Article

A Robust Real-Time Automatic Recognition Prototype for Maritime Optical Morse-Based Communication Employing Modified Clustering Algorithm

Xiaozheng Wang [†], Minglun Zhang ^{*,†}, Hongyu Zhou, Xinglong Lin and Xiaomin Ren

State Key Laboratory of Information Photonics and Optical Communications, Beijing University of Posts and Telecommunications, Beijing 100876, China; wangduo666@bupt.edu.cn (X.W.); zhouhongyu@bupt.edu.cn (H.Z.); xllin199409@gmail.com (X.L.); xmren@bupt.edu.cn (X.R.)

* Correspondence: zhangml@bupt.edu.cn

† These authors contributed equally.

Received: 31 December 2019; Accepted: 7 February 2020; Published: 12 February 2020

Abstract: In maritime communications, the ubiquitous Morse lamp on ships plays a significant role as one of the most common backups to radio or satellites just in case. Despite the advantages of its simplicity and efficiency, the requirement of trained operators proficient in Morse code and maintaining stable sending speed pose a key challenge to this traditional manual signaling manner. To overcome these problems, an automatic system is needed to provide a partial substitute for human effort. However, few works have focused on studying an automatic recognition scheme of maritime manually sent-like optical Morse signals. To this end, this paper makes the first attempt to design and implement a robust real-time automatic recognition prototype for onboard Morse lamps. A modified k-means clustering algorithm of machine learning is proposed to optimize the decision threshold and identify elements in Morse light signals. A systematic framework and detailed recognition algorithm procedure are presented. The feasibility of the proposed system is verified via experimental tests using a light-emitting diode (LED) array, self-designed receiver module, and microcontroller unit (MCU). Experimental results indicate that over 99% of real-time recognition accuracy is realized with a signal-to-noise ratio (SNR) greater than 5 dB, and the system can achieve good robustness under conditions with low SNR.

Keywords: automatic recognition; Morse code; k-means clustering; machine learning; maritime communication; real-time; free space optical communication

1. Introduction

Free space optical (FSO) communication has attracted widespread attention from both academia and industry. It has been demonstrated as a promising technology as a complementary solution to conventional radio frequency (RF), fiber optics, and microwave communications, particularly for applications with size, weight, and power restrictions. The potential advantages of FSO communication including eye safety, no spectrum licensing issues, smaller and lighter payloads, low probability of intercept, and immunity from jamming make it a natural candidate for maritime long-range communication scenarios [1–4].

Among the various modulation schemes, Morse code is still widely used as a common language between maritime vessels due to its simplicity, efficiency, and small bandwidth costs. Traditionally, the ubiquitously deployed Morse lamp with LEDs, particularly on naval ships, is operated by a signalman translating the message into Morse code (dots and dashes) and manually turning the lamp “on” and “off” in the correct sequence (on–off keying), to transmit a light signal. On the recipient ship, the blinking

light signal is visually observed and directly interpreted by persons trained in the skill. However, there remains several outstanding issues to this kind of manual signaling method nowadays [5,6]. First, it would require a long training period for a skilled recipient on shipboard. In practice, Morse code relies on precise time intervals between elements while operators have difficulties in maintaining an absolutely stable sending pace, the data length may vary even if it is operated by the same person, and, thus, the reliability of communication cannot be guaranteed [7]. Moreover, as it is long-term and repetitive work, the recognition accuracy is susceptible to the physical and psychological conditions of the human body, which can make them unwieldy in an emergency as well [8].

To cope with the aforementioned issues, research efforts have been made for Morse code automatic recognition in wireless communications. Previous related works are mostly based on signal processing tools, such as Kalman filters, phase-locked loops, and time–frequency analysis [8–11]. Wei et al. [9] proposed an automatic method to lock the frequency of Morse code based on the phase-locked loop circuit. Xiao et al. [10] presented an automatic reception approach for high-frequency (HF) continue-wave (CW) telegrams. A Kalman filtering algorithm combined with the support vector machine (SVM) is utilized to extract the time-domain characteristics of CW signals and deal with unstable code speed. The authors of [11] adopted the Cooley–Tukey Fast Fourier Transform (FFT) algorithm to analyze the spectrogram of a noisy audio Morse signal, and implemented a real-time decoder based on a digital signal processor. In [12], a wavelet transform-based automatic decoding method was proposed for multiplexed Morse telegraph recognition, and the feasibility of this method was analyzed via experiments.

