares method (LSM) proposed in [17], the least squares solution of [81], the geometric consensus-based approach of [58], MLPnP [76], and HIAL-3.

	Mean error	Processing time	Test area	Number of Beacons	Fault tolerant
MEX [17]	3.2 cm	0.36 sec	1 m x 1 m	5	✗
LSM [17]	14.7 cm	0.001 sec	1 m x 1 m	5	✗
LSM [81]	18 cm	0.17 sec	5.4 m x 7.5 m	~15-20	✗
LookUp [58]	17 cm	~2 sec	15 m x 30 m	4	✓
MLPnP [76]	3.4 cm	< 0.001 sec	4 m x 4 m	6	✗
HIAL-3 [S1]	3.9 cm	~ 0.002 sec	4 m x 4 m	3	✓

Table 6.1 Comparison of various VLC localization methods.

The comparison includes the mean positioning error in real measurements, and the computation time for one estimate. To provide better comparison, the size of the test area and the number of beacons is also included. Methods resilient to faulty detections are also marked in Table 6.1.

The proposed HIAL method is able to provide one of the smallest localization errors, with the smallest number of beacons. The method, with higher number of beacons, is fault tolerant against false detections/outliers. The speed of the proposed algorithm is also remarkable for beacon counts occurring in a practical case (e.g., 12 beacons are visible ~ 12 msec with multithreading), allowing real-time calculation for realistic camera (e.g., with 30 Hz). However, the required amount of time needed to calculate the position scales badly as the number of beacons is increasing. Also, various error sources in the system may cause an increased number of generated intersection points, as it is possible that 2 arcs generate 2 intersection points in worst case, which slows down the scoring part of the method, especially when high number of beacons are present.

During the real measurements (see Figure 6.9 and Figure 6.10) the center point of each beacon was only approximated based on the camera image frames: First, the center point of the blob was determined based on the pixels in the distorted camera image, for which a better approach would be to first apply the coordinate transformation, based on the camera calibration data, to convert each corresponding pixel into a unit direction vector in the camera's local coordinate system, then calculate the mean unit direction vector of them. Second, for higher accuracy, the beacon's pre-measured central coordinate should be compensated to be in the position where the mean direction vector would intersect it (within the bounds of the beacon), which is not applicable with this method. The nature of the error is systematic with respect to the position of the camera, and the actual beacons geometrical alignment, which can be compensated with post-processing the resultant estimate. This size of the error is proportional with the size of the beacon in the camera's image, and with its real-world size, and inversely proportional with the distance between the camera and the beacon, and in cases, the camera's resolution. Also, choosing the correct beacon size and camera based on the environmental properties the error causing the distortion can be reduced into a negligible amount.

Due to the score-based selection of the intersections, which are used to generate the output estimate, this method may be sensitive to noisy measurements, since it can cause the intersection-intersection and intersection-arc distances to wiggle around the method's thresholds (ε_1 and ε_2) when they are not properly set. The appearance of a new beacon or the removal of a previous one may also cause sudden jumps in the estimate. Thus, additional smoothing/filtering may be necessary for the output estimate.

In the next section a random sample consensus-based approach will be introduced, which is not only tolerant against outliers, but possesses better computational performance, and is able to provide a smoother output estimate.

7 Novel RBL ADoA positioning algorithm

The proposed solution in Section 6 is able to provide outlier-tolerant, robust position and orientation estimates most of the time, however, after extensive usage and testing it was found that in certain beacon alignments the distance-based scoring method may provide noisy estimates if not properly set (see e.g. Figure 6.6). Let us consider the case, when the measured beacon angles are noisy. In this case the constructed arcs may be larger or smaller, providing 0 or 2 intersection points with other arcs instead of the optimal 1 (error-free case). Also, if the filter parameters are not properly set, it may also happen that when 2 intersection points are generated from 2 arcs, only one of them will contribute to the largest group. Finding a good value for the ϵ_1 and ϵ_2 parameters is also cumbersome, and would require a trial-and-error approach, or simulations.

The calculation performance of the HIAL algorithm is sufficient for practical cases (~10 beacons are visible at a time), but scales badly: from every possible beacon pair an arc is generated; from every possible arc pair 0-2 intersection points are generated; we also need to compare the intersection points with each other, and with the non-constructing arcs. A conventional laptop for example, can handle ~10-15 beacons to calculate 30 estimates per second (real-time operation for a camera sampling with 30 Hz), but it will struggle when the beacon count is greater, leading to dropped estimations (the process is still computing and will skip e.g., every second estimation resulting in a larger delay, and breaking the real-time requirement of the system).

These problems urged the development of the RBL method, which is presented in this section. The proposed solution operates with the same principle (ADoA) and requirement (the normal vector of the camera is known) as HIAL (see Section 6 for details). It requires angle measurements of at least 3 beacons. With minimal number of beacons an algebraic solution is provided, which is naturally sensitive to outliers. When redundant (i.e., four or more) measurements are present, random sample consensus (RANSAC) is used to select the largest possible consistent beacon set to calculate a robust position estimate. A practical error model for cameras is proposed to determine the position-dependent inlier angle tolerance of the beacons. The performance of the proposed solution is illustrated by simulation examples and practical measurements as well.

7.1 Error model

The beacons are detected on the camera's image stream. On the left-hand side of Figure 7.1 a bird's-eye view of the area is shown, with the camera C', beacons P_i', P_j', P_k', P_l', and an obstacle partly covering beacon P_i'. On the right-hand side the fish-eye camera image is shown, with two possible sources of detection error.

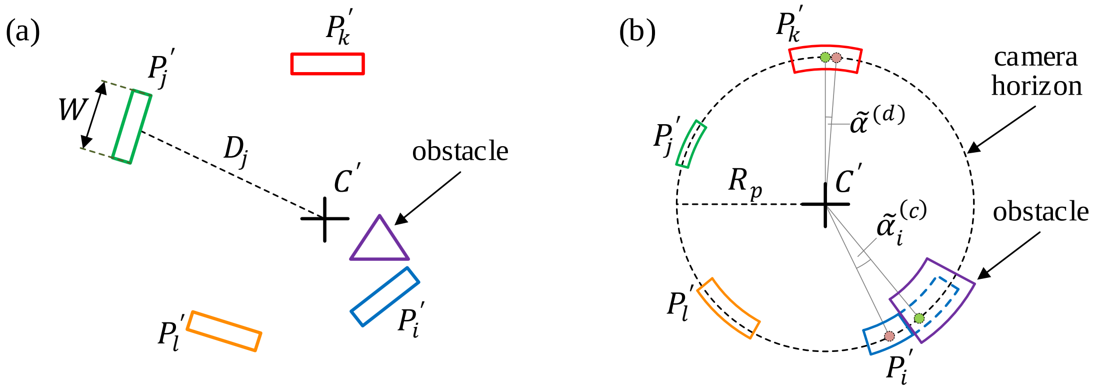


Figure 7.1 Detection error sources. (a) Bird's-eye view of the detection area. (b) Fisheye camera image. The detected and true beacon centers are denoted by red and green dots.

First, the center estimate of a beacon's image may be inaccurate, as illustrated on P_k'. This center detection error ΔC_p is usually small (around one pixel), and if all beacons are approximately on the horizon, the angle error is independent of the distance between the beacon and the camera.

This error is modeled as a random additive error $\tilde{\alpha}^{(d)}$ with uniform distribution between $-\Delta\alpha_d$ and $+\Delta\alpha_d$, where $\Delta\alpha_d$ is constant. If the radius of the horizon on the camera image is R_p pixels, then the value of $\Delta\alpha_d$ can be approximated as follows:

$$\Delta\alpha_d = \operatorname{atan2}(\Delta C_p, R_p)\,, \tag{7.1}$$

where atan2 is the two-argument arctangent function.

The second type of detection error may happen when a beacon is partially obscured (see P_i' in Figure 7.1). In this case the camera image contains a misshapen beacon, the center of which may have an offset with regard to the true center. The error is the largest when almost the whole beacon is covered; in this case the center of the observed beacon may appear $W/2$ away from the true position, where W is the horizontal size of the beacon. The resulting angle error may be higher if the beacon is closer to the camera (i.e., the beacon image is larger). This angle error is modeled as a random additive error $\tilde{\alpha}_i^{(c)}$, with uniform distribution between $-\Delta\alpha_{c,i}$ and $+\Delta\alpha_{c,i}$, where $\Delta\alpha_{c,i}$ is the maximal angle error as follows:

$$\Delta\alpha_{c,i} = \operatorname{atan2}\left(\frac{1}{2}W, D_i\right), \tag{7.2}$$

where W is the horizontal size of beacon P_i', and D_i is the distance between P_i' and the camera.

The measured angle of beacon P_i' is modelled as follows:

$$\alpha_i = \alpha_i^{(true)} + \tilde{\alpha}^{(d)} + \tilde{\alpha}_i^{(c)}, \tag{7.3}$$

7.2 RANSAC-based localization process

In this section the operation of the proposed localization algorithm is briefly summarized. The technical details will be discussed later; the summary contains references to sections with the detailed discussion. The pseudo-code of the proposed algorithm is the following:

```
Repeat Steps 1-4 $N_Q$ times. The number $N_Q$ of iterations is discussed in Section 7.2.1.
    Step 1: Select 3 beacons randomly. Let the selected beacon triplet be $BT_j$.
    Step 2: Calculate initial position estimate $\hat{C}_j$ and orientation estimate $\hat{\varphi}_j$ using $BT_j$. The
            estimates are calculated from the positions of beacons $BT_j$ and the measured angles
            of these beacons. The details of the estimation are presented in Section 7.2.2.
    Step 3: For all the other beacons, check whether their measurements are consistent with $\hat{C}_j$
            and $\hat{\varphi}_j$. The set of supporter beacons of $BT_j$ (i.e., the inliers) is denoted by $SUP_j$.
            The proposed inlier-outlier classification method uses a position-dependent
            threshold; thus, the effect of the measurement errors will be location-independent.
            The method is discussed in Section 7.2.3.
    Step 4: Calculate the number of supporter beacons $NS_j = \mathrm{n}(SUP_j)$ and their mean squared
            angle measurement error $E_j$, as described in Section 7.2.3.
End Repeat
    Step 5: From the $N_Q$ experiments select the triplet with the highest value $NS_k$. If multiple
            triplets exist with the same $NS_k$, then select the one with the smallest value $E_k$.
    Step 6: Using all of the supporter beacons in $SUP_k$, calculate refined estimates $\hat{\hat{C}}_k$ and $\hat{\hat{\varphi}}_k$,
            starting from $\hat{C}_k$, as described in Section 7.2.4.
```

7.2.1 Number of iterations

The estimation process uses random beacon triplets to find an initial position estimate. In order the estimator algorithm to provide a good estimate the measurement must fulfill the following two conditions:

a) All three initial beacons must be detected correctly, i.e., with small measurement error. If some of the initial measurements are outliers, the initial estimate will be wrong.

b) According to the phenomenon called geometrical dilution of precision (GDOP), some setups are more sensitive to measurement errors than others [82]. Thus, the geometry of the initial beacons and the camera must allow a good estimate when the measurements are correct (i.e., the measurement error is small, according to condition (a)).

In order for the estimator algorithm to provide a reliable estimate, it is necessary to have at least one good initial triplet that fulfills both requirements among the N_Q experiments. Since beacon triplets BT_j are chosen randomly, it is possible to determine the probability Pr_{hit} of having at least one good beacon triplet among the randomly chosen N_Q triplets. On the reverse direction, we can use this design parameter Pr_{hit} to determine the necessary number of iterations N_Q [83].

Let us denote the number of beacons and the number of total experiments by N_B and N_Q, respectively. Let us suppose that out of N_B measurements N_{GOOD} are inliers. The probability of choosing a reliable triplet is

$$Pr\left(BT_j \; is \; reliable\right) = \frac{\binom{N_{GOOD}}{3}}{\binom{N_B}{3}} P^*, \tag{7.4}$$

where P^* is the conditional probability of providing a good estimate if all of the initial measurements are inliers [83].

