Optical Camera Communication for Active Marker Identification in Camera-based Positioning Systems

Lorenz Gorse*○, Christoffer Löffler*, Christopher Mutschler*, and Michael Philippsen○
* Machine Learning and Information Fusion Group, Fraunhofer Institute for Integrated Circuits IIS, Germany
○ Programming Systems Group, Friedrich-Alexander University Erlangen-Nürnberg (FAU), Germany

Email: [christoffer.loeffer | christopher.mutschler]@iis.fraunhofer.de
Email: [lorenz.gorse | michael.philippsen]@fau.de

Abstract—Outside-in camera-based localization systems determine the position of mobile objects (with markers attached to them) by observing a tracking area with multiple camera anchors. Up to now the identification of the objects in the camera images only works for close objects (with passive 3-dimensional marker constellations) or when tracking gaps or a slow identification are acceptable (with an active LED that blinks a unique identification code).

We present a novel Optical Camera Communication method whose LEDs are never switched off completely to allow continuous tracking. We encode bits with variations of light intensity and show how an outside-in optical localization system can reliably identify the markers within a few frames.

The identification works with a bit error ratio of $7.5 \cdot 10^{-5}$. It works for moving markers (with speeds of at least up to $4 \frac{m}{s}$), for distant markers (with distances of at least up to $44 m$), for multiple markers, and under moderate ambient infrared light.

I. INTRODUCTION

Local positioning systems are used to track mobile objects in warehouse management [1], sports [2], virtual reality [3], [4], etc. Outside-in optical localization systems that use static cameras to capture the tracking area are cheap and accurate, but often they are also unreliable, especially when multiple objects are present, when objects are far away, or under ambient light.

Attaching unique geometrically aligned retro-reflective (passive) markers to the tracked objects [5]–[9] does not work well for distant objects as either the central light source needs to be very bright to clearly identify the reflection in the camera images or the markers need to be extensive which restricts object mobility. Static active markers that permanently emit light fail if multiple distant objects are close together as their points in the camera image are inseparable. Active markers that blink an identification code [10]–[17] help, but they cannot be tracked during the off-phase of the blinking LED.

This paper also uses an active marker. But instead of switching the LED off for transmitting a binary 0, we set the LED to a low intensity, thus making continuous tracking possible. The main technical contribution of the paper is how to reliably detect the two LED states. Similar to other Optical Camera Communication systems [10]–[18] we also do not explicitly synchronize transmitters and receivers, but instead exploit that receivers can correctly synchronize on frequent signal edges and by using a sampling frequency that is larger than the transmission frequency.

The paper is organized as follows. After a discussion of related work in Section II, Section III sketches the system architecture from the markers perspective and focuses on the encoding of the identification codes. The next section describes in detail how anchors decode received light intensities back to binary data. Section V evaluates the transmission reliability for various common scenarios and shows that our identification scheme works for moving markers, across long distances, for multiple markers, and in both dark and bright environments.

II. RELATED WORK

Above we have already summarized the problems of unique 3-dimensional geometric constellations of passive markers made from retro-reflective material [5]–[9]. Our active LED markers operate across significantly larger distances and avoid those occlusion problems or mobility restrictions. While passive markers can be identified in a single camera frame, current active blinking markers [10]–[17] not only are invisible when the LED is off but they also add a latency to the identification as several frames are needed to decode the transmitted bit sequence. In contrast, we never switch off our LEDs completely and thus never interrupt tracking.

Sakata et al. [16] buy shorter dark phases with an even longer identification latency. They let exactly one marker blink at a time while all other markers are on. A camera then identifies the blinking marker and can still track the others. With many markers it may take a while before it is an object’s turn to blink. In contrast, all our LEDs are always on.