With the further advances of computers and embedded technologies, various high-performance microcontrollers have come out, facilitating the implementation of automated process control, signal recognition, data processing, and algorithm verification. Furthermore, motivated by the remarkable achievements of machine learning, some machine learning (ML)-based approaches have been proposed to model Morse signals and enhance the system robustness in a time-varying noisy environment. Wei et al. [13] provided a machine-learning method for automatic Morse signal detection. An SVM classifier, named HSVM, was proposed to be trained, based on graphical features extracted from the Morse spectrum. The authors of [14] combined signal processing and the deep learning method to construct a Morse identification system, an improved feature extraction algorithm was proposed, and the experimental results indicated a better result. Yuan et al. [15] proposed a deep learning framework on blind Morse signal detection in the wideband spectrum and achieved state-of-art performance using real-world datasets. In [16], a k-means clustering algorithm was introduced to differentiate the elements in Morse signals after parameter extraction of the time–frequency image, and the suitability of the proposed method was investigated by simulation under different signal-to-noise ratio (SNR) conditions. In addition, Qu et al. [17] used this k-means clustering algorithm for dynamic threshold and Morse code recognition in a noisy environment. However, existing works have focused on Morse telegraph signals recognition in RF wireless communications, and, to the best of our knowledge, no reported work has tried to present an automatic recognition scheme and performance analysis for maritime manually sent-like Morse light signals. Besides, the ML-based k-means algorithms have not been well studied in this case as well, and there is no related real-time experimental measurement yet.

To address these issues, we first propose and experimentally demonstrate an automatic recognition framework of Morse light signals in maritime optical communications by an ML-based clustering approach. Specifically, we implement a flexible FSO hardware prototype to collect real-measured Morse light signals in the physical environment for the recognition algorithm test. A modified k-means clustering is designed to compute the decision threshold for binarization and identify elements including dot–dash and space recognition in Morse light signals. To improve the decoding efficiency, we make modifications to the classical k-means algorithm, and a selection sorting-assisted method has been adopted. To tackle with the emerging glitches in the binarization waveform, resulting from strong noise channels, a novel error correction scheme is presented to enhance the robustness to ambient light noises. The proposed recognition approach is implemented in a microcontroller unit (MCU), and the

performance of the prototype is analyzed in terms of recognition accuracy and system robustness via real-time decoding results derived from the MCU.

2. Principle

2.1. Optical Morse Signal Characterization

Morse code is a classical encoding scheme in telecommunications composed of a unique sequence of short–long dots, dashes, and spaces to represent different letters of the alphabet, numbers, and procedural notations. It is usually transmitted by on–off radio, light, or tones [18].

While using light as a transmitting medium, the dots and dashes are represented as high level (light on), and the spaces or pauses refer to low level (light off). In addition, the Morse code depends on accurate time intervals, which correspond to the pulse width in the signal waveform. The duration of a dot is regarded as a basic unit, and the dash duration is three times that. The spaces or time intervals between dot–dashes, between letters, and between words are 1, 3, and 5 units, respectively [19]. As illustrated in Figure 1, it is more like the Return-to-Zero OOK (RZ-OOK) scheme in optical wireless communications. Unlike RF, the optical Morse-encoded signal should be real and non-negative due to the intensity modulation/direct detection (IM/DD) mechanism [20].

Figure 1. Waveform of optical Morse signal interpreting “SOS.”

As mentioned above, the recognition is achieved based on the different time intervals and signal amplitude to distinguish different elements. However, the automatic recognition of manually sent Morse light signals in the real world is facing two obvious challenges. One is that the unstable typing pace can lead to variations in different elements durations, and the other is the amplitude fluctuation resulting from ambient light noise.

Assume the time intervals of dots, dashes, and spaces in unstable Morse light signals are distributed in a nearly normal fashion [21]. Let us denote D_1 and D_2 as the dot and dash durations samples, respectively; then, the probability density function (PDF) is expressed as

$$f(D) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(D - \mu)^2}{2\sigma^2}\right), \quad (1)$$

where $D_1 \sim N(\mu, \sigma^2)$, $D_2 \sim N(3\mu, \sigma^2)$, μ denotes the means, and σ is the scale parameter. By controlling proper μ and σ^2 , it is able to generate manually sent-like Morse light signals based on the Morse encoding rules as the data source to verify the designed algorithm later. A histogram of the distribution of different element durations is shown in Figure 2, and it can be clearly seen that the duration ratio changes. Generally, the durations of different elements in the two figures nearly follow a ratio of 1:3 and 1:3:5:7, respectively. In addition, additive white Gaussian noise (AWGN) is considered to be added to the raw Morse data. Figure 3 plots the signal waveform in the time domain with/without noise. It is observed that the noise will cause amplitude fluctuation, but not affect the duration of different elements. In addition, these duration and amplitude variations can lead to errors for recognition.

Figure 2. Histogram of (a) space and (b) dot-dash duration distributions.

Figure 3. Time domain of optical Morse signal with additive white Gaussian noise (AWGN).