The probability Pr_{hit} of having at least one reliable triplet out of N_Q trials can be expressed as follows:

$$Pr_{hit} = 1 - \left(1 - \left(\frac{\binom{N_{GOOD}}{3}}{\binom{N_B}{3}}\right) P^*\right)^{N_Q} \tag{7.5}$$

Thus, the number of necessary trials, given probability Pr_{hit} and P^*, is the following:

$$N_Q = \frac{log(1 - Pr_{hit})}{log\left(1 - \left(\frac{\binom{N_{GOOD}}{3}}{\binom{N_B}{3}}\right) P^*\right)} \tag{7.6}$$

In practice the number of inlier/outlier measurements is usually unknown. However, it is usually possible to give a safe upper bound for the outliers (e.g., if there are rarely more than one outlier, then a safe assumption is that $N_{GOOD} \geq N_B - 2$, and an even safer assumption is that $N_{GOOD} \geq N_B - 3$). If the number of inliers is at least N_{GOOD}, then the calculated N_Q, according to (7.6), provides an upper bound for the required number of trials.

The probability Pr_{hit} is a design parameter, set close to 1 (e.g., 0.999). Probability P^* can be estimated using simulations, using the inlier angle tolerance value, as input parameter [83].

For small number of beacons, it is feasible to use all the possible beacon triplets. Even for 15 visible beacons, the number of trials is only 455 (see the measurement speed in Section 7.3.1). However, for higher number of beacons (6) may really be helpful: e.g., for 50 visible beacons the total number of trials would be 19600, which is not feasible for real-time applications. In the tests, we had a small number of beacons and thus used all possible triplets.

7.2.2 Calculation of the initial estimate

For the sake of simplicity, and without loss of generality, in iteration step j let us denote the selected beacons by $BT_j = \{P_1', P_2', P_3'\}$. Our goal is to determine the 2D camera position C_j' and orientation φ_j. For simpler notation for now, iteration index j will be omitted, and the estimates will be denoted by $\hat{C}$ and $\hat{\varphi}$. Let us use the following notations:

$$\boldsymbol{v}_{\boldsymbol{i}}' = P_i' - C', \tag{7.7}$$

$$\alpha_{i,j} = \angle P_i'C'P_j' = \alpha_j - \alpha_i \tag{7.8}$$

where $i, j = 1,2,3$. Let us choose two beacons (e.g., P_i' and P_j'), and suppose that $\alpha_{i,j} \neq 0$ and $\alpha_{i,j} \neq \pi$. Let us express the square of their distance $d_{i,j}$, using the law of cosines, as follows:

$$d_{i,j}^2 = \|\boldsymbol{v}_{\boldsymbol{i}}'\|^2 + \left\|\boldsymbol{v}_{\boldsymbol{j}}'\right\|^2 - 2\|\boldsymbol{v}_{\boldsymbol{i}}'\|\left\|\boldsymbol{v}_{\boldsymbol{j}}'\right\| \cos\alpha_{i,j}\,. \tag{7.9}$$

Using the fact that $\boldsymbol{v}_{\boldsymbol{i}}'$ and $\boldsymbol{v}_{\boldsymbol{j}}'$ are on the same x-y plane, the equality $\boldsymbol{v}_{\boldsymbol{i}}' \times \boldsymbol{v}_{\boldsymbol{j}}' = \boldsymbol{n}_{\boldsymbol{z}}\|\boldsymbol{v}_{\boldsymbol{i}}'\|\left\|\boldsymbol{v}_{\boldsymbol{j}}'\right\| \sin\alpha_{i,j}$, holds, where $\boldsymbol{n}_{\boldsymbol{z}}$ is the unit vector in direction of axis z. In this case (7.9) takes the following form:

$$d_{i,j}^2 = \|\boldsymbol{v}_{\boldsymbol{i}}'\|^2 + \left\|\boldsymbol{v}_{\boldsymbol{j}}'\right\|^2 - 2\frac{\left[(x_i - x_0)(y_j - y_0) - (x_j - x_0)(y_i - y_0)\right]\cos\alpha_{i,j}}{\sin\alpha_{i,j}} \tag{7.10}$$

Substituting $d_{i,j}^2 = \left(x_j - x_i\right)^2 + \left(y_j - y_i\right)^2$ and $\|\boldsymbol{v}_{\boldsymbol{i}}'\|^2 = (x_i - x_0)^2 + (y_i - y_0)^2$, and using notation $c_{i,j} = \cot\alpha_{i,j}$, we get the following equation:

$$\begin{gathered} x_i x_j + y_i y_j + \left(x_j y_i - x_i y_j\right)c_{i,j} = \\ x_0\left[x_i + x_j + \left(y_i - y_j\right)c_{i,j}\right] + y_0\left[y_i + y_j + \left(x_j - x_i\right)c_{i,j}\right] - x_0^2 - y_0^2 \end{gathered} \tag{7.11}$$

Similar result is obtained if beacon pair B_j and B_k is used:

$$\begin{gathered} x_j x_k + y_j y_k + \left(x_k y_j - x_j y_k\right)c_{j,k} = \\ x_0\left[x_j + x_k + \left(y_j - y_k\right)c_{j,k}\right] + y_0\left[y_j + y_k + \left(x_k - x_j\right)c_{j,k}\right] - x_0^2 - y_0^2 \end{gathered} \tag{7.12}$$

By subtracting (7.12) from (7.11) the squared terms vanish and we get the following linear equation:

$$\begin{gathered} -x_j\,\xi_{k,i} - y_j\upsilon_{k,i} - \omega_{i,j}c_{i,j} + \omega_{j,k}c_{j,k} = \\ x_0\left[-\xi_{k,i} + \upsilon_{i,j}c_{i,j} - \upsilon_{j,k}c_{j,k}\right] + y_0\left[-\upsilon_{k,i} - \xi_{i,j}c_{i,j} + \xi_{j,k}c_{j,k}\right], \end{gathered} \tag{7.13}$$

where $\xi_{a,b} = x_a - x_b$, $\upsilon_{a,b} = y_a - y_b$, and $\omega_{a,b} = x_a y_b - x_b y_a$.

Now let us calculate (7.13), by substituting $(i, j, k) = (1,2,3), (2,3,1)$, and $(3,1,2)$. We get the following overdetermined linear equation system:

$$\boldsymbol{A} \cdot \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \boldsymbol{b}\,, \tag{7.14}$$

where

$$\boldsymbol{A} = \begin{bmatrix} -\xi_{3,1} + \upsilon_{1,2}c_{1,2} - \upsilon_{2,3}c_{2,3} & -\upsilon_{3,1} - \xi_{1,2}c_{1,2} + \xi_{2,3}c_{2,3} \\ -\xi_{1,2} + \upsilon_{2,3}c_{2,3} - \upsilon_{3,2}c_{3,1} & -\upsilon_{1,2} - \xi_{2,3}c_{2,3} + \xi_{3,1}c_{3,1} \\ -\xi_{2,3} + \upsilon_{3,1}c_{3,1} - \upsilon_{1,2}c_{1,2} & -\upsilon_{2,3} - \xi_{3,1}c_{3,1} + \xi_{1,2}c_{1,2} \end{bmatrix}, \tag{7.15}$$

and

$$\boldsymbol{b} = \begin{bmatrix} -x_2\,\xi_{3,1} - y_2\upsilon_{3,1} - \omega_{1,2}c_{1,2} + \omega_{2,3}c_{2,3} \\ -x_3\,\xi_{1,2} - y_3\upsilon_{1,2} - \omega_{2,3}c_{2,3} + \omega_{3,1}c_{3,1} \\ -x_1\,\xi_{2,3} - y_1\upsilon_{2,3} - \omega_{3,1}c_{3,1} + \omega_{1,2}c_{1,2} \end{bmatrix}. \tag{7.16}$$

If none two of the three beacons are on the same line with C (i.e. $\sin\alpha_{1,2} \neq 0, \sin\alpha_{2,3} \neq 0$, and $\sin\alpha_{3,1} \neq 0$) then $c_{1,2}$, $c_{2,3}$, and $c_{3,1}$ are finite and any two lines can be selected from $\boldsymbol{A}$ and $\boldsymbol{b}$ to construct $\boldsymbol{A}'$ and $\boldsymbol{b}'$, resulting in an equation system with two equations and two unknowns. The initial location estimate is the following:

$$\hat{C} = \begin{bmatrix} \hat{x}_0 \\ \hat{y}_0 \end{bmatrix} = (\boldsymbol{A}')^{-1}\boldsymbol{b}' \,. \tag{7.17}$$

If one pair of beacons is on the same line with C' (e.g., P_i' and P_j'), then $\sin\alpha_{i,j} = 0$. In this case only one row is selected from $\boldsymbol{A}$ and $\boldsymbol{b}$, the one which does not include the infinite term $c_{i,j}$. (E.g., if B_1, B_2, and C' are on the same line then $c_{1,2}$ would be infinite, thus only the second row is used from $\boldsymbol{A}$ and $\boldsymbol{b}$). In this case another equation must be constructed for $\boldsymbol{A}'$ and $\boldsymbol{b}'$; from (7.13) using limit $\sin\alpha_{i,j} \to 0$ the following equation can be obtained:

$$x_0 \upsilon_{i,j} - y_0 \xi_{i,j} = -\omega_{i,j}, \tag{7.18}$$

and again (7.17) can be used to calculate the position estimate.

If all three beacons and C' are on the same line (i.e., $\sin\alpha_{1,2} = \sin\alpha_{2,3} = \sin\alpha_{3,1} = 0$), then the position cannot be determined from this set of measurements. Also, there is no solution if all three beacons and C' are on the same circle ($\det(\boldsymbol{A}') \cong 0$).

In practice the angle differences $\alpha_{i,j}$ are checked before calculating $\boldsymbol{A}$ and $\boldsymbol{b}$: if any of them is closer to 0 or π than a small threshold (e.g., 1e-6) then (7.18) is applied. If all measurements are close to 0 or π then the triplet is ignored (since the position cannot be reliable determined), and the iteration is continued by selecting a different beacon triplet.

Once the initial position is estimated, the initial orientation estimate can be calculated as follows: Let us choose any of the three beacons, e.g., P_k'. From point C' beacon P_k' is observed in K_1 under angle

$$\beta_k = \operatorname{atan2}(y_k - \hat{y}_0, x_k - \hat{x}_0). \tag{7.19}$$

The observed angle in K_2 was α_k, thus the estimated orientation, according to Figure 6.2 (a), is the following:

$$\hat{\varphi} = \beta_k - \alpha_k. \tag{7.20}$$

7.2.3 Location-dependent inlier-outlier classification

In the previous step the initial position $\hat{C} = (\hat{x}_0, \hat{y}_0)$, and the initial orientation $\hat{\varphi}$ were estimated. If the initial estimates are correct then any beacon P_i' should be seen from ideal direction β_i, in the world coordinate system:

$$\beta_i = \operatorname{atan2}(y_i - \hat{y}_0, x_i - \hat{x}_0), \tag{7.21}$$

In coordinate system K_2 the ideal angle is

$$\alpha_i^{id} = \beta_i - \hat{\varphi}. \tag{7.22}$$

The observation error of beacon P_i' is the following:

$$\varepsilon_i = \alpha_i - \alpha_i^{id} = \alpha_i - \beta_i + \hat{\varphi}. \tag{7.23}$$

Let us use design parameters $\Delta\alpha_d$ and $\Delta\alpha_{c,i}$, according to (7.1) and (7.2), where

$$D_i = \sqrt{(x_i - \hat{x}_0)^2 + (y_i - \hat{y}_0)^2}, \tag{7.24}$$

and let us calculate *inlier angle tolerance* $\Delta\alpha_i$ for beacon P_i' as follows:

$$\Delta\alpha_i \; = 2\big(\Delta\alpha_d + \Delta\alpha_{c,i}\big). \tag{7.25}$$

Using (23) and (25), the location-dependent inlier/outlier classification is the following:

$$\begin{aligned} P_i' &\in SUP_j(inlier): && if\ |\varepsilon_i| \leq \Delta\alpha_i \\ P_i' &\notin SUP_j\ (outlier): && if\ |\varepsilon_i| > \Delta\alpha_i{}' \end{aligned} \tag{7.26}$$

where SUP_j is the set of supporter beacons for triplet BT_j. Notice that beacons in BT_j are naturally elements of SUP_j. The mean squared angle measurement error E_j [17], using the consistent measurements only, is the following:

$$E_j = \frac{1}{\mathrm{n}(SUP_j)} \sum_{P_i' \in SUP_j} \varepsilon_i^2. \tag{7.27}$$

7.2.4 Refined location estimate

While the algorithm can provide a usable pose estimate after selecting the best hypothesis, which has the largest number of supporters, it is not guaranteed, that it is a local optimum. It may happen, that we did not select the best initial beacon triplet (during the random selection); or the supporter beacons may be able to lower the error of the position estimate. Thus, the calculation of a refined estimate using not only the 3 initial beacons, but also the supporter beacons is necessary to find a locally optimal position estimate.