If there is no central unit that picks the marker that blinks, all decentralized markers blink their identification codes continuously and repetitively. Then there is the problem of detecting the start of a bit code. Aiténbichler et al. [14] use a unique separator character between two adjacent transmissions of the code. Like us, they can also identify multiple markers simultaneously, but they again switch off the LEDs temporarily. Instead of separator characters, we use codes that are unique within $k$ bits, even in repetitive transmissions.

Kuo et al. [19] exploit the rolling shutter used by most CMOS sensors to decode a binary beacon transmitted by an LED in only one frame. This works well for short ranges, but is unsuitable for distant objects.

While we use a stable blinking frequency, other popular OCC schemes (Undersampled Frequency Shift On-Off Keying...
Tracking area

Marker A
Transmitter

Anchor
Receiver

Marker B
Transmitter

Fig. 1: One compute node (CN) observes the markers in the tracking area. More CNs are needed for tracking/triangulation.

(UFSOOK) [11], Undersampled Phase Shift On-Off Keying (UPSOOK) [15], and Lookup [17]) use multiple frequencies or work with phase shifts. They all suffer from dark phases that prevent continuous tracking. In UFSOOK the LED blinks at a high frequency to encode a separator. Two other frequencies that need to be aligned with the camera’s frame rate encode bits. While \( f_{\text{same}} \) (for a binary 1) ensures that two adjacent camera frames see the same intensity, \( f_{\text{diff}} \) ensures that two adjacent frames differ. UPSOOK uses only a single frequency that is chosen as an integer multiple of the camera’s frequency. If two succeeding bits are equal (either 00 or 11) the LED simply blinks at a frequency \( f \), while if the bits toggle, the signal is shifted by 180°, i.e., the LED stays off for a short while. To solve ambiguity problems introduced by the phase shift UPSOOK uses a header, i.e., a mark symbol. Lookup encodes a binary 1 by letting the LED blink at a frequency \( f \) with a 50% duty cycle at full intensity. It encodes a binary 0 with a constant signal at 50% intensity. Similar to us, Lookup uses bit sequences that do not require separators.

III. TRANSMITTER SIDE

The camera anchors of our positioning system track markers from outside the tracking area. Every anchor consists of a compute node (CN) and a camera. Although triangulating markers needs at least two anchors, Fig. 1 shows only one because this suffices to describe the identification of markers and the Optical Camera Communication.

Our marker is a single infrared LED with a microcontroller that can set the LED’s intensity. We use a 20% duty cycle, i.e., intensity, (called \( f_0 \)) and a 100% duty cycle (\( f_1 \)). Each marker continually transmits the bit representation of its unique identification code by using \( f_0 \) for transmitting a binary 0 and \( f_1 \) for a binary 1. Anchors receive the identification code from a visible marker by observing the LED’s intensity over several consecutive frames. The transmission frequency \( f_i \) is half the camera frame rate \( f_s \), so that we sample every bit with at least two frames. The transmitted data must not contain more than two identical bits in a row, because with more signal edges it is easier to synchronize the receivers.

Markers repeat their identification codes endlessly. To avoid the latency caused by a special separating pattern, the \( k \) bit long) identification code of one marker may not appear anywhere within the cyclic repetition of another marker’s code. Below we thus describe how to construct such a set \( S \) of non-interchangeable identification codes.

1) Let \( B_k \) be the set of all \( 2^k \) binary strings of length \( k \).
2) Remove all elements from \( B_k \) that, when transmitted repeatedly, would produce a binary sequence with more than two identical bits in a row (sparse signal edges).
3) A string \( b \) is called a shift of a string \( a \) iff \( a \) can be turned into \( b \) by repeatedly moving the last character to the front. Example: The shifts of the string \( abcde \) are \( abcd\), \( eabcd\), \( deabc\), \( cdeab\), and \( bcdea\).

Partition \( B_k \) such that two \( b_1, b_2 \in B_k \) are in the same subset of the partition iff \( b_1 \) is a shift of \( b_2 \).
4) Let \( S \) be a set that contains exactly one representative from each subset of the partition of \( B_k \).