2.2. Modified K-Means Clustering Algorithm

The k-means algorithm is one of the most commonly used clustering methods due to its simplicity and efficiency. The basic idea is to partition data samples into k clusters based on the calculated distance between each element and centroid [22–24]. As the practical FSO channels suffer from many factors such as ambient light noise and multi-path dispersion [20], this method is able to optimize the decision threshold of the time-varying received signal and reduce the error rate in this noisy environment. Furthermore, as analyzed in Section 2.1, a notable clustering feature is observed in terms of the element distribution of optical Morse signals, which can be effectively differentiated using the k-means clustering algorithm. The detailed procedure in this case is as follows:

Let X denote the sampled voltage data set, and initialize k cluster centroids $C = [C_1, C_2, \dots, C_k]$ randomly from X , compute the Euclidean distance between each voltage sample and centroid to find the positions of the clusters' centroids minimizing the cost function [21], and then assign each sample to its closest cluster centroid, defined as

$$j = \arg(\min_{1 \leq i \leq k} \|X - C_i(m)\|^2), \quad (2)$$

where $\|\bullet\|$ represents the Euclidean distance, $C_i(m)$ is the i -th cluster centroid during the m -th iteration, $X \in S_j(m)$, and $S_j(m)$ denotes the cluster with the centroid of $C_j(m)$. Then, update the centroid of each cluster by calculating the mean as follows:

$$C_i(m+1) = \frac{1}{n_i} \sum_{X \in S_i(m)} X, \quad (i = 1, 2, \dots, k), \quad (3)$$

where $C_i(m+1)$ is the i -th cluster centroid during the $(m+1)$ -th iteration, n_i stands for the number of data samples in the i -th cluster, $S_j(m)$ denotes the cluster with the centroid of $C_j(m)$, and X represents all the data samples in cluster $S_j(m)$. If the updated centroids remain the same, it indicates that convergence is reached and then stops, described as

$$\text{abs}([C_i(m+1) - C_i(m)]) < \text{abs}([C_i(m) - C_i(m-1)]) \times 2\%, \quad (i = 1, 2, \dots, k). \quad (4)$$

Otherwise, repeat the step until it reaches convergence.

It is noted that there are also another enhanced clustering method and modified version of the Euclidean distance that yield good performance for high-dimensional datasets [25–27]. The data to be processed by the proposed automatic recognition algorithm include voltage value and level duration, both of which are one-dimensional datasets. Thus, for the ease of implementation in the embedded system, the simplest k-means method and Euclidean distance are chosen in this work, which can also effectively meet the requirement.

In order to adapt this algorithm to the application scenario, we propose several modifications to the clustering process to improve the recognition accuracy. The optimization of the k-means clustering algorithm is as follows:

1. **Cluster Centroid Initialization:** The digital filter will cause an amplitude level hopping effect in the head and tail of the passing signal, leading to significant deviation. Thus, this part of the signal is designed to be abandoned in this case while selecting the cluster centroid during the iteration.
2. **Selection Sort-Assisted (Figure 4):** It can be easily known that the obtained k cluster centroids are in an unordered state. As the purpose of cluster analysis is to distinguish the signals with different durations in the Morse decoding, it is necessary to sort the final k centroids so that the device can effectively identify different clusters. Considering the small quantity of centroids and easy implementation in this case, we exploit the widely used selection sort algorithm to rearrange the centroids in a certain order [28].

Figure 4. Selection sort process.

3. System Overview

A schematic diagram of the proposed system is shown in Figure 5, composed of a source computer, arbitrary waveform generator (AWG), LED driver, LED source, lens, receiver module, analog–digital converter (ADC), and MCU. First, the information bits are encoded by a PC into an unstable Morse code set and converted to an analog signal by AWG. Then, the signal is amplified and modulated to drive the LED via the driver circuit for transmitting visible light signals. At the receiver side, the incoming light signal is received by a photodetector (PD) for photoelectric conversion and amplified by a transimpedance amplifier (TIA). Then, the ADC is used to convert the electrical analog signal into a digital signal. Afterward, the digital signal is fed to the MCU for further processing.

Figure 5. Block diagram of hardware setup.

The Cortex M4-based MCU STM32 is the core part of this whole hardware system, which has a rich peripheral interface with high integration, composed of a serial peripheral interface (SPI), finite impulse response (FIR) filter, timers (TIM), universal synchronous/asynchronous receiver transmitters (USART), and a decode module. The control signal and sampled data stream are transmitted between the MCU chip and ADC using a SPI. After FIR denoising, the sampled signal is further processed by the decode module, where the recognition algorithm is implemented. Then, the USART module is used for the interaction between the automatic recognition system and the host computer, including the transmission of control commands and return of the decoding results. In addition, the TIM module provides a working clock for the entire system to ensure that each functional component can run in an orderly manner.

As indicated in Figure 6, the implementation of our proposed recognition algorithm contains three stages: Signal pretreatment, Morse decoding, and accuracy analysis. The input sampled signal is filtered first by the FIR module to reduce the effects of noise on the subsequent process. Then, the threshold between the high level and low level is determined to classify all the voltage samples into two levels (0 and 1) in the stage of binarization. As there still exist some irregular hopping points after FIR denoising, which will affect the automatic decoding, resulting in the decline of recognition accuracy, the error correction stage is designed as a necessary complement to address this issue by eliminating the spikes in the signal. Afterward, a relatively pure and stable Morse code signal is obtained and classified according to the characteristics of the dot, dash, and space code. After utilizing the dot–dash and space recognition algorithm, the identified Morse code set is able to be interpreted

into a string message sequence by table lookup decoding, and is then finally output to the computer for accuracy computation.