Let us construct error function $E(x, y)$ as follows: if the camera position were in (x, y) then beacon P_i' would be observed in direction β_i, in coordinate system K_1:

$$\beta_i(x, y) = \mathrm{atan2}(y_i - y, x_i - x), \tag{7.28}$$

From measurements α_i of the consistent beacons, the orientation estimate is refined as follows:

$$\hat{\hat{\varphi}}(x, y) = \frac{1}{\mathrm{n}(SUP_j)} \sum_{P_i' \in SUP_j} (\beta_i(x, y) - \alpha_i). \tag{7.29}$$

Thus, the observation error of beacon P_i', provided that the camera position is (x, y), is

$$\varepsilon_i(x, y) = \alpha_i - \mathrm{atan2}(y_i - y, x_i - x) + \hat{\hat{\varphi}}(x, y), \tag{7.30}$$

and the mean squared error function [17], similarly to (27), is the following:

$$E(x, y) = \frac{1}{\mathrm{n}(SUP_j)} \sum_{P_i' \in SUP_j} \varepsilon_i^2\,(x, y). \tag{7.31}$$

The refined position estimate $\hat{\hat{C}}_j$ is at the minimum of $E(x, y)$:

$$\hat{\hat{C}}_j = (\hat{x}_0', \hat{y}_0') = \underset{(x,y)}{\mathrm{argmin}}\, E(x, y), \tag{7.32}$$

and the refined orientation estimate $\hat{\hat{\varphi}}_j$ is the following:

$$\hat{\hat{\varphi}}_j = \hat{\hat{\varphi}}(\hat{x}_0', \hat{y}_0'). \tag{7.33}$$

The location estimate is calculated by the downhill simplex method on the error function $E(x, y)$, starting from initial position estimate $(\hat{x}_0, \hat{y}_0)$. The search finds a (possibly local) minimum (x_0', y_0') near the initial estimate. It is not guaranteed that the global minimum of (7.31) is found, but since the selected measurements of beacons in SUP_j are consistent and the search is started from a consistent initial estimate $(\hat{x}_0, \hat{y}_0)$, the iterative search in practice quickly and accurately finds the correct estimate, usually close to the initial value $(\hat{x}_0, \hat{y}_0)$.

7.3 Evaluation

The proposed method was evaluated using simulations (Section 7.3.1) and real measurements (Section 7.3.2).

7.3.1 Simulation results

The tests were performed in the simulated environment, shown in Figure 7.2. The size of the simulated room was 22 m x 22 m, where 12 beacons were deployed in the positions shown in Figure 7.2. Performance tests were made on a test grid containing 23 x 23 points, the grid coordinates being integer numbers in meters, between -11 m and $+11$ m.

The measurements were generated as follows. In each test point the ideal angles were calculated, then a random error between $-0.4°$ and $+0.4°$ was added to simulate the center detection error, which is equivalent to $\Delta\alpha_d = 0.4°$. This value corresponds to the case when the distance between the beacon's image and the center of the camera is ~150 pixels and the center detection error is $\Delta C_p = 1$ pixel. Also, a potential coverage error was simulated with beacon width value $W = 3$ cm, according to (2). In each test points 20 independent noise realizations (i.e., measurement errors $\tilde{\alpha}^{(d)} + \tilde{\alpha}_i^{(c)}$, according to (7.3), were generated.

The outliers were generated as follows: When one outlier was present, a random beacon was selected and a random angle measurement error in the range of $\pm[10°, 170°]$ was added. When $K \geq 2$ outliers were present then K beacons were randomly selected and their generated measurements were shuffled. This error simulates multiple incorrect beacon identifications and thus resulting false measurements.

The proposed method was compared with the method of exhaustion (MEX), proposed in [17], applied for the 2D case. The MEX method utilizes error function E_{ref}, which is similar to the error function of the proposed method:

$$E_{ref}(x, y) = \frac{1}{N_B} \sum_{i=1}^{N_B} \varepsilon_i^2 (x, y). \tag{7.34}$$

MEX uses an exhaustive (brute-force) search on a grid and is guaranteed to find the optimal position on the search grid. In the tests a search grid was used with resolution of 2 cm for MEX to find the smallest error function value. On the other hand, the proposed algorithm is not restricted to a search grid, but finds the solution anywhere on the search plane, without any guarantee for global optimum. Since the error functions are similar (see (7.31) and (7.34)), the expected behavior of the two algorithms is the same, when there are no outliers.

Since the number of beacons is low ($N_B = 12$), in the tests all possible beacon triplets were used, i.e., $N_Q = \binom{N_B}{3} = 220$.

In the first test measurement noise was added for the ideal angle measurements, but no outliers were present. Figure 7.2 shows the position errors for the proposed and the reference methods. The performances of the algorithms are very similar, as it was expected. The error is small, mainly below 10 cm. At certain points of the search space slightly higher errors can be observed: this is because of the poor GDOP; here 8 of the beacons (P_1',P_3', P_4', P_6', P_7', P_9',)($P_1', P_3', P_4', P_6', P_7', P_9', P_{10}', P_{12}'$) are on the same circle, while the other 4 beacons are also close to this circle.

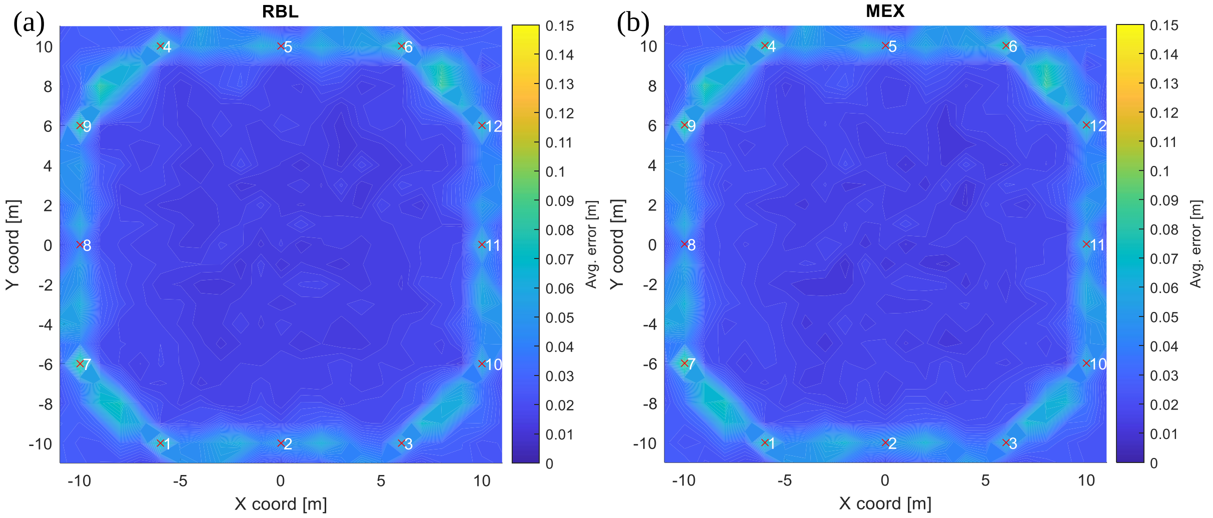


Figure 7.2 Average position errors for the proposed RBL method (a) and the reference MEX method (b) for simulated noisy measurements without outliers.

To allow detailed analysis of the errors, the error distributions are shown in Figure 7.3, using all of the experiments (23 x 23 x 20 = 10580 test runs). The behavior of the two algorithms is apparently quite similar, according to the expectations. The average error was 2.65 cm and 2.67 cm for the proposed and reference methods, respectively. The slight difference is probably due to the fact that the reference method was evaluated on a 1 cm grid, while the proposed method provides results on the continuous plane.

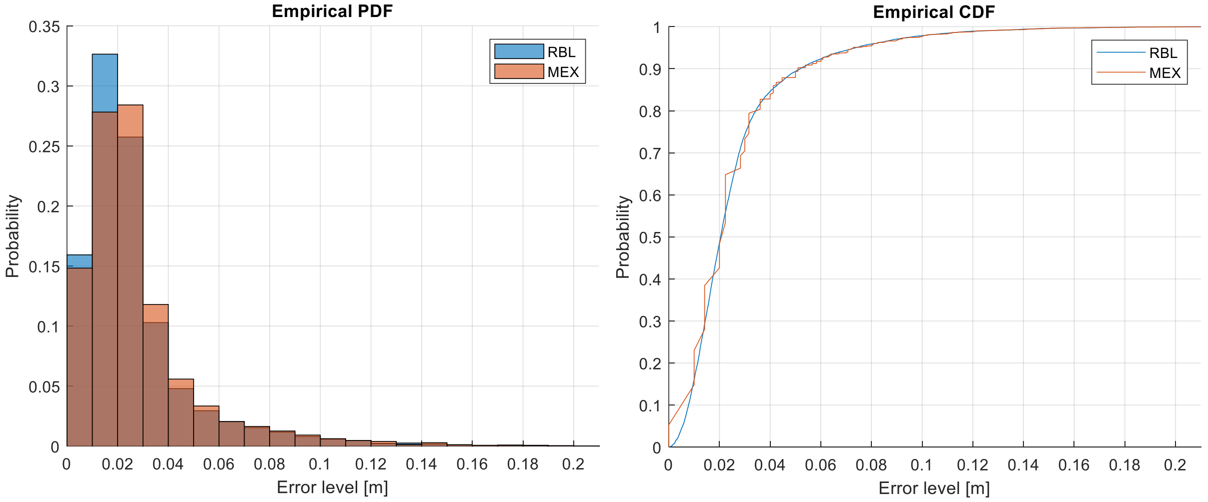


Figure 7.3 Error distribution of the proposed RBL method and the reference MEX method for simulated noisy measurements, without outliers. Empirical PDF is illustrated as a histogram with bin size of 1 cm.

In the next experiments 1, 3, and 5 outlier measurements were included. The position errors in the search area, for 1 outlier, are shown in Figure 7.4. The results for the reference method clearly show the dramatic effect of the outlier: most of the position estimates have significant error now, up to a few meters. The position errors of the proposed method did not change significantly, illustrating the effectiveness of the outlier detection and removal.

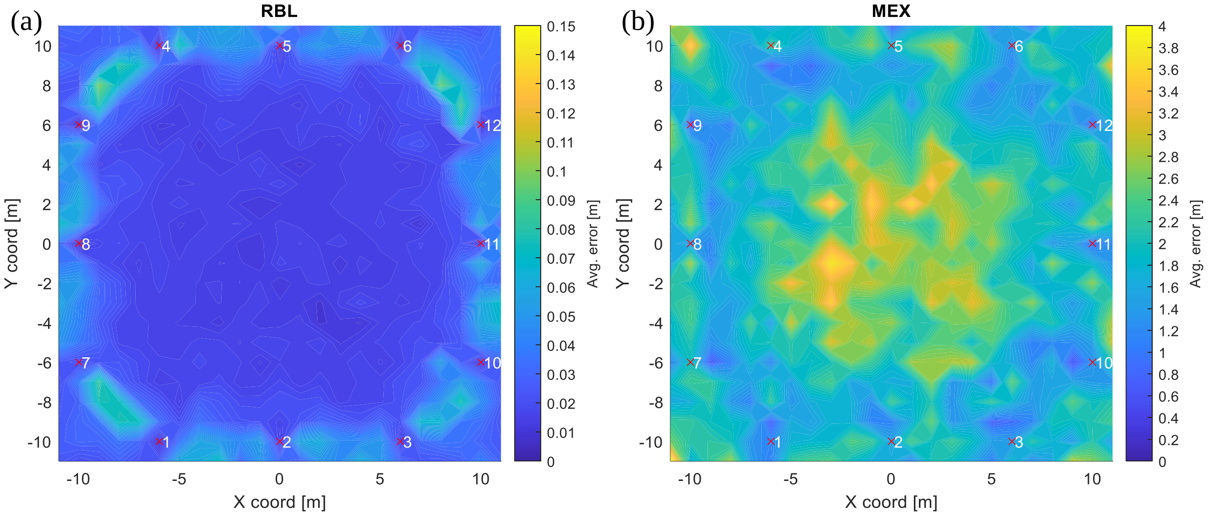


Figure 7.4 Position errors for the proposed (a) and the reference (b) methods for simulated noisy measurements with 1 outlier.