Any of the resulting identification codes in \( S \) can be identified after just \( k \) bits are received. For example, for \( k = 8 \), a resulting set is \( S = \{ 00100101, 00101101, 00110110, 00101101, 00110011, 01011010, 01011011, 01010101 \} \). After receiving the bits 01100110, we can detect the originally transmitted code 00110011, because no other identification code could have caused this bit sequence.

IV. RECEIVER SIDE

Markers appear as bright spots in otherwise mostly dark camera images. A CN uses an established detection and tracking algorithm [20] to detect markers in every frame and track them across frames. With the knowledge where a marker is, its intensity needs to be measured. To do this, we consider the bounding rectangle around the contour of a detected marker and extend it by a few pixels in every direction to include all the light from the marker within the rectangle (we found 5 pixels to work well in practice). We then measure the intensity of a marker by summing the grayscale values of all pixels within the bounding rectangle.

A CN tracks the intensity of every visible marker separately over multiple frames. For example, if an anchor has seen two markers \( A \) and \( B \) in the last 30 frames, it produces one list of intensities for marker \( A \) and another list of intensities for marker \( B \), each with 30 elements (one intensity value for every frame). We call such a list of intensity values a marker trail. Formally, a marker trail with \( N \) elements is a function

\[
A : \{0, \ldots, N-1\} \mapsto \mathbb{N}^+_0.
\]

where \( A(n) \) is the intensity of the marker as measured in frame \( n \). A CN then processes the marker trails individually to determine the transmitted binary data.

Fig. 2 shows the processing steps that we discuss below.

![Fig. 2: Steps of the receiver side processing. A: Intensity Normalization, B: Signal Reconstruction, C: Data Derivation.](image-url)
A. Intensity Normalization

Depending on how far away a marker is from the anchor and how much ambient light there is, the measured intensity value $A(n)$ of a marker varies. Fig. 3(a) illustrates this effect for a moving marker that transmits a random bit sequence with frequent signal edges. A received intensity $A(n)$ for a sent $I_0$ can even be higher than the received value for an $I_1$ when the object is closer to the anchor. To deal with this problem, we first normalize the measured intensity values to $[0; 1]$.

Fig. 3(a) and the zoomed-in version Fig. 3(b) show that the measured intensity values vary between a minimum and a maximum within the context of a few frames. Moreover, it is obvious that there are strong bursts at the minimum and at the maximum. This is intentional and caused by both the fact that the camera frame rate is twice the transmission frequency and that there is a bit change after at most two bits.

First, while some frames may capture a combination of two bits, the transmission duration of two frames per bit guarantees that there is at least one frame for every bit that captures this bit exclusively. For instance, at least one frame must completely observe the marker when it is set to the intensity $I_0$ for the transmission of a binary 0. Similarly, if the marker transmits a binary 1, at least one frame must completely observe the marker while it is set to the intensity $I_1$. These frames must measure a minimal (when observing $I_0$) or maximal (when observing $I_1$) intensity value.

Second, because transmitted data may not contain more than two identical bits in a row, at least every third transmitted bit is a binary 0 and at least every third transmitted bit is a binary 1.

Hence, at least every sixth frame must measure a minimal intensity value and at least every sixth frame must measure a maximal intensity value. In a neighborhood of 6 frames of each frame $n$ there are minimal and maximal measured intensities that can be used to normalize the measurement. With a known minimum $A_0(n)$ and a known maximum $A_1(n)$ for the neighborhood of frame $n$ we normalize the intensity in frame $n$ as:

$$N(n) = \frac{A(n) - A_0(n)}{A_1(n) - A_0(n)}.$$  (2)

What is left to do is to find the minima/maxima that are to be used for this normalization. Let us describe how this is done for the minima, the maxima work analogously.

The idea is to find a sequence of support frames $s_i$ with $A(s_i) = A_0(s_i)$.

For the frames between two support frames we use a linear interpolation to determine their $A_0$.