Figure 6. Flowchart of optical Morse signal processing.

4. Signal Pretreatment

The pretreatment stage includes digital filtering, binarization, and hopping points correction. The purpose of this stage is to provide a pure and stable Morse code signal for the subsequent decoding process. Detailed signal processing is provided in the following subsections.

4.1. Digital Filter Denoising

Considering that low-frequency components dominate in the frequency domain of the optical Morse signal (Figure 7), we adopt a FIR low-pass filter with a cutoff frequency of 40 Hz using the Hamming window to mitigate the noise effect, and for the ease of implementation in the MCU as well.

Figure 7. Frequency domain of optical Morse signal.

4.2. Binarization

In this stage, the proposed modified k-means clustering algorithm is utilized to compute a classification voltage threshold V_T and partition the voltage samples into two levels, representing “1” (i.e., high level) and “0” (i.e., low level), respectively. The specific steps of this process are as follows:

1. Initialize a similarity value s , to quantify how similar two clusters are to one another;
2. Perform the mk-means algorithm to classify all the voltage data samples into two sets with the cluster centroids of C_{V1} and C_{V2} ;

3. Calculate the distance between two centroids by $d = \text{abs}(C_{V2} - C_{V1})$. If $d < s$, it means that two clusters are similar. At this time, all sampling values are divided into high or low level. Then, set $V_T = 0$; otherwise, set $V_T = (C_{V2} + C_{V1})/2$;
4. Iterate over all data samples; a value above the obtained threshold in step (3) indicates “1;” meanwhile, a value below indicates “0.”

4.3. Error Correction

In many cases, there exist multiple unexpected glitches in signals after binarization due to the noise effects. As Figure 6 shows, the red circle marks in the block binarization indicate hopping points. Thus, we design the error correction algorithm to eliminate these abnormal hopping voltages and guarantee an accurate subsequent decoding process.

The first step is to compute a threshold of durations for the purpose of localizing the emerging glitch positions. By using the algorithm in Section 2.2, duration data corresponding to each level can be partitioned into three clusters with three centroids in an ascending sort order, denoted by $C_d = [C_{d1}, C_{d2}, C_{d3}]$. Then, if $C_{d2}/C_{d1} > 3.5$, it means that data samples clustered around C_{d1} are from the glitches, and the basic dot duration unit is given by $D_0 = C_{d2}$. Otherwise, it means the waveform is good, no jump level exists, and $D_0 = C_{d1}$. Then, the duration threshold can be obtained as $D_T = e_1 * D_0$, where e_1 denotes an adjustable parameter. In addition, we assume $e_1 = 0.3$ at this time. The ratio 3.5 and $e_1 = 0.3$ are established based on the experimental dataset observations for the optimal decoding result.

To increase the identifying accuracy, we introduce the average voltage amplitude as the second judgement criteria beyond the duration data. The average voltage amplitude refers to the statistical average of all sampled voltage values during a period of one level (“0” or “1”). Assume two new voltage thresholds, described as

$$\begin{aligned} V_{TH} &= V_T + V_{pp} \times e_2 \\ V_{TL} &= V_T - V_{pp} \times e_2 \end{aligned} \quad (5)$$

where V_T is the binarization voltage threshold computed in Section 4.2, V_{pp} denotes the peak–peak voltage value, and e_2 represents another adjustable parameter; it is also established based on the experimental dataset observations and is set at 0.25 in this case for the optimal decoding result. In addition, using a different value may lead to erratic performance in our experiment.

Let us denote D_c and V_{ac} to the current duration and average voltage value during one level, respectively; then, the glitches can be localized under the following constraint:

$$D_c < D_T \parallel V_{TL} < V_{ac} < V_T \parallel V_T < V_{ac} < V_{TH}. \quad (6)$$

After marking the hopping glitches, it is time to identify their position relations to the normal voltage level. To be precise, there should be six different statuses of the emerging voltage spikes, which can represent all potential situations. In addition, the voltage spike can be either low or high. As long as the duration of a level is less than the computed duration threshold, it is considered as a voltage spike and merging is performed until meeting a complete voltage level (duration higher than threshold). These six cases are distinguished according to whether the transition part can form a complete level and its adjacent level condition (high or low).

As the Figure 8 shows: (a) If several consecutive spikes can form a complete level after merging their duration (no longer a spike) and its adjacent level is low, then the combined one is set at high, denoted by the red line; (b) similar to (a), if the adjacent is high, then the combined one is set at low; (c) if the combined spike is still a spike after merging their duration and its adjacent level is low, then change it to low level, denoted by the red line; (d) similar to (c), if the adjacent level is high, change it to low level; (e) and (f) if the combined spike still cannot form a complete level after merging their duration, and its previous and next complete levels are of different types (one high and one low), then superimpose half of its duration time to the preceding and succeeding complete levels, respectively.