The position error distributions are shown in Figure 7.5, for 1, 3, and 5 simultaneous outliers. The reference method, as shown in Figure 7.5 (b), cannot provide meaningful estimates in the presence of outliers. Even a single outlier can destroy the estimate, but more outliers affect the estimation quality more severely.

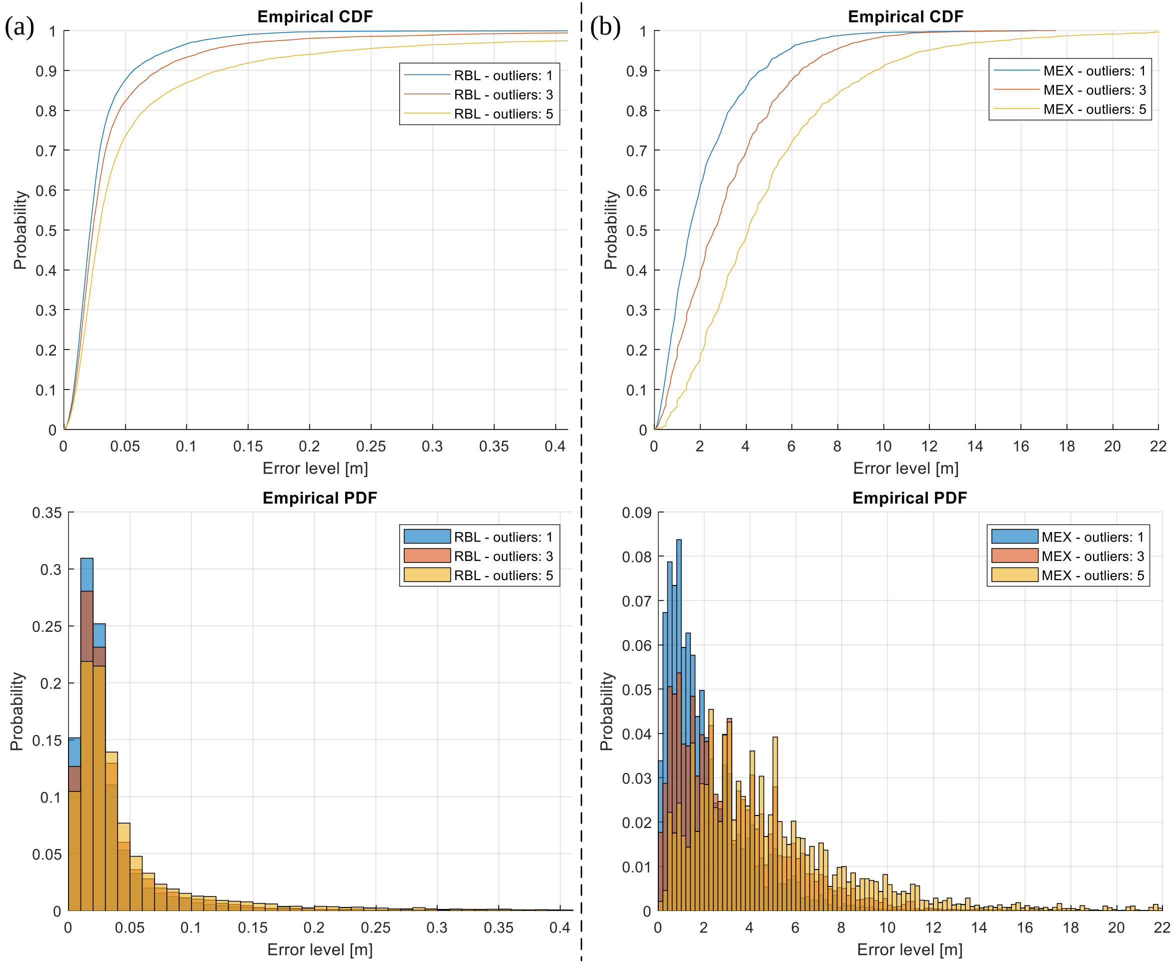


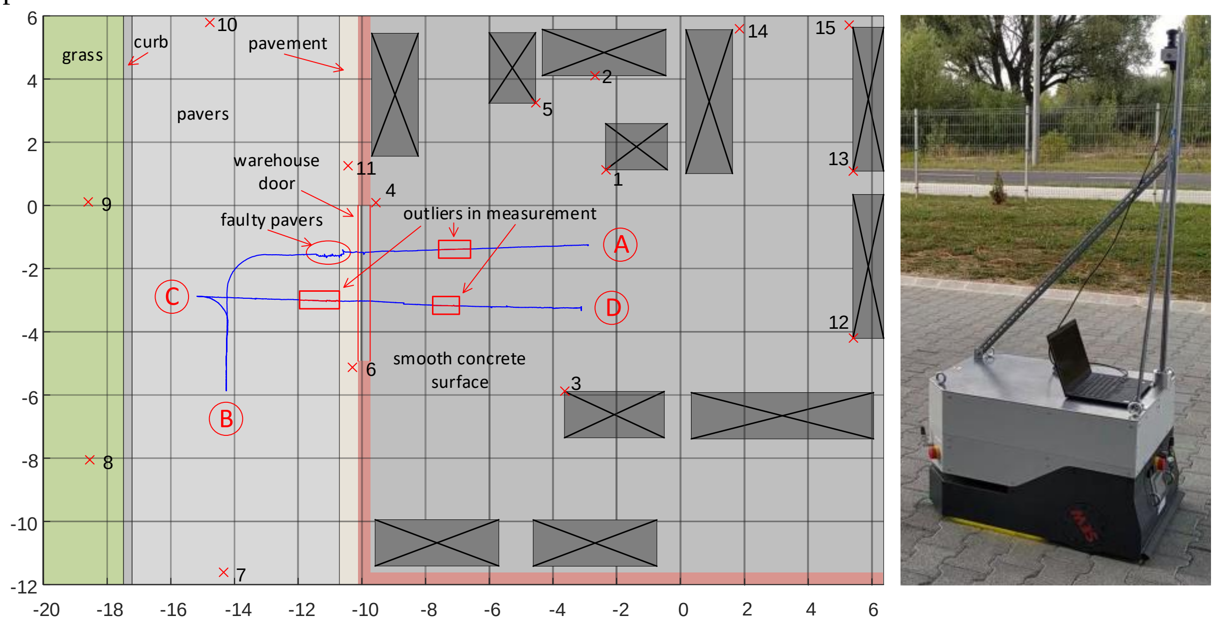
Figure 7.5 Error distribution of the proposed RBL method (a) and the reference MEX method (b) for simulated noisy measurements and outliers. Empirical PDF is illustrated as a histogram with bin sizes of 1 cm (a), and of 20 cm (b).

For the proposed method the quality of the estimate did not change significantly, as shown in Figure 7.5 (a). However, there is a slight shift towards larger errors when outliers are present. The reason is the following: when more outliers are present then there are fewer correct measurements that can be used in the estimation process, thus the accuracy of the estimation naturally decreases. But since there are no outliers included in the estimation, the quality degradation is only minor.

The computational costs between the proposed and the reference method were compared. Both methods were implemented in Matlab and run on a conventional laptop with Intel Core i7-9750H processor. The proposed method (with 12 beacons, using all possible beacon triplets) required 56 microseconds in average to compute an estimate, while for the reference MEX method the average runtime was 278 milliseconds.

7.3.2 Measurement results

The test measurements were made inside of an industrial warehouse and in front of the warehouse. The map of the test area is shown in Figure 7.6 (a). On the right-hand side a part of the warehouse is visible, where nine beacons were deployed along the walls and on the shelves. There were another six beacons deployed outside the building on the walls and poles. The camera was deployed on top of the vehicle, as shown in Figure 7.6 (b). An estimated path of the vehicle is shown in Figure 7.6 (a) in blue: the vehicle started from point A in the warehouse, exited the building, crossed a bumpy part of the pavement and turned left to point B. Here the vehicle moved backwards and made another left turn to point C. From point C the vehicle entered the building on a straight trajectory to the final point D.



(a) (b)

Figure 7.6 (a) Map of the measurement environment. On the right-hand side there is a warehouse, the left-hand side shows the open-air area in front of it. The beacons are denoted by numbered crosses. The estimated vehicle route A-B-C-D is shown in blue. (b) Photo of the vehicle with the installed camera on top of it.

During the measurements the number of visible beacons varied from 3 to 13. Apart from the bumpy pavement, where the camera swayed significantly on the top of the vehicle, the trajectory estimate is smooth and contains variation of a few centimeters only. During the test, the algorithm detected outliers (due to reflecting surfaces in the warehouse) at three different path segments, in total of ~550 image frames, as marked in Figure 7.6 (a). The outliers had no significant effect on the position estimates which clearly illustrates the fault-tolerance of the proposed system.

7.4 Conclusion

Measurements with large error (i.e., outliers), usually resulting from incorrect detections or reflections, may cause large error in the position estimate. Such outliers must be removed from the data used in the estimation process, to avoid incorrect estimates. Popular methods include model-based approaches (using e.g., Kalman-filters) [84], statistical methods [85], consistency-based approaches [58], and Random Sample Consensus (RANSAC) [86]. RANSAC [87] is popular because it iteratively adds inliers to and excludes outliers from a small starting set of measurements, thus provides robust estimates with reasonable amount of computation.

I proposed a robust RANSAC-based solution for 2D position estimation using azimuth-only angle measurements. The proposed method uses a practical error model and a corresponding position-dependent inlier angle tolerance to detect outliers. The initial estimate is calculated by a linearized equation system, using measurements from 3 beacons. The number of supporter beacons is increased incrementally using RANSAC. The best initial estimate (with the largest set of supporters and the smallest error function value) is used as an initial point for the downhill simplex method to find the final position estimate; in the search those measurements are used which are consistent with the winner initial estimate.

The performance of the proposed method was illustrated by simulation examples. It was shown that the proposed method and the reference MEX method has essentially the same accuracy when no outliers are present. This is due to the similar cost functions used by the two methods. However, in the presence of outliers, the reference method provided no usable results, since it used the bad measurements as well for the estimation, generating large estimation errors. The proposed method was not sensitive to outliers: its performance degradation was minor, even when as much as 5 simultaneous outliers were present.

The computational cost of the proposed method is low: in the realistic test cases the computational time of one position estimate (including refinement) was as low as 56 microseconds on a conventional laptop. The speed of the reference method, due to its exhaustive (brute-force) nature, was more than three orders of magnitude lower.

The performance of the proposed method was tested using real measurements as well, in an industrial warehouse. During the tests, the outliers had no effect on the position estimates, allowing positioning with only a few centimeters of error.

Compared to HIAL (presented in Section 6), the RBL method (presented in Section 7) uses location-independent, angle-based filtering of outlier measurements instead of distance-based ones, as a more practical approach. The calculation speed for HIAL is ∼28 msec for 12 beacons on a conventional laptop, while the RBL method needs only ∼56 μsec to provide an estimate, which makes it ~500x faster. Also, with the help of the location-refinement step, the estimate provided by RBL is much more consistent (smooth), than HIAL’s.

The issue of beacon center point approximation in the camera’s image, mentioned in Section 6.3, is also present in these physical measurements. As opposed to HIAL, however, this proposed solution makes it possible to adjust the error function used at the refined location estimation stage to use a compensated beacon coordinate based on: the current search point that is being evaluated; and the properties of the LED light fixture (e.g., size, shape, orientation in real-world coordinate system). The compensation method, however, is yet to be implemented.

8 Summary

8.1 New results

The first part of the thesis (Sections 2-3) presented the detailed error analysis of two visible light communication protocols (UFSOOK, UPSOOK) found in the literature, which can be used to provide a simple, yet effective method to identify anchor points used in the localization system. By understanding the effect of various error sources on the symbol transmission of the OOK protocols, the parameters of the communication methods can be designed to achieve ideal detection rate given the environmental conditions, and requirements (e.g., flicker-free operation). For the two VLC protocols the following design rules can be stated in general: the frequency offset (error) should be kept small, since the minimum error depends on this value; the threshold should be close to the ideal value (half amplitude of the detected signal); using small exposure time minimizes the effect of suboptimal thresholding; the design parameter of the two protocols should be kept small, to provide large symbol times (low number of signal transitions, where the unsafe regions can be found), while also satisfying the flicker-free requirement; the signal amplitude should be as high as possible to provide a wide range for optimal threshold, and also to provide high SNR. I showed that the effect of noise is the lowest at the optimal threshold, and the highest near the signal's extrema. The performance of UFSOOK was analyzed using the bit error rate of distinct symbols. The performance of UPSOOK was analyzed using a worst-case analysis to obtain the guaranteed minimum performance of the protocol by calculating the highest possible packet error rate. The results of the UFSOOK analysis for calculating the BER can also be used for UPSOOK, after proper adjustments (e.g., symbol time, symbol's rule set, offset parameter δ, etc.). As it was found, the transmission of these protocols will inevitably fail from time to time, when operated without synchronization, and using global shutter cameras.