Assume that frame $s_i$ is a support frame. Since by design there must be a new minimum within the next 6 frames, we determine the gradient to the next six intensities and pick the smallest:

$$s_{i+1} = \arg \min_{n\in\{s_i+1,...,s_i+6\}} \frac{A(n) - A(s_i)}{n - s_i}.$$  (4)

Starting from $s_0 = 0$, this process yields a sequence of minima (support frames), each of which is in place for at most 6 frames.
This construction process of \( A_0(n) \) and \( A_1(n) \) ensures that
\[
A_0(n) \leq A(n) \leq A_1(n),
\]
which implies \( N(n) \in [0, 1] \) for all frames \( n \).

Fig. 3(b) shows the support values and the linear interpolations of the minima \( A_0(n) \) (orange) and maxima \( A_1(n) \) (black). Fig. 3(c) holds the resulting normalization to \([0; 1]\).

\[ \text{Fig. 3(b):} \quad \text{support values and linear interpolation.} \]
\[ \text{Fig. 3(c):} \quad \text{resulting normalization.} \]

**B. Signal Reconstruction**

The purpose of the signal reconstruction step is to determine the approximate points in time within the duration of a frame when the signal changes from \( I_0 \) to \( I_1 \) or vice versa. This is not trivial, as transmitter and receiver are unsynchronized, i.e., signal edges may not only happen at the beginning of a frame, but also during a frame, i.e., while the camera is collecting the light.

For the reconstruction of the signal, we make use of two facts. First, during the exposure of a frame \( n \), the signal changes either exactly once or not at all. If it does not change, the normalized intensity (small blue line segments in Fig. 3(d)) is either close to 0 (\( N(n) < \epsilon \)) if the signal was at \( I_0 \) for the full duration of the frame, or close to 1 (\( N(n) > 1 - \epsilon \)) if the signal was as \( I_1 \) for the full frame. We use \( \epsilon \) to account for measurement errors and found \( \epsilon = 0.2 \) to work well in practice. In those cases the rectangular wave in Fig. 3(d) that approximates the transmitted signal (and that is the result of the receiver side signal reconstruction) is either at the \( I_0 \)-level or at the \( I_1 \)-level for the whole length of the frame.

Second, the normalized intensity \( N(n) \) is approximately the same as the fraction of the exposure duration of the frame \( n \) during which the marker was set to the high intensity \( I_1 \). To see this, assume the signal is at \( I_0 \) at the beginning of a frame and changes to \( I_1 \) after 70\% of the length of the frame. In this case, the camera collects \( I_0 \)-level light for \( \frac{2}{3} \) of the frame and \( I_1 \)-level light for the remaining \( \frac{1}{3} \). The accumulated normalized intensity therefore is \( i = 0.3 \). Hence, the normalized intensity \( i \in [0, 1] \) is also an indicator for the fraction of the frame after which the signal must have changed. Hence in this case the rectangular wave in Fig. 3(d) switches from \( I_0 \) to \( I_1 \) after an \( (1 - i) \)-th of the frame. For a switch from \( I_1 \) to \( I_0 \) it toggles after an \( i \)-th of the frame.

The red rectangular wave in Fig. 3(d) captures the result of the signal reconstruction. It toggles between two states without being synchronized to the frame rate.

\[ \text{Fig. 3(d):} \quad \text{resulting binary data.} \]

**C. Data Derivation**

The final step derives the original binary data from the reconstructed binary signal. The idea is to measure the duration for which the rectangular wave of Fig. 3(d) stays at an intensity level and to use this information to decide whether one or two bits were transmitted during this time.