Figure 8. Error correction cases.

Take the string “SOS” as an example; Figure 9 presents its corresponding signal waveform during the whole pretreatment process. In addition, some of the error cases mentioned above are marked by the vertical red lines. It can be clearly seen that the emerging voltage spikes after binarization are effectively eliminated using this method, and a pure and stable Morse-encoded signal interpreting “SOS” is derived after the designed pretreatment stage.

Figure 9. Pretreatment of Morse signal interpreting “SOS”.

5. Morse Code Decoding

Considering that we have obtained a relatively stable Morse-encoded signal after the pretreatment stage, it is easy to identify the elements using the k-means algorithm and derive the decoded data based on the Morse code table. This stage includes dot–dash recognition, space recognition, and accuracy computation.

5.1. Dot–Dash Recognition

Considering that both dots and dashes are represented by “1” (high level) according to Morse encoding rules, the only difference between them is the time duration. The dot duration is the basic unit of time measurement in code transmission, while the dash duration is three units. As the time duration of the voltage level changes after pretreatment, it is necessary to recalculate the dot, as well as the

dash, duration thresholds. Thus, the mk-means clustering algorithm is performed to accomplish the classification of different high-level samples and dot–dash recognition. Figure 10a shows a flowchart of our dot–dash recognition algorithm:

1. Extract the high-level samples output after signal pretreatment, stored in an array;
2. Perform the mk-means method to partition these data samples into two clusters, and store the classification results in a new array HA ;
3. Sort the two cluster centroids in ascending order, stored in an array CH , and ensure that $CH(0)$ is less than $CH(1)$. If the order of the cluster centroids changes, then update the centroid index stored in HA ;
4. Assume the basic dot duration computed above is D_0 and $s = CH(1)/CH(0)$. If $s < 2$, it indicates that the two clusters belong to one common category. At this point, if $CH(1)/D_0 \leq 2$, then it means all the samples in HA denote “dot,” and all indexes in the array need to be updated to 0; else, if $CH(1)/D_0 > 2$, then it means HA represents “dash,” and the indexes should be updated to 1.

Figure 10. Flowchart of (a) dot–dash, and (b) space recognition algorithms.

5.2. Space Recognition

In Morse coding rules, space refers to a period of signal absence following each dot or dash, which represents 0 (low level) in binary optical signals. Spaces between adjacent dots and dashes, two letters, words, and sentences are 1, 3, 5, and 7 units long, respectively. Therefore, a similar concept is used to classify all the low-level samples into these four statuses by calculating the time duration threshold. Assume $D_n = CH(0)$ is the new computed basic duration, used to initialize the threshold, and it is noted that while two clusters both indicate “dash” at the same time, D_n is supposed to be 1/3 of the smaller cluster centroid value. As illustrated in Figure 10b, the space recognition results are stored in

an array LA , where the element value indicates the exact time duration units. For example, 3 denotes 3 units, representing spaces between adjacent letters.

5.3. Accuracy Computation

To quantify the accuracy of our recognition algorithm, the Levenshtein distance (edit distance) is used by counting the minimum number of edit operations required to transform one string into the other. The modification operations include: (a) Insert a character; (b) delete one character; and (c) replace one character with another. Let us denote $R = [r_1 r_2 \dots r_i]$ and $D = [d_1 d_2 \dots d_j]$ as the raw data string and the decoded data string, respectively. $m[i, j]$ represents the edit distance between the first i characters of R and the first j characters of D , and is expressed as [29]

$$m(i, j) = \begin{cases} 0 & i = 0, j = 0 \\ j & i = 0, j > 0 \\ i & i > 0, j = 0 \\ \min(m(i-1, j) + 1, m(i, j-1) + 1, m(i-1, j-1) + flag) & i > 0, j > 0 \end{cases}, \quad (7)$$

where the flag variable is the indicator function equal to 0 when $R[i] = D[j]$, and equal to 1 otherwise. Then, we have the recognition accuracy,

$$A = 1 - \frac{|m(i, j)|}{\max(|R|, |D|)}, \quad (8)$$

where $|R|$ and $|D|$ represent the length of the raw and decoded data string, respectively. In addition, the procedure of this dynamic programming algorithm to derive the edit distance is shown in Algorithm 1:

Algorithm 1 String similarity algorithm

```

1: procedure Levenshtein Distance ( $R, D$ )
2: int  $m[i, j] = 0$ 
3: for each  $i \leftarrow 1$  to  $|R|$  do
4:  $m[i, 0] = i$ 
5: end for
6: for each  $j \leftarrow 1$  to  $|D|$  do
7:  $m[0, j] = j$ 
8: end for
9: for each  $i \leftarrow 1$  to  $|R|$  do
10: for each  $j \leftarrow 1$  to  $|D|$  do
11:  $t \leftarrow$  if  $(R[i] = D[j])$  then 0 else if 1
12:  $m[i, j] = \min\{m[i-1, j-1] + t, m[i-1, j] + 1, m[i, j-1] + 1\}$ 
13: end for
14: end for
15: return  $m[|R|, |D|]$ 
16: end procedure

```

6. Experimental Validation

6.1. Experimental Setup

The experimental testbed for our proposed system prototype is shown in Figure 11; we use this prototype to generate a real optical Morse encoded signal and verify the feasibility of our proposed automatic recognition approach.