The second part of the thesis (Sections 4-5) contained my contributions that addresses the problem of trackability and robust identifiability of beacons for visible light-based positioning systems. The beacon design of TUPSOOK makes it possible to track beacons in the camera's image stream providing reliable reference points for positioning. The protocol's encoding scheme provides means to identify the beacon in the image with using only one frequency component for symbols, and two signal states. The beacon design consists of two white LEDs: a small circular one, and a larger ring one around it. This beacon design also makes it possible, to use the LED beacon to illuminate the environment. Other approaches use e.g., multiple colors, or multiple separate LEDs, which either makes the beaconing infrastructure not suitable for proper illumination purposes, or the LED beacons are hard to detect/track reliably, when the camera is moving. The proposed equivalent sampling-based RUPSOOK protocol deals with the problem of delayed beacon identification that occur from time to time due to the unsynchronized case of the transmitter and the receiver. RUPSOOK's robustness lies in the wide range of system parameters where the operation of the data transmission is guaranteed to be error-free. The price for its robustness is the decreased transmission data rate of ~0.28 bits/sample (UFSOOK ~0.5 BPS, UPSOOK ~1 BPS). Since RUPSOOK also uses only two LED states for its operation, it can be integrated with the trackable beacon design of TUPSOOK, achieving a VLC protocol that is not only robust, but its beacons are also trackable.

The third part of the thesis (Sections 6-7) presented two proposed positioning algorithms that aims to provide robust location and orientation estimates by eliminating the threat of outlier measurements. HIAL uses a heuristic geometric approach, where the geometrically constructed possible location targets are scored based on two distance-based filtering parameters. The HIAL method is able to provide outlier-tolerant operation in most cases, but finding the optimal value for its filtering parameters is not straightforward (requires trial-and-error method or simulations). Also, the performance of the protocol scales poorly: only supports ~12 beacons on a conventional laptop for real-time operation with 30 Hz, which is sufficient for most practical use cases, but this limitation strips the HIAL method of the opportunity to provide more accurate estimates with the use of additional beacons. The proposed random sample consensus-based RBL method is not only tolerant against outliers, but also has much better computational performance while providing a smoother output estimate as opposed to HIAL. Also, the RBL method uses angle-based filtering of outliers in a location-independent manner instead of HIAL's distance-based filtering. This approach provides more reliable inlier/outlier classification. Finally, the refinement of the position estimate provides a locally optimal solution based on the largest consistent beacon group, that supported the best hypothesis of the RANSAC part with the least amount of error. Due to the refinement, the output of the RBL method is not only reliable, but also consistent (much less noisy than HIAL's).

An additional topic covered during my studies involved precise camera shutter speed measurement methods [SO1]-[SO4]. However, this topic was excluded from the dissertation for two main reasons. Firstly, it is not closely connected to indoor positioning methods but rather pertains to visible light communication. Secondly, it was considered unsuitable for inclusion in the dissertation due to specific formal requirements and constraints.

8.2 Future research

The localization system introduced in Section 1.4 still has room for improvements. To mention some: a main improvement would be the implementation of a beacon position correction component for the error function used in RBL method's estimate-refinement step, which would adjust the beacon's pre-measured position based on the position of evaluation, the beacon's physical properties, and its placement, since the measured direction is actually the mean direction (half-angle) of the beacon's visible portion (not its center point), which is location-dependent;

Another improvement would be an automated beacon registration process, with which the system could automatically learn the position of a new beacon based on measurements with already present references; the development of a beacon-placement aiding solution may also be beneficial for the system to minimize the number of beacons needed, and to lower the effect of geometric dilution of precision with an optimized geometric arrangement.

In [9] an interesting concept of a visible light-based indoor positioning system is introduced, that uses the TDoA principle with LED beacons modulated in the MHz range. The theoretical results in the paper are validated using simulations. It would be interesting to integrate the proposed solution of [9] into the IPS introduced in Section 1.4 to achieve a more accurate and robust system by utilizing both ADoA and TDoA principles. For example, RUPSOOK could be operated with TUPSOOK's beacon design in a way, that only the outer ring LED is blinking according to RUPSOOK, and the central circular LED can be modulated based on [9] (the high frequency modulation in the MHz range will result in an always ON state for the camera sensor). One or more photodiodes can be attached to the camera to detect the LEDs modulated in the MHz range. The resulting system would be beneficial in numerous ways: increased outlier-rejection possibilities; very precise, sub-centimeter positioning with orientation estimation from just 3 beacons; etc.

Theses

Thesis 1

I provided the detailed analysis of various error sources affecting the communication performance of the Undersampled Frequency-Shift On-Off Keying (UFSOOK) and Undersampled Phase-Shift On-Off Keying (UPSOOK) visible light communication protocols. With the achieved results the protocol's parameters can be designed to achieve optimal data transmission between LED light fixtures and digital cameras with regard to the protocol's limitations. I validated the theoretical results with simulations and real measurements.

1.1 I proposed a new equivalent mathematical operation model for digital cameras operating with modulated light sources to allow the mathematical analysis of the protocols' behavior.

1.2 I determined the bit error rate (BER) in the presence of frequency slip that is inevitable when there is no synchronization between the transmitter and the receiver. I proved that the best achievable BER depends on the frequency slip for both data symbols of the protocols.

1.3 I determined the BER as a function of threshold parameter Q, which is used by the protocol to determine the state of a transmitter. I proved that the value of Q has no effect on the BER for symbols utilizing same transmitter states (SPACE for UFSOOK, MARK for UPSOOK). I also proved that there is an optimal range for parameter Q, where the BER is minimal, for symbols using opposite transmitter states (MARK for UFSOOK, SPACE for UPSOOK).

1.4 I determined the effect of the measurement noise on the BER, for both data symbols. I proved that the effect is highest for values Q near the extrema of the sensed signal. The effect of noise is minimal for Q values at the mean signal amplitude.

1.5 I determined the frequency of occurrence and size of unsafe intervals, where the operation of the protocols may possibly be faulty. I showed that in case of UPSOOK, the unsafe intervals are aligned, as opposed to UFSOOK. I determined the packet error rate (PER) for the UPSOOK protocol based on the aforementioned achieved results.

Related publications: [S1], [S2], [S3], [S4], [S5]

Other related publications: [SO1], [SO2], [SO3], [SO4]

Thesis 2

I proposed two novel visible light communication solutions that address the issue of trackability and robust identifiability of beaconing infrastructures in visible light-based positioning systems. The proposed Trackable UPSOOK (TUPSOOK) solution, achieved through a special beacon design, makes it possible to detect the beacons in every image frame of a camera, even from large distances, while maintaining its communication capabilities and usability as practical environmental lighting infrastructure. The proposed Robust UPSOOK (RUPSOOK) protocol is able to operate in an error-free manner on a wide range of system parameters and can be used with the beacon design of TUPSOOK to also fulfill the trackability requirement. I verified the usability of the solutions with simulations and real measurements.

2.1 I proposed a novel beacon design containing an internal circular LED and an outer ring-shaped LED around it. The proposed architecture allows continuous detection and thus, continuous tracking of the transmitter. I modified the coding scheme of UPSOOK, to only utilize two transmitter states and one data frequency. The modified coding scheme (TUPSOOK) fits well the new beacon design. The price of the modifications is the somewhat decreased data rate.

2.2 I proposed a novel encoding-decoding scheme (RUPSOOK) for visible light communication based on Manchester coding and equivalent sampling, which is able to periodically transmit constant data (e.g., beacon ID). I proposed a design method to determine the nominal frequency slip to achieve optimal robustness in the protocol. I proved that the RUPSOOK protocol is robust over a finite range of system parameters, including: exposure time, frequency slip, threshold, jitter, saturation, and noise.

2.3 I proposed an adaptive thresholding method which provides robust transmitter state detection for changing signal amplitudes that may inevitably occur when the transmitter or the receiver is moving.

Related publications: [S3], [S4], [S6], [S7]

Other related publications: [SO1], [SO2], [SO3], [SO4]

Thesis 3

I proposed two novel positioning methods for angle difference of arrival (ADoA)-based positioning to determine the position and orientation of a sensor based on angle difference measurements of reference points (beacons). The proposed solutions require the sensor's normal vector (e.g., facing upwards) and the position of reference points to be known. A minimum number of 3 beacons is needed for their operation, which makes them useable for practical applications. The solutions are able to filter outlier measurements to provide robust operation. I verified the usability of the proposed solutions with simulations and real measurements.

3.1 I proposed a geometry-based heuristic method (HIAL) to determine the position and orientation of a sensor using angle difference measurements of reference points, provided that the normal vector of the sensor is known. The advantage of the proposed method with respect to the general (3D) geometric solutions is that the search is performed on a plane instead of space, thus the speed and performance is improved. The algorithm calculates potential solutions by constructing circular arcs based on beacon pairs and subsequently determining intersections based on arc pairs; each intersection point represents a potential solution. I proposed a heuristic ranking to quantify the relative confidence of these possible solutions, using two distance-based filtering parameters.

3.2 I proposed a novel, outlier-tolerant, RANSAC-based method (RBL) to determine the position and orientation of a sensor using angle difference measurements of reference points, given that the normal vector of the sensor is known. The proposed solution calculates an initial position and orientation estimate with RANSAC using a linearized equation system. Subsequently, it refines the pose estimate utilizing the best consistent beacon group of the initial estimate. The solution uses location-independent, angle-based outlier-filtering criteria, which are simple, and user-friendly to set, in contrast to HIAL's filtering parameters. The estimate-refinement step of the method provides a consistent output.

3.3 I provided a method to determine the number of necessary trials for the RANSAC-based selection of the initial beacon set based on an upper bound for outliers, and a hit rate probability design parameter. With the proposed solution the number of necessary trials can be reduced by orders of magnitude while still maintaining the desired hit rate probability of a good initial beacon set.

Related publications: [S8], [S9]

Publication of new results

[S1] M. Rátosi and G. Simon, "Performance Analysis of the UFSOOK Protocol," *2019 IEEE International Symposium on Measurements & Networking (M&N)*, Catania, Italy, pp. 1-6, 2019
DOI: 10.1109/IWMN.2019.8805018

[S2] G. Simon and M. Rátosi, "Characterization and Measurement of Performance Properties of the UFSOOK Camera Communication Protocol," in *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 10, pp. 7982-7989, 2020
DOI: 10.1109/TIM.2020.2981219
IF: 3.658

[S3] M. Rátosi and G. Simon, "Towards Robust VLC Beacon Identification in Camera Based Localization Systems," *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN),* Pisa, Italy, pp. 1-8, 2019
DOI: 10.1109/IPIN.2019.8911767

[S4] M. Rátosi and G. Simon, "Robust VLC Beacon Identification for Indoor Camera-Based Localization Systems," *Sensors*, vol. 20, no. 9, p. 2522, 2020
DOI: 10.3390/s20092522
IF: 3.576

[S5] M. Rátosi, G. Simon, "Undersampled On-Off Keying Camera Communication Methods for Beacon ID Transmission," Pannonian Conference on Advances in Information Technology (PCIT2019), Veszprém, Hungary, pp. 36-41, 2019
ISBN: 978-963-396-127-8

[S6] M. Rátosi and G. Simon, "Trackable Visible Light Beaconing and Detection for Indoor Localization Applications Using Undersampling Cameras," *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN),* Nantes, p. 4, 2018
Online:https://ipin2018.ifsttar.fr/fileadmin/contributeurs/IPIN2018/Work_In_Progress/Work_In_Progress.zip
(accessed on 15 Jan 2024, paper id: 211108)

[S7] G. Simon, G. Vakulya and M. Rátosi, "On the Utilization of Equivalent Sampling in Undersampled Asynchronous Camera Communication Protocols," 2022 IEEE International Symposium on Measurements & Networking (M&N), Padua, Italy, pp. 1-6, 2022
DOI: 10.1109/MN55117.2022.9887654