Ideally, a duration of an intensity level would either be exactly 2.0 (for one transmitted bit) or exactly 4.0 (for two bits). However, due to noise, measurement errors, and inaccuracies introduced in the previous steps, the measured durations may deviate slightly from these ideals.

\[ \text{Fig. 4:} \quad \text{L.I.N.K. test center. Four trajectories A (red), B (green), C (blue), and D (purple) used to evaluate the impact of distance and velocity on the transmission. Position E for measuring the bit error ratio (BER). F–H for ambient light experiments. I–K for interference experiments.} \]

We found that it works well in practice to decode durations of less than 3.0 frames to one bit and durations of more than 3.0 frames to two bits. Recall that three or more identical bits in a row are illegal by design. Fig. 3(d) gives the resulting binary data in red.

\[ \text{Fig. 4:} \quad \text{L.I.N.K. test center. Four trajectories A (red), B (green), C (blue), and D (purple) used to evaluate the impact of distance and velocity on the transmission. Position E for measuring the bit error ratio (BER). F–H for ambient light experiments. I–K for interference experiments.} \]

**V. Evaluation**

For gauging the performance of the marker identification in outside-in camera-based localization systems there are two main aspects that matter: the identification accuracy and the object disambiguation capabilities. Since for the accuracy the crucial ingredient is the reliability of the transmission, this is what we quantitatively evaluate here. We discuss how the velocity of a marker and its distance from the anchor affects the reliability of the transmission, and we gauge the effect of ambient light on the transmission quality. We discuss the disambiguation capabilities of our system by measuring how it can separate interfering markers.

In the experiments there is one anchor. The CN is a Raspberry Pi 3 Model B [21], equipped with an official Raspberry Pi NoIR Camera V2 [22] that is set to a resolution of 1920 \( \times \) 1080 at 30 fps. Markers are single infrared LEDs (Osram SFH 4232A: 850 nm wavelength, \( 1 \times 1 \) mm size) controlled by an ATTiny85 microcontroller. We perform our experiments in the Fraunhofer IIS L.I.N.K. (localization, identification, navigation, communication) test center, see [1]. This is a large hall measuring 44 m \( \times \) 30 m, as depicted in Fig. 4.

A. Distance & Velocity

For evaluating varying distances and velocities, we mount a single marker on a vehicular robot platform and let it move with a configurable velocity along the trajectories shown in Fig. 4. The ambient light is negligible in all tests.

For each of the first four tests A–D the robot repeatedly travels back and forth on a straight line for 5 minutes (9 000 frames at 30 fps). In that time span the robot’s marker
Fig. 5: Absolute intensities $A(n)$ over 2500 frames for test D. Local minima $A_0(n)$ in orange, maxima $A_1(n)$ in black.

sends 4500 bits. From the turning points the robot accelerates linearly up to a speed of $4 \frac{m}{s}$ in the middle of the trajectory before it linearly decelerates until it stops and turns around. The trajectories A–C are 15 m long. The minimal distances to the camera are 28 m, 24 m, and 20 m, resp. In trajectory D the distance to the camera varies between 11 m and 28 m.

In each of the four tests transmission was perfectly reliable, regardless of the distance to the camera and for speeds of up to $4 \frac{m}{s}$. We observed at most one bit error. If at all, it happened at the very beginning of the transmission, i.e., when the intensity normalization could not use a previous support frame for the local minimum $A_0(n)$ and maximum $A_1(n)$.

As an inappropriate support frame caused the initial bit errors, we studied how good the support frames and their local minima and maxima can keep up while the markers move. Fig. 5 shows the measured intensities for the first 2500 frames (1 min 23 s) of test D. The local minima and maxima (orange and black) change significantly in this test. The waves are similar but less pronounced in tests A–C. Valleys happen when the marker is far away from the camera and therefore smaller. At the peaks it is close and therefore larger.

For all frames of all tests the normalized intensity values were inside the $[0; 1]$-interval. Moreover, the minima and maxima changed in line with the distance and the motion. Between any two frames $n$ and $n+1$ the minima/maxima did not change by more than 0.05 %.

Although we only tested speeds of at most $4 \frac{m}{s}$, we expect a reliable transmission for higher velocities as well.