Figure 11. Demonstration of proposed free space optical (FSO) prototype: (a) Transmitter side, (b) receiver side.

At Tx, an unstable Morse encoded signal is first produced by an arbitrary waveform generator (AWG 7051). Then, it is modulated to the LED (XLamp® XP-L2) driven by the self-designed driving circuit packed inside the lamp. Note that a signal lamp consisting of LED arrays is employed in our system for the purpose of simulating the maritime optical communication scenarios. At Rx, we use an optical lens (ZLKC-KM5012MP8) with a focal length of 50 mm to narrow the field-of view (FOV) of the receiver, thereby mitigating the incoming ambient light noise. The emitted light signal is captured by a self-designed optical receiving module including an avalanche photo-diode (S8664-50K) and a transimpedance amplifier (LTC6268-10) with a high gain factor of 2 million times. Then, after analog–digital conversion by ADS8866 (16-bit resolution), the signal is applied to the STM32F446 embedded system [30] and processed based on our proposed approach. The sampling period is $T_s = 2.7$ ms, and the sampling rate of the ADC is set at $F_s = 375$ Hz to guarantee that there are at least 20 C30 samples during the period of a basic dot unit.

Finally, the output recognition results are analyzed on the computer in real time using the algorithm mentioned in Section 4.3, and the accuracy performance is determined. The key parameters of the devices used in this system are listed in Table 1.

Table 1. System configuration parameters.

Device	Manufacturer	Model
AWG	Tektronix	AWG 7051
LED	Cree Inc	XLamp® XP-L2
Lens	ZhiSai Technology	ZLKC-KM5012MP8
APD	Hamamatsu Photonics	S8664-50K
TIA	Linear Technology	LTC6268-10
ADC	Texas Instruments	ADS 8866
MCU	STMicroelectronics	STM32F446

6.2. Recognition Accuracy Evaluation

In the experiments, a Morse-encoded signal interpreting the string “hello world” is generated on the computer with MATLAB using the method in Section 2.1, and the durations and amplitude fluctuation are considered by controlling the μ and σ^2 and adding AWGN, respectively. We first investigate the offline performance. Figure 12 shows the off-line pretreatment results based on the mk-means clustering algorithm. One can observe that the obtained waveform is just the same as the original signal, which indicates that our method works well.

Figure 12. Simulation results of Morse signal interpreting “hello world.”

Then, the prototype is used to test the real-time decoding accuracy. The optical signal emitted from LED arrays is received by the photo detector and then processed in the embedded system; the translation results are obtained by the serial port software on the computer. Figure 13 displays the real-time recognition results in the Serial Assistant software derived from STM32. It is also observed that the decoding results are exactly the same as the raw string “hello world.” It is worth noticing that the transceiver was originally designed for the long-range outdoor environment and can support 4.8 km FSO communication [31]. The transmitter consists of more than 30 LEDs, each at a DC forward current of 1500 mA and a corresponding electrical power of 4.35 W, which is enough for indoor optical wireless communication. Thus, the received optical power maintains a high level in the laboratory environment, and the recognition accuracy also remains the same after increasing the distance from 1 to 6 m.

Figure 13. Real-time optical Morse code recognition results.

6.3. System Robustness Evaluation

To evaluate the performance and robustness of our system, we investigate the recognition accuracy under different signal-to-noise ratio (SNR) conditions. Considering that it is difficult to accurately control the SNR of the input signal over the optical path, the experiment is conducted between the PC and STM32 through the serial port. During the test, two English texts containing 2196 characters were selected as the raw data, and each character appeared randomly. Note that the transmitting data volume is far more than that of usual use in our considered real maritime scenario.

The specific test procedure is as follows: First, the raw data of length 2196 characters are Morse-encoded and superimposed with noise in MATLAB. By controlling the proportion of noise samples, the Morse-encoded signal data with different SNRs (−3–6 dB) is generated on the PC and

then transmitted to the STM32-embedded system through the serial port; secondly, STM32 stores and decodes the received data and returns the recognition results back to the host computer. Finally, in the host computer, the original and decoded data file are examined, and the decoding accuracy is statistically analyzed. The statistical decoding results during the experiment are listed in Table 2.

Table 2. Recognition results under different signal-to-noise ratios (SNRs).