[S8] M. Rátosi and G. Simon, "Real-Time Localization and Tracking using Visible Light Communication," *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN),* Nantes, France, pp. 1-8, 2018
DOI: 10.1109/IPIN.2018.8533800

[S9] M. Rátosi and G. Simon, "Fault Tolerant Indoor Positioning Based on Azimuth Measurements," *2021 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Lloret de Mar, Spain, pp. 1-12, 2021
Online: https://ceur-ws.org/Vol-3097/paper27.pdf
(accessed on 15 Jan 2024)

Other related publications

[SO1] G. Simon, M. Rátosi and G. Vakulya, "Automatic Measurement of Digital Cameras' Exposure Time Using Equivalent Sampling," in *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1-10, Art no. 5015110, 2022
DOI: 10.1109/TIM.2022.3186372
IF: 5.332

[SO2] G. Simon, G. Vakulya, and M. Rátosi, "The Way to Modern Shutter Speed Measurement Methods: A Historical Overview," *Sensors*, vol. 22, no. 5, p. 1871, 2022
DOI: 10.3390/s22051871
IF: 3.9

[SO3] M. Rátosi, G. Vakulya and G. Simon, "Measuring Camera Exposure Time Using Equivalent Sampling," 2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Glasgow, United Kingdom, pp. 1-6, 2021
DOI: 10.1109/I2MTC50364.2021.9459789

[SO4] M. Rátosi, G. Vakulya, G. Simon, "Methods for High Precision Shutter Speed Measurements," AIS 2021 – 16th International Symposium on Applied Informatics and Related Areas, Székesfehérvár, Hungary, pp. 30-34, 2021
ISBN: 978-963-449-263-4

List of Abbreviations

ADoA	Angle difference of arrival
AoA	Angle of arrival
AGV	Automated guided vehicle
BER	Bit error rate
CSI	Channel state information
FOV	Field of view
GDOP	Geometric dilution of precision
GNSS	Global Navigation Satellite Systems
HIAL	Heuristic, inscribed angle-based localization (positioning method)
ID	Identification number
IPS	Indoor positioning system
IMU	Inertial measurement unit
LSM	Least squares method (positioning method type)
LED	Light emitting diode
LOS	Line of sight
LoRa	Long range (wide-area network technology)
MLPnP	Maximum Likelihood Perspective-n-Point (positioning method)
MEX	Method of exhaustion (positioning method)
MCU	Microcontroller unit
MCOB	Multiple chip-on-board (LED type)
NLOS	Non-line of sight
PER	Packet error rate
QoS	Quality of Service
RFID	Radio frequency identification (technology)
RANSAC	Random sample consensus
RBL	RANSAC-based localization (positioning method)
RSS	Received signal strength
RSRP	Reference signal received power
RSRP	Reference signal received quality
RUPSOOK	Robust Undersampled Phase-Shift On-Off Keying (VLC protocol)
SPB	Samples per bit
SLAM	Simultaneous localization and mapping
SNR	Signal-to-noise ratio
TDoA	Time difference of arrival
ToA	Time of arrival
ToF	Time of flight
TUPSOOK	Trackable Undersampled Phase-Shift On-Off Keying (VLC protocol)
UWB	Ultra-wideband (radio technology)
UFSOOK	Undersampled Frequency-Shift On-Off Keying (VLC protocol)
UPSOOK	Undersampled Phase-Shift On-Off Keying (VLC protocol)
VLC	Visible light communication
Wi-Fi	Wireless network technology

List of Notations and common Parameters

Visible light communication

$\lvert . \rvert$	Absolute value
S	Exposure time
$d(t)$	Detected light intensity signal as a function of time
$I(t)$	Luminous intensity of a light source as a function of time
$n(t)$	Additive noise component
n	Design parameter
f_X	Frequency of X (e.g., a symbol's or the camera's)
T_X	Time period of X (e.g., a symbol's or the camera's)
$Qf(.)$	Q-function
$\mathrm{erf}\,(.)$	Gauss error function
δ	Size of time offset between two consecutive samples caused by frequency error
δ_E	Virtual sampling period during equivalent sampling
λ	Width of unsafe interval (where the communication may possibly fail)
Ξ	Decoded symbol / symbol set
Q	Threshold parameter for LED state determination
Q_0	Ideal threshold parameter
ΔQ	Size of difference from the ideal threshold parameter Q_0
A_H	Maximum signal amplitude
A_L	Minimum signal amplitude
μ_X	Bit error rate (BER) of symbol X

Pose estimation methods

$\lVert . \rVert$	Euclidean norm
$\boldsymbol{v}$	Vector
$\boldsymbol{A}$	Matrix
$\boldsymbol{v_1} \times \boldsymbol{v_2}$	Cross product of $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$
$\boldsymbol{p_j} = (x_{p,j}, y_{p,j}, z_{p,j})$	3D vector in the real-world coordinate system (K1), with logical connection to the jth beacon
$\boldsymbol{p_j^{(2)}} = (x_{p,j}{}^{(2)}, y_{p,j}{}^{(2)}, z_{p,j}{}^{(2)})$	3D vector in the K2 coordinate system, with logical connection to the jth beacon
$\hat{C}$	Position estimate (2D)
$\hat{\hat{C}}$	Refined position estimate (2D)
$\hat{\varphi}$	Orientation estimate (rotation around z axis)
$\hat{\hat{\varphi}}$	Refined orientation estimate (rotation around z axis)
BT_j	Initial beacon triplet of the jth hypothesis
SUP_j	Set of beacons that support the jth hypothesis
NS_k	Number of beacons that support the jth hypothesis
$\mathrm{n}(.)$	Set cardinality operator (set size)

List of Figures

List of Tables

Appendix

HIAL’s evaluation

This appendix contains the HIAL and MLPnP algorithm’s error analysis results (presented in Section 6.1): mean position error, maximum position error, and the position error’s standard deviation.

Effect of camera orientation error - simulation

3D	**HIAL-3**			**HIAL-4**			**HIAL-5**			**HIAL-6**		
Angle error	**Mean**	**Max**	**SD**	**Mean**	**Max**	**SD**	**Mean**	**Max**	**SD**	**Mean**	**Max**	**SD**
1°	59.3	82.7	59.5	57.2	76.6	57.4	57.4	112.0	57.5	57.8	113.6	57.9
2°	118.5	165.4	119.0	114.5	153.9	114.7	114.8	153.5	115.1	115.4	163.1	115.7
3°	177.7	248.1	178.4	171.7	231.6	172.1	172.2	232.3	172.6	173.1	237.6	173.5
4°	236.9	331.0	237.9	228.9	310.3	229.4	229.4	310.5	229.9	230.6	317.7	231.2
5°	296.1	414.1	297.3	286.1	389.2	286.7	287.0	386.9	287.6	288.5	396.8	289.1
2D	**HIAL-3**			**HIAL-4**			**HIAL-5**			**HIAL-6**		
Angle error	**Mean**	**Max**	**SD**	**Mean**	**Max**	**SD**	**Mean**	**Max**	**SD**	**Mean**	**Max**	**SD**
1°	52.7	53.8	52.7	52.7	60.7	52.7	52.8	93.9	52.8	52.8	104.8	52.8
2°	105.4	109.9	105.4	105.4	110.7	105.4	105.5	122.5	105.5	105.5	154.6	105.5
3°	158.1	168.1	158.1	158.1	170.3	158.1	158.3	196.6	158.3	158.3	196.6	158.3
4°	210.8	228.5	210.9	210.8	226.8	210.8	210.8	254.7	210.8	210.9	250.1	210.9
5°	263.4	291.1	263.6	263.5	282.8	263.6	263.7	310.9	263.7	263.7	308.4	263.7

Table A.1 Effect of camera orientation (tilt) error to the localization error (in millimeter).

Effect of detection error - simulation

3D	**MLPnP**			**HIAL-3**			**HIAL-4**			**HIAL-6**		
Pixel error	**Mean**	**Max**	**SD**	**Mean**	**Max**	**SD**	**Mean**	**Max**	**SD**	**Mean**	**Max**	**SD**
$n = 1$	16.0	65.1	17.5	16.7	86.3	18.5	12.0	61.1	13.2	13.7	88.3	15.5
$n = 2$	32.0	121.2	35.0	33.2	185.9	36.8	24.7	289.9	27.0	27.7	532.6	31.8
$n = 3$	48.1	180.4	52.7	50.1	296.9	55.5	42.1	1388.5	60.9	43.7	1371.0	53.8
2D	**MLPnP**			**HIAL-3**			**HIAL-4**			**HIAL-6**		
Pixel error	**Mean**	**Max**	**SD**	**Mean**	**Max**	**SD**	**Mean**	**Max**	**SD**	**Mean**	**Max**	**SD**
$n = 1$	14.0	65.0	15.8	13.9	84.1	15.9	9.9	55.7	11.3	11.4	71.9	13.2
$n = 2$	28.0	105.3	31.4	27.6	184.1	31.7	20.5	287.9	23.2	22.9	244.6	26.8
$n = 3$	42.1	164.3	47.4	41.6	294.9	47.7	35.8	1376.3	56.5	35.9	1369.0	44.5

Table A.2 Effect of the detection error (in pixels) to the localization error (in millimeter).

Effect of reference position error - simulation

3D	MLPnP			HIAL-3			HIAL-4			HIAL-6		
Ref. error	Mean	Max	SD	Mean	Max	SD	Mean	Max	SD	Mean	Max	SD
$d = 1$	11.7	40.6	12.1	13.1	55.1	13.9	10.1	63.0	10.3	13.7	386.8	16.6
$d = 2$	23.3	74.8	24.1	26.2	108.2	27.8	19.8	86.2	19.9	26.5	443.9	30.3
$d = 3$	35.2	117.4	36.4	39.3	163.8	41.5	30.0	118.4	30.3	38.7	492.4	43.0
$d = 4$	46.6	147.6	48.1	52.3	221.8	55.3	40.3	155.8	40.7	52.2	641.5	58.2
2D	MLPnP			HIAL-3			HIAL-4			HIAL-6		
Ref. error	Mean	Max	SD	Mean	Max	SD	Mean	Max	SD	Mean	Max	SD
$d = 1$	10.5	34.7	11.0	11.7	54.5	12.5	9.2	62.7	9.5	11.7	113.8	13.7
$d = 2$	20.8	70.1	21.8	23.5	106.9	25.1	18.0	83.7	18.3	22.7	134.0	24.9
$d = 3$	31.5	105.4	33.0	35.0	162.4	37.4	27.2	117.4	27.8	33.3	273.6	36.0
$d = 4$	41.8	136.2	43.6	46.8	219.1	49.8	36.8	155.3	37.6	44.3	354.3	47.5

Table A.3 Effect of the reference position error to the localization error (in millimeter).

Effect of outliers - simulation

	MLPnP			HIAL-4			HIAL-5			HIAL-6		
	Mean	Max	SD	Mean	Max	SD	Mean	Max	SD	Mean	Max	SD
3D	702.0	2113.2	718.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	1.2	0.0
2D	624.8	1523.3	649.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	1.2	0.0

* 0,0 error means error smaller than 0.05mm

Table A.4 Effect of outliers on the localization error (in millimeter).

Static accuracy – measurement

	MLPnP			HIAL-3			HIAL-4			HIAL-5			HIAL-6		
	Mean	Max	SD	Mean	Max	SD	Mean	Max	SD	Mean	Max	SD	Mean	Max	SD
3D	33.9	81.1	2.2	38.7	84.4	3.2	31.3	78.0	5.0	28.1	82.7	4.4	28.9	83.0	5.4
2D	33.2	75.7	2.1	36.6	88.7	3.6	27.6	74.5	5.0	25.6	80.9	4.0	26.3	76.2	5.1

Table A.5 Static localization error in real measurements (in millimeter).

Dynamic accuracy - measurement

3D	Mean	Max	SD
MLPnP	18.6	61.1	15.7
HIAL-3	19.0	51.1	13.1
HIAL-4	19.9	71.1	17.5
HIAL-5	12.5	70.2	16.5
HIAL-6	13.6	58.4	16.0

Table A.6 Dynamic localization error in real measurements (in millimeter).