### B. Bit Error Ratio

We did not observe enough bit errors during tests A–D to evaluate the impact of velocity on the bit error ratio (BER). We expect acceleration/deceleration to affect transmission reliability more than velocity, though future work is needed to prove this hypothesis.

To estimate the BER, test E runs much longer (17.6 h) and uses a stationary marker that is farther away (44 m) from the camera than in tests A–D. Among the 948273 transmitted bits we detected 71 bit errors. This results in a bit error ratio of

$$BER = \frac{71}{948273} \approx 7.45 \cdot 10^{-5}$$

for a distance of 44 m and with only negligible ambient light.

\[\text{(a) Exemplary frames for all three tests. The ambient light visibly increases from left to right.}\]

\[\text{(b) Box plots of the measured intensities of all three tests. Boxes from the first to the third quantile; orange lines: median; whiskers at 2 % and 98 %}.\]

**Fig. 6:** Comparison of the three ambient light tests.

### C. Ambient Light

Ambient light and a bright environment reduces the signal-to-noise ratio and impacts transmission reliability. We performed three tests F–H with ambient light of different intensity, that also includes light of the IR wavelength. For F–H a stationary marker is 2 m away from the camera. A heating lamp close to the camera aims at the marker and illuminates the scene with infrared light. This heating lamp emits light of a constant intensity. The signal-to-noise ratio of the average brightness of the marker to the average brightness of the background ranges from 1.59 in test F over 1.38 in test G down to 1.16 in test H. Each test runs for 1 h ($= 273600$ s, $= 78000$ frames, $= 390000$ bits). Fig. 6(a) shows one frame from each test. The box plots in Fig. 6(b) illustrate the distribution of the measured intensities over all frames.

In the three tests we saw 3, 4, and 310 bit errors, i.e., BERs of $7.69 \cdot 10^{-5}, 10.2 \cdot 10^{-5}$, and $7.95 \cdot 10^{-3}$.

The box plots in Fig. 6(b) suggest a reason for the growing BER: The brighter the ambient light is, the smaller is the difference between the two intensities $I_0$ and $I_1$.

For very bright environments, e.g., the sunny outdoors, this hinders the decoding process. For a moderate brightness or for artificially lit environments that in general are free of infrared light, the decoding still achieves the same BER that we saw for darkness.

### D. Interference

Interference between two close markers impacts transmission reliability, as the light emitted by one marker affects the intensity measurements in the other marker trail. To measure the strength of this effect we position two markers with a small distance $d$ between them 2 m away from the camera. As the range of $d$ is small (F: 125 mm, G: 45 mm, H: 35 mm) there is only one dot for them in Fig. 4. Fig. 7 shows one representative frame from each test. Measurements were performed with negligible ambient light.
In tests I and J, we did not observe any bit errors at all during 20 min of transmission. In test K, among the 39,310 transmitted bits during 40 min we saw 1,467 bit errors for marker 0 and 2,956 bit errors for marker 1. This translates to a BER of 3.73% and 7.52%, respectively.

Fig. 8 shows the measured intensities for a segment of 2,500 frames that contains bit errors. Notice that the local minima (orange) and maxima (black) oscillate heavily within only a few frames. This renders the normalization to become less meaningful and ultimately causes the bit errors.

We conclude that our marker identification is reliable as long as the bounding rectangles around the contours of the markers (that we use to measure the markers’ intensities) do not overlap. If needed, a higher camera resolution can buy a better marker separation.

VI. CONCLUSION

The presented robust and reliable marker identification scheme works for distances of at least up to 44 m and for velocities of at least up to 4 m/s. This significantly advances the state of the art. Moreover, since we never switch off the marker LED, continuous tracking is possible. Even under moderate ambient infrared light, even for markers that are as close as 160 pixels in the camera frame, and even without using any error correcting codes known from the literature (e.g., [23]–[26]), we rarely see bit errors (BER ≈ 7·10⁻⁵) in our Optical Camera Communication.

REFERENCES