SNR (dB)	Total Characters	Correct Characters	Accuracy
−3	2196	1979	90.1%
−1	2196	2003	91.2%
1	2196	2038	92.8%
3	2196	2102	95.7%
5	2196	2128	96.9%
6	2196	2176	99.1%

The curve of recognition accuracy versus SNR derived from Table 2 is plotted in Figure 14. It is noticed that the decoding accuracy increases with the growth of the SNR. Our proposed approach can achieve an average automatic recognition accuracy of more than 90% when the SNR of the input signal is greater than −3 dB. Furthermore, an accuracy of over 95% is observed with an SNR greater than 3 dB. In the case that the noise is relatively strong with an SNR of −3 dB, the system can still achieve a good accuracy of 90.1%. Thus, we conclude that, in the presence of ambient light noise, the proposed mk-means based recognition system still performs well in terms of the decoding accuracy performance.

Figure 14. Recognition accuracy under different SNRs.

7. Conclusions

In conclusion, we proposed an automatic recognition scheme for maritime optical Morse lamps based on a modified k-means clustering algorithm. A flexible FSO communication prototype consisted of LED arrays, and an MCU was constructed for real data collection and real-time decoding. We also investigated the performance of the proposed approach including recognition accuracy, as well as system robustness. Real-time recognition results derived from the MCU indicate that the accuracy increases with the growth of the SNR, and can reach more than 99%. It is worth mentioning that our proposed modified k-means clustering algorithm can also be applied in other digital communication systems for decision threshold optimization and is adaptive to different channel characteristics. In the future, we will further investigate the recognition performance based on other ML-based clustering algorithms.

Author Contributions: Conceptualization, X.W. and M.Z.; methodology, X.W. and M.Z.; software, X.W., M.Z., H.Z. and X.L.; validation, X.L.; formal analysis, X.W. and M.Z.; investigation, X.W. and M.Z.; resources, H.Z. and X.L.; data curation, X.W. and M.Z.; writing—original draft preparation, X.W.; writing—review and editing, M.Z.; visualization, X.W., H.Z. and X.L.; supervision, M.Z. and X.R.; project administration, M.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Khalighi, M.A.; Uysal, M. Survey on free space optical communication: A communication theory perspective. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 2231–2258. [CrossRef]
2. Grant, K.J.; Mudge, K.A.; Clare, B.A.; Perejma, A.S.; Martinsen, W.M. *Maritime Laser Communications Trial 98152-19703, DSTO*; Endinburgh: South Australia, Australia, 2012.
3. Sluz, J.E.; Riggins, J., II; Juarez, J.C.; Sova, R.M.; Young, D.W.; Nelson, C. Characterization of data transmission through a maritime free-space optical channel with a custom bit error rate tester. *Proc. Spie* **2010**, *7700*, 77000D.
4. Yu, J.; Li, Q.; Li, H.; Wang, Q.; Zhou, G.; He, D.; Xu, S.; Xia, Y.; Huang, Y. High-Precision Light Spot Position Detection in Low SNR Condition Based on Quadrant Detector. *Appl. Sci.* **2019**, *9*, 1299. [CrossRef]
5. New Atlas. Available online: <https://newatlas.com/us-navy-signal-lamps-fltc-texting/50523/> (accessed on 10 October 2019).
6. Sapaico, L.R.; Sato, M. Analysis of vision-based text entry using Morse code generated by tongue gestures. In Proceedings of the 2011 4th International Conference on Human System Interactions (HSI), Yokohama, Japan, 19–21 May 2011; pp. 158–164.
7. Hsieh, M.C.; Luo, C.H.; Mao, C.W. Unstable Morse code recognition with adaptive variable-ratio threshold prediction for physically disabled persons. *IEEE Trans. Rehabil. Eng.* **2000**, *8*, 405–413. [CrossRef] [PubMed]
8. Popular Mechanics. Available online: <https://www.popularmechanics.com/military/navy-ships/a27391/us-navy-Morse-code-software/> (accessed on 10 October 2019).
9. Ma, W.; Zhang, J.; Wang, H. Automatic Decoding System of Morse Code. *O.I. Autom.* **2007**, *26*, 51–52.
10. Xiao-Na, Z.; Guo-Jun, L.; Yong, J.; Li, Z. Automatic reception of High-frequency CW telegraph with Support Vector Machine. In Proceedings of the 2nd IEEE International Conference on Future Computer and Communication, Wuhan, China, 21–24 May 2010; pp. 335–338.
11. Zahradnik, P.; Šimák, B. Implementation of Morse decoder on the TMS320C6748 DSP development kit. In Proceedings of the 2014 6th IEEE European Embedded Design in Education and Research Conference, Milan, Italy, 11–12 September 2014; pp. 128–131.
12. Xiao, S.; Gao, Y. Multiplexed Morse Telegraph Automatic Decoding Based on Wavelet Transform. *Comput. Digit. Eng.* **2017**, *45*, 632–636.
13. Wei, Z.; Jia, K.; Sun, Z. An Automatic Detection Method for Morse Signal Based on Machine Learning. In Proceedings of the 12th International Conference on Intelligent Information Hiding and Multimedia Signal, Matsue, Japan, 12–15 August 2017; pp. 185–191.
14. Wang, X.; Zhao, Q.; Ma, C.; Xiong, J. Automatic Morse Code Recognition Under Low SNR. In Proceedings of the 2018 International Conference on Mechanical, Electronic, Control and Automation Engineering, Manila, Philippines, 5–7 January 2018.
15. Yuan, Y.; Sun, Z.; Wei, Z.; Jia, K. DeepMorse: A Deep Convolutional Learning Method for Blind Morse Signal Detection in Wideband Wireless Spectrum. *IEEE Access* **2019**, *7*, 80577–80587. [CrossRef]
16. Wang, Y.; Sun, Z.; Jia, K. An Automatic Decoding Method for Morse Signal based on Clustering Algorithm. In Proceedings of the 12th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kaohsiung, China, 21–23 November 2016; pp. 235–242.
17. Qu, S.; Liu, H.; Zhang, X. Morse Recognition Algorithm Based on K-means. In Proceedings of the 2019 IEEE Cross Strait Quad-Regional Radio Science and Wireless Technology Conference, Taiyuan, China, 19–22 July 2019; pp. 1–2.
18. Lee, C.T.; Shen, T.C.; Lee, W.D. A Novel Optical Morse Code-Based Electronic Lock Using the Ambient Light Sensor and Fuzzy Controller. *Appl. Sci.* **2017**, *7*, 140. [CrossRef]
19. Bilal, M.; Liu, S.; Qiao, G.; Wan, L.; Tao, Y. Bionic Morse Coding Mimicking Humpback Whale Song for Covert Underwater Communication. *Appl. Sci.* **2020**, *10*, 186. [CrossRef]
20. Ma, S.; Dai, J.; Lu, S.; Li, H.; Zhang, H.; Du, C.; Li, S. Signal Demodulation with Machine Learning Methods for Physical Layer Visible Light Communications: Prototype Platform, Open Dataset, and Algorithms. *IEEE Access* **2019**, *7*, 30588–30598. [CrossRef]