3D	**MLPnP**			**HIAL-3**			**HIAL-4**			**HIAL-5**			**HIAL-6**		
Movement speed	**Mean**	**Max**	**SD**	**Mean**	**Max**	**SD**	**Mean**	**Max**	**SD**	**Mean**	**Max**	**SD**	**Mean**	**Max**	**SD**
0.05 m/s	12.1	19.2	1.9	8.1	15.0	2.9	7.7	14.2	2.2	11.5	42.8	2.6	11.5	42.8	2.6
0.1 m/s	11.9	19.1	2.1	8.1	15.8	3.1	7.6	14.4	2.4	11.2	27.8	2.5	11.2	27.8	2.5
0.3 m/s	11.7	19.6	2.6	7.7	16.5	4.1	7.2	15.6	3.2	11.3	19.9	2.3	11.3	19.9	2.3
0.5 m/s	11.4	20.2	2.3	7.0	15.7	3.7	7.0	14.5	2.7	11.2	24.5	2.3	11.2	24.5	2.3
1 m/s	10.9	22.6	3.1	6.4	19.1	4.4	6.5	18.4	3.4	11.2	27.6	2.2	11.2	27.6	2.2

Table A.7 3D localization error as a function of object speed, in a linear trajectory (in millimeter).

References

[1] L. Banin, U. Schtzberg, Y. Amizur, "Next Generation Indoor Positioning System Based on WiFi Time of Flight," *26th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2013)*, Nashville, TN, pp. 975-982., 2013
Online:https://www.researchgate.net/publication/273443111_Next_Generation_Indoor_Positioning_System_Based_on_WiFi_Time_of_Flight
(accessed on 2 January 2024)

[2] D. Giovanelli, E. Farella, D. Fontanelli and D. Macii, "Bluetooth-Based Indoor Positioning Through ToF and RSSI Data Fusion," *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Nantes, France, pp. 1-8, 2018
DOI: 10.1109/IPIN.2018.8533853

[3] S. Holm, "Ultrasound positioning based on time-of-flight and signal strength," *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Sydney, NSW, Australia, pp. 1-6, 2012
DOI: 10.1109/IPIN.2012.6418728

[4] Z. Liu, L. Chen, X. Zhou, Z. Jiao, G. Guo and R. Chen, "Machine Learning for Time-of-Arrival Estimation With 5G Signals in Indoor Positioning," *IEEE Internet of Things Journal*, vol. 10, no. 11, pp. 9782-9795, 2023
DOI: 10.1109/JIOT.2023.3234123

[5] L. Banin, O. Bar-Shalom, N. Dvorecki, Y. Amizur, "High-Accuracy Indoor Geolocation Using Collaborative Time of Arrival (CToA)," *Intel White Paper* 2017
Online: https://www.researchgate.net/publication/320146822_High-Accuracy_Indoor_Geolocation_using_Collaborative_Time_of_Arrival_CToA (accessed on 2 January 2024)

[6] J. J. Perez-Solano, S. Ezpeleta, and J. Claver, "Indoor localization using Time Difference of Arrival with UWB signals and unsynchronized devices," *Ad Hoc Networks*, vol. 99, p. 102067, 2019
DOI: 10.1016/j.adhoc.2019.102067

[7] C. Wu, H. Hou, W. Wang, Q. Huang and X. Gao, "TDOA Based Indoor Positioning with NLOS Identification by Machine Learning," *10th International Conference on Wireless Communications and Signal Processing (WCSP)*, Hangzhou, China, pp. 1-6, 2018
DOI: 10.1109/WCSP.2018.8555654

[8] A. Makki, A. Siddig, M. Saad, J. R. Cavallaro and C. J. Bleakley, "Indoor Localization Using 802.11 Time Differences of Arrival," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 3, pp. 614-623, 2016
DOI: 10.1109/TIM.2015.2506239

[9] S. -Y. Jung, S. Hann and C. -S. Park, "TDOA-based optical wireless indoor localization using LED ceiling lamps," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 4, pp. 1592-1597, 2011
DOI: 10.1109/TCE.2011.6131130

[10] T.-H. Do and M. Yoo, "TDOA-based indoor positioning using visible light," *Photonic Network Communications*, vol. 27, no. 2, pp. 80–88, 2014
DOI: 10.1007/s11107-014-0428-4

[11] M. H. Bergen, F. S. Schaal, R. Klukas, J. Cheng, and J. F. Holzman, "Toward the implementation of a universal angle-based optical indoor positioning system," *Frontiers of Optoelectronics*, vol. 11, no. 2, pp. 116–127, 2018
DOI: 10.1007/s12200-018-0806-0

[12] M. Zhao, T. Chang, A. Arun, R. Ayyalasomayajula, C. Zhang, and D. Bharadia, "ULoc: Low-Power, Scalable and Cm-Accurate UWB-Tag Localization and Tracking for Indoor Applications," *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 3, Sep. 2021
DOI: 10.1145/3478124

[13] C. Li, J. Zhen, K. Chang, A. Xu, H. Zhu, and J. Wu, "An Indoor Positioning and Tracking Algorithm Based on Angle-of-Arrival Using a Dual-Channel Array Antenna," *Remote Sensing*, vol. 13, no. 21, 2021
DOI: 10.3390/rs13214301

[14] A. Arafa, S. Dalmiya, R. Klukas, and J. F. Holzman, "Angle-of-arrival reception for optical wireless location technology," *Optics Express*, vol. 23, no. 6, pp. 7755–7766, 2015
DOI: 10.1364/OE.23.007755

[15] S. Wielandt and L. D. Strycker, "Indoor Multipath Assisted Angle of Arrival Localization," *Sensors*, vol. 17, no. 11, 2017
DOI: 10.3390/s17112522

[16] M. Girolami, F. Furfari, P. Barsocchi and F. Mavilia, "A Bluetooth 5.1 Dataset Based on Angle of Arrival and RSS for Indoor Localization," *IEEE Access*, vol. 11, pp. 81763-81776, 2023
DOI: 10.1109/ACCESS.2023.3301126

[17] B. Zhu, J. Cheng, Y. Wang, J. Yan and J. Wang, "Three-Dimensional VLC Positioning Based on Angle Difference of Arrival With Arbitrary Tilting Angle of Receiver," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 1, pp. 8-22, Jan. 2018
DOI: 10.1109/JSAC.2017.2774435

[18] M. Heydariaan, H. Dabirian and O. Gnawali, "AnguLoc: Concurrent Angle of Arrival Estimation for Indoor Localization with UWB Radios," *16th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, Marina del Rey, CA, USA, pp. 112-119, 2020
DOI: 10.1109/DCOSS49796.2020.00028

[19] Chunhan Lee et al., "Indoor positioning system based on incident angles of infrared emitters," *30th Annual Conference of IEEE Industrial Electronics Society, 2004. IECON 2004*, Busan, Korea (South), pp. 2218-2222 Vol. 3, 2004
DOI: 10.1109/IECON.2004.1432143

[20] A. Olivier, G. Bielsa, I. Tejado, M. Zorzi, J. Widmer and P. Casari, "Lightweight Indoor Localization for 60-GHz Millimeter Wave Systems," *13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, London, UK, pp. 1-9, 2016
DOI: 10.1109/SAHCN.2016.7732999

[21] J. Geng, L. Xia, J. Xia, Q. Li, H. Zhu, and Y. Cai, "Smartphone-Based Pedestrian Dead Reckoning for 3D Indoor Positioning," *Sensors*, vol. 21, no. 24, 2021
DOI: 10.3390/s21248180

[22] B. Khalili, R. Ali Abbaspour, A. Chehreghan, and N. Vesali, "A Context-Aware Smartphone-Based 3D Indoor Positioning Using Pedestrian Dead Reckoning," *Sensors*, vol. 22, no. 24, 2022
DOI: 10.3390/s22249968

[23] S. Jeong, J. Min and Y. Park, "Indoor Positioning Using Deep-Learning-Based Pedestrian Dead Reckoning and Optical Camera Communication," *IEEE Access*, vol. 9, pp. 133725-133734, 2021
DOI: 10.1109/ACCESS.2021.3115808

[24] B. Hussain, Y. Wang, R. Chen, H. C. Cheng and C. P. Yue, "LiDR: Visible-Light-Communication-Assisted Dead Reckoning for Accurate Indoor Localization," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 15742-15755, 2022
DOI: 10.1109/JIOT.2022.3151664.

[25] Y. Wang, Z. Li, J. Gao and Z. Long, "Deep neural network-based wi-fi/pedestrian dead reckoning indoor positioning system using adaptive robust factor graph model," *IET Radar, Sonar & Navigation*, vol. 14, no. 1, p. 36-47, 2020
DOI: 10.1049/iet-rsn.2019.0260

[26] F. Qin, T. Zuo, and X. Wang, "CCpos: WiFi Fingerprint Indoor Positioning System Based on CDAE-CNN," *Sensors*, vol. 21, no. 4, 2021
DOI: 10.3390/s21041114

[27] M. Nabati and S. A. Ghorashi, "A real-time fingerprint-based indoor positioning using deep learning and preceding states," *Expert Systems with Applications*, vol. 213, p. 118889, 2023
DOI: 10.1016/j.eswa.2022.118889

[28] J. Wang and J. Park, "An Enhanced Indoor Positioning Algorithm Based on Fingerprint Using Fine-Grained CSI and RSSI Measurements of IEEE 802.11n WLAN," *Sensors*, vol. 21, no. 8, 2021
DOI: 10.3390/s21082769

[29] B. Ezhumalai, M. Song, and K. Park, "An Efficient Indoor Positioning Method Based on Wi-Fi RSS Fingerprint and Classification Algorithm," *Sensors*, vol. 21, no. 10, 2021
DOI: 10.3390/s21103418

[30] Y. Wu, R. Chen, W. Li, Y. Yu, H. Zhou and K. Yan, "Indoor Positioning Based on Walking-Surveyed Wi-Fi Fingerprint and Corner Reference Trajectory-Geomagnetic Database," *IEEE Sensors Journal*, vol. 21, no. 17, pp. 18964-18977, 2021
DOI: 10.1109/JSEN.2021.3086485

[31] T. Kim Geok *et al.*, "Review of Indoor Positioning: Radio Wave Technology," *Applied Sciences*, vol. 11, no. 1, article no. 279, 2020
DOI: 10.3390/app11010279

[32] M. Cominelli, P. Patras, and F. Gringoli, "Dead on Arrival: An Empirical Study of The Bluetooth 5.1 Positioning System," *13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, Los Cabos, Mexico, pp. 13–20, 2019
DOI: 10.1145/3349623.3355475

[33] L. Zwirello, T. Schipper, M. Jalilvand and T. Zwick, "Realization Limits of Impulse-Based Localization System *for* Large-Scale Indoor Applications," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 1, pp. 39-51, 2015
DOI: 10.1109/TIM.2014.2332241

[34] Yao Zhao, Liang Dong, Jiang Wang, Bo Hu and Yuzhuo Fu, "Implementing indoor positioning system via ZigBee devices," *42nd Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, pp. 1867-1871, 2008
DOI: 10.1109/ACSSC.2008.5074752

[35] H. Chen *et al.*, "Research on indoor positioning method based on LoRa-improved fingerprint localization algorithm," *Scientific Reports*, vol. 13, no. 1, p. 13981, 2023
DOI: 10.1038/s41598-023-41250-x

[36] C. L. F. Mayorga *et al.*, "Cooperative Positioning Techniques for Mobile Localization in 4G Cellular Networks," *IEEE International Conference on Pervasive Services*, Istanbul, Turkey, pp. 39-44, 2007
DOI: 10.1109/PERSER.2007.4283887

[37] B. Camajori Tedeschini *et al.*, "A feasibility study of 5G positioning with current cellular network deployment," *Scientific Reports*, vol. 13, no. 1, p. 15281, Sep. 2023
DOI: 10.1038/s41598-023-42426-1

[38] A. Diallo, Z. Lu, and X. Zhao, "Wireless Indoor Localization Using Passive RFID Tags," *Procedia Computer Science*, vol. 155, pp. 210–217, 2019
DOI: 10.1016/j.procs.2019.08.031.