21. Zhang, X. The Distinction of Ship's Communication Signals. *J. Jimei. Navig. Inst.* **1996**, *14*, 19–25.
22. Zhang, J.; Chen, W.; Gao, M.; Shen, G. K-means-clustering-based fiber nonlinearity equalization techniques for 64-QAM coherent optical communication system. *Opt. Express* **2017**, *25*, 27570–27580. [[CrossRef](#)] [[PubMed](#)]
23. Peng, Y.; Liu, X.; Shen, C.; Huang, H.; Zhao, D.; Cao, H.; Guo, X. An Improved Optical Flow Algorithm Based on Mask-R-CNN and K-Means for Velocity Calculation. *Appl. Sci.* **2019**, *9*, 2808. [[CrossRef](#)]
24. Hou, X. An Improved K-means Clustering Algorithm Based on Hadoop Platform. In Proceedings of the International Conference on Cyber Security Intelligence and Analytics, Shenyang, China, 21–22 February 2019; pp. 1101–1109.
25. Cardarilli, G.C.; Di Nunzio, L.; Fazzolari, R.; Re, M.; Spanó, S. AW-SOM, an algorithm for high-speed learning in hardware self-organizing maps. *IEEE Trans. Circuits Syst. II Exp. Briefs* **2019**, *67*, 380–384. [[CrossRef](#)]
26. Cardarilli, G.C.; Di Nunzio, L.; Fazzolari, R.; Nannarelli, A.; Re, M.; Spanò, S. N-Dimensional Approximation of Euclidean Distance. *IEEE Trans. Circuits Syst. II Exp. Briefs* **2019**, *55*, 161. [[CrossRef](#)]
27. Tolentino, J.A.; Gerardo, B.D.; Medina, R.P. Enhanced Manhattan-Based Clustering Using Fuzzy C-Means Algorithm. In Proceedings of the International Conference on Computing and Information Technology, Chiangmai, Thailand, 5–6 July 2018; pp. 126–134.
28. Heineman, G.T.; Pollice, G.; Selkow, S. *Algorithms in a Nutshell: A Practical Guide*, 2nd ed.; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2016; pp. 53–89.
29. Zhao, Z.; Yin, Z.; Wang, Q.; Xu, X.; Jiang, H. An improved algorithm of Levenshtein Distance and its application in data processing. *J. Comput. Applicat.* **2009**, *29*, 424–426. [[CrossRef](#)]
30. STMicroelectronics 32-bit Arm Cortex MCUs. Available online: <https://www.st.com/en/microcontrollers-microprocessors/stm32f446.html> (accessed on 26 January 2020).
31. Zhou, H.; Zhang, M.; Wang, X.; Ren, X. Implementation of High Gain Optical Receiver with the Large Photosensitive Area in Visible Light Communication. In Proceedings of the Asia Communications and Photonics Conference, Chengdu, China, 2–5 November 2019; pp. 99–101.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).