[39] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, "Indoor Localization without the Pain," *Sixteenth Annual International Conference on Mobile Computing and Networking*, Chicago, Illinois, USA, pp. 173–184, 2010
DOI: 10.1145/1859995.1860016

[40] K. Long, D. F. Nsalo Kong, K. Zhang, C. Tian, and C. Shen, "A CSI-Based Indoor Positioning System Using Single UWB Ranging Correction," *Sensors*, vol. 21, no. 19, p. 6447, 2021
DOI: 10.3390/s21196447

[41] G. Zachár and G. Simon, "Radio interferometric tracking using redundant phase measurements," *IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, Pisa, Italy, 2015, pp. 2003-2008
DOI: 10.1109/I2MTC.2015.7151590

[42] F. G. Serrenho, J. A. Apolinário, A. L. L. Ramos, and R. P. Fernandes, "Gunshot Airborne Surveillance with Rotary Wing UAV-Embedded Microphone Array," *Sensors*, vol. 19, no. 19, p. 4271, 2019,
DOI: 10.3390/s19194271.

[43] M. Cobos, et al., "Simultaneous ranging and self-positioning in unsynchronized wireless acoustic sensor networks," *IEEE Transactions on Signal Processing*, vol. 64, no. 22, pp. 5993–6004, 2016
DOI: 10.1109/TSP.2016.2603972

[44] J. R. Gonzalez and C. J. Bleakley, "High-Precision Robust Broadband Ultrasonic Location and Orientation Estimation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 5, pp. 832-844, Oct. 2009
DOI: 10.1109/JSTSP.2009.2027795

[45] A. Colombo, D. Fontanelli, D. Macii and L. Palopoli, "Flexible Indoor Localization and Tracking Based on a Wearable Platform and Sensor Data Fusion," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 4, pp. 864-876, 2014,
DOI: 10.1109/TIM.2013.2283546

[46] C. Lu, H. Uchiyama, D. Thomas, A. Shimada, and R. Taniguchi, "Indoor Positioning System Based on Chest-Mounted IMU," *Sensors*, vol. 19, no. 2, p. 420, Jan. 2019
DOI: 10.3390/s19020420

[47] R. Zhang, W. Zhong, Q. Kemao and S. Zhang, "A Single LED Positioning System Based on Circle Projection," *IEEE Photonics Journal*, vol. 9, no. 4, pp. 1-9, Art no. 7905209., 2017
DOI: 10.1109/JPHOT.2017.2722474

[48] Liang-Fu Li, Zu-Ren Feng and Qin-Ke Peng, "Detection and model analysis of circular feature for robot vision," *International Conference on Machine Learning and Cybernetics*, Shanghai, China, pp. 3943-3948 vol.6, 2004
DOI: 10.1109/ICMLC.2004.1380541

[49] G. Zachár, G. Vakulya and G. Simon, "Design of a VLC-based beaconing infrastructure for indoor localization applications," *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, Turin, Italy, 2017, pp. 1-6
DOI: 10.1109/I2MTC.2017.7969837

[50] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao, "Epsilon: a visible light based positioning system," *11th USENIX Conference on Networked Systems Design and Implementation (NSDI'14)*, pp. 331-343, Seattle, WA, USA, 2014
ISBN: 978-1-931971-09-6

[51] A. J. Davison, I. D. Reid, N. D. Molton and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052-1067, June 2007
DOI: 10.1109/TPAMI.2007.1049

[52] C. Sánchez, et al., "Localization and tracking in known large environments using portable real-time 3D sensors," *Computer Vision and Image Understanding*, vol. 149, pp. 197-208, 2016
DOI: 10.1016/j.cviu.2015.11.012

[53] S.-H. Yang, E.-M. Jeong, D.-R. Kim, H.-S. Kim, Y.-H. Son, and S.-K. Han, "Indoor three-dimensional location estimation based on LED visible light communication," *Electronics Letters*, vol. 49, no. 1, pp. 54–56, 2013
DOI: 10.1049/el.2012.3167

[54] M. Aminikashani,W. Gu, and M. Kavehrad, "Indoor location estimation with optical based orthogonal frequency division multiplexing communications," *Optical Engineering*, vol. 55, pp. 1572–1579, 2016
DOI: 10.1117/1.OE.55.5.056116

[55] 2. B. Lin, X. Tang, Z. Ghassemlooy, C. Lin and Y. Li, "Experimental Demonstration of an Indoor VLC Positioning System Based on OFDMA," *IEEE Photonics Journal*, vol. 9, no. 2, pp. 1-9, Art no. 7902209., 2017
DOI: 10.1109/JPHOT.2017.2672038

[56] P. Du et al., "Experimental Demonstration of 3D Visible Light Positioning Using Received Signal Strength With Low-Complexity Trilateration Assisted by Deep Learning Technique," *IEEE Access*, vol. 7, pp. 93986-93997, 2019
DOI: 10.1109/ACCESS.2019.2928014

[57] Huynh, P.; Yoo, M., "VLC-Based Positioning System for an Indoor Environment Using an Image Sensor and an Accelerometer Sensor," *Sensors*, vol. 16, no. 6, Article Number 783, 2016

[58] G. Simon, G. Zachár and G. Vakulya, "Lookup: Robust and Accurate Indoor Localization Using Visible Light Communication," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 9, pp. 2337-2348, Sept. 2017
DOI: 10.1109/TIM.2017.2707878

[59] S. Cincotta, C. He, A. Neild, and J. Armstrong, "High angular resolution visible light positioning using a quadrant photodiode angular diversity aperture receiver (QADA)," *Opt. Express*, vol. 26, no. 7, pp. 9230–9242, Apr. 2018
DOI: 10.1364/OE.26.009230

[60] G. Zachár, G. Vakulya, and G. Simon, "Bearing Estimation for Indoor Localization Systems Using Planar Circular Photodiode Arrays," *Applied Sciences*, vol. 10, no. 11, p. 3683, May 2020
DOI: 10.3390/app10113683

[61] R. D. Roberts, "Undersampled frequency shift ON-OFF keying (UFSOOK) for camera communications (CamCom)," *22nd Wireless and Optical Communication Conference*, Chongqing, China, 2013, pp. 645-648
DOI: 10.1109/WOCC.2013.6676454

[62] P. Luo, Z. Ghassemlooy, H. Le Minh, X. Tang and H. -M. Tsai, "Undersampled phase shift ON-OFF keying for camera communication," *Sixth International Conference on Wireless Communications and Signal Processing (WCSP)*, Hefei, China, 2014, pp. 1-6
DOI: 10.1109/WCSP.2014.6992043

[63] S. M. Berman, D. S. Greenhouse, I. L. Bailey, R. D. Clear, and T. W. Raasch, "Human Electroretinogram Responses to Video Displays, Fluorescent Lighting, and Other High Frequency Sources," *Optometry and Vision Science*, vol. 68, no. 8, 1991
Online: https://eta-publications.lbl.gov/sites/default/files/24666.pdf
(accessed on 8 Jan 2024)

[64] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta, "Luxapose: Indoor positioning with mobile phones and visible light," *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, 2014
DOI: 10.1145/2639108.2639109

[65] S. Urban, J. Leitloff, and S. Hinz, "Improved wide-angle, fisheye and omnidirectional camera calibration, " *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 108, pp. 72–79, 2015
DOI: 10.1016/j.isprsjprs.2015.06.005

[66] Improved OcamCalib MatLab toolbox.
Online:https://github.com/urbste/ImprovedOcamCalib
(downloaded: 2017.10.27)

[67] F. Remondino and C. S. Fraser, "Digital camera calibration methods: Considerations and comparisons," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, pp. 266–272, 2006
DOI: 10.3929/ethz-b-000158067

[68] Open-source Computer Vision (OpenCV) library. Online:
https://opencv.org/ (version 3.3.1, downloaded: 2017.12.22)

[69] Multitarget-tracker library. Online:
https://github.com/Smorodov/Multitarget-tracker/ (downloaded: 2018.03.15)

[70] C. Liu, T. Liu, C. Xing and J. Zhao, "LED Visible Light Communication Indoor Positioning Method Based on Mobile Phone at Any Horizontal Orientation," 2018 37th Chinese Control Conference (CCC), Wuhan, China, pp. 7246-7251, 2018
DOI: 10.23919/ChiCC.2018.8482891

[71] Y. Li, Z. Ghassemlooy, X. Tang, B. Lin and Y. Zhang, "A VLC Smartphone Camera Based Indoor Positioning System," *IEEE Photonics Technology Letters*, vol. 30, no. 13, pp. 1171-1174, 2018
DOI: 10.1109/LPT.2018.2834930

[72] X. Cui, K. Yu, S. Zhang and H. Wang, "Azimuth-Only Estimation for TDOA-Based Direction Finding With 3-D Acoustic Array," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 4, pp. 985-994, 2020
DOI: 10.1109/TIM.2019.2908694.

[73] V. Pierlot and M. Van Droogenbroeck, "A New Three Object Triangulation Algorithm for Mobile Robot Positioning," *IEEE Transactions on Robotics*, vol. 30, no. 3, pp. 566-577, 2014
DOI: 10.1109/TRO.2013.2294061

[74] P. Hu, P. H. Pathak, X. Feng, H. Fu, and P. Mohapatra, "ColorBars: Increasing Data Rate of LED-to-Camera Communication Using Color Shift Keying," *11th ACM Conference on Emerging Networking Experiments and Technologies*, Heidelberg, Germany, 2015
DOI: 10.1145/2716281.2836097

[75] P. M. Hubel, J. Liu, and R. J. Guttosch, "Spatial frequency response of color image sensors: Bayer color filters and Foveon X3," *SPIE 5301, Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications V*, vol. 5301, pp. 402-407, 2004
DOI: 10.1117/12.561568

[76] S. Urban, J. Leitloff, and S. Hinz, "MLPNP – A Real-Time Maximum Likelihood Solution To The Perspective-N-Point Problem," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. III–3, pp. 131–138, June 2016
DOI: 10.5194/isprsannals-III-3-131-2016

[77] R. D. Roberts, "A MIMO protocol for camera communications (CamCom) using undersampled frequency shift ON-OFF keying (UFSOOK)," *2013 IEEE Globecom Workshops (GC Wkshps),* Atlanta, GA, USA, 2013, pp. 1052-1057
DOI: 10.1109/GLOCOMW.2013.6825131

[78] P. Luo, Z. Ghassemlooy, H. L. Minh, H. -M. Tsai and X. Tang, "Undersampled-PAM with subcarrier modulation for camera communications," *2015 Opto-Electronics and Communications Conference (OECC)*, Shanghai, China, pp. 1-3, 2015
DOI: 10.1109/OECC.2015.7340212

[79] P. Luo et al., "Experimental Demonstration of RGB LED-Based Optical Camera Communications," *IEEE Photonics Journal*, vol. 7, Art no. 7904212, no. 5, pp. 1-12, 2015
DOI: 10.1109/JPHOT.2015.2486680

[80] D. Scaramuzza, A. Martinelli and R. Siegwart, "A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion," *Fourth IEEE International Conference on Computer Vision Systems (ICVS'06),* New York, NY, USA, 2006, p. 45
DOI: 10.1109/ICVS.2006.3

[81] Y. Nakazawa, H. Makino, K. Nishimori, D. Wakatsuki and H. Komagata, "LED-tracking and ID-estimation for indoor positioning using visible light communication," *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN),* Busan, 2014, pp. 87-94
DOI: 10.1109/IPIN.2014.7275471

[82] R. B. Langley, "Dilution of Precision," *GPS World*, Vol. 10, No. 5, 1999, pp. 52-59.

[83] G. Vakulya and G. Simon, "Fast Adaptive Acoustic Localization for Sensor Networks," in *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 5, pp. 1820-1829, May 2011
DOI: 10.1109/TIM.2011.2108074

[84] H. Zhang and Z. Zhang, "AOA-Based Three-Dimensional Positioning and Tracking Using the Factor Graph Technique," *Symmetry*, vol. 12, no. 9, p. 1400, 2020
DOI: 10.3390/sym12091400

[85] Z. Wang, et al., "A Set-membership Approach for Visible Light Positioning with Fluctuated RSS Measurements," *10th International Conference on Indoor Positioning and Indoor Navigation* - Work-in-Progress Papers, Pisa, Italy, pp. 275-282, 2019
Online:https://ceur-ws.org/Vol-2498/short36.pdf
(downloaded: 2023.11.10.)

[86] P. Li and X. Ma, "Robust Acoustic Source Localization with TDOA Based RANSAC Algorithm," *Emerging Intelligent Computing Technology and Applications*, pp. 222–227, 2009
DOI: 10.1007/978-3-642-04070-2_25

[87] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Readings in Computer Vision*, pp. 726–740, 1987
DOI: 10.1016/B978-0-08-051581-6.50070-2