SHORT PAPER: TOWARDS UNDERWATER VIDEO TRANSMISSION

L. Dubreuil Vall, D. Sura, and M. Stojanovic

MITSG 11-27
Short paper: Towards Underwater Video Transmission

Laura Dubreuil Vall  
Massachusetts Institute of Technology  
Cambridge, MA 02139  
lauradv@mit.edu

Daniel Sura  
Massachusetts Institute of Technology  
Cambridge, MA 02139  
dsurfer@mit.edu

Milica Stojanovic  
Northeastern University  
Boston, MA 02115  
millitsa@ece.neu.edu

ABSTRACT

This paper addresses the feasibility of underwater wireless video transmission using an acoustic system, which consists of a video compressor/decompressor and an OFDM modulator/demodulator. This work includes two main parts: the design of an algorithm aimed to compensate for the Doppler effect and the assessment of a set of error resilience tools offered by the MPEG-4 standard. The results of the simulations and in-air experiments demonstrate the performance improvement in terms of visual quality of the received video in cases with motion, as well as the system robustness against channel induced errors.

Categories and Subject Descriptors

C.3 [Special-purpose and based-application systems]: Signal processing systems; H.4 [Information Systems Applications]: Communications Applications

General Terms


Keywords

Underwater Communications, OFDM, Video Transmission.

1. INTRODUCTION

Real-time underwater wireless video transmission has recently become a demanding communication application for subsea operations in a wide range of fields, such as supervisory control of Autonomous Underwater Vehicles (AUVs) in deep-sea oil fields, port security, environmental monitoring of marine life, and even open ocean exploration, where high resolution sidescan sonar imagery from an AUV could be compressed and sent as video for real-time review.

In the early 1990s a group of Japanese scientists succeeded in transmitting a video of a slowly-moving object at a depth of 6,500 m [8]. The video was a sequence of independently recorded images played at a frame rate of a few frames every 10 seconds, which is quite sufficient to capture the slow motion of an ocean-floor scene. Since then, further research on video compression and modulation techniques has been conducted, enabling the transmission of higher resolution video [1].

The design of any underwater video transmission system faces several challenging problems. On the one hand, video signals typically have large information content, which requires a high bit rate for real-time transmission. On the other hand, the underwater acoustic channel has a limited bandwidth that can only support low bit rates. In order to find a balance between the video bit rate requirements and the limited bandwidth of the channel, the system needs to include a video compression technique that reduces the high bit rate of the video data, while the modulation technique used to transmit the data must efficiently utilize the available bandwidth in order to increase the bit rate supported by the system.

This paper addresses the implementation of an underwater wireless video transmission system built upon the acoustic implementation described in [5]. Such a system is of interest over short distance links, with short delays and high available bandwidth, as near real-time imagery can be transmitted to monitor targeted underwater spots. In that project, the basic structure of the system was designed and the MPEG-4 standard and the OFDM (Orthogonal Frequency-Division Multiplexing) technique were chosen as the most suitable video compression and modulation techniques for the system, respectively.

The purpose of the present research is to continue with the MPEG-4/OFDM approach with the goal of implementing new features to make the system robust to the distortion of the acoustic channel. Three major steps are taken: (i) the design of an algorithm that corrects the synchronization problems in OFDM modulations caused by the Doppler effect, (ii) the use of error resilience tools offered by the MPEG-4 standard, which enable robust video communication over noisy wireless channels by offering additional protection over error bursts and maintaining the visual quality of the decoded video, and (iii) the replacement of the coherent detection method used in the previous design by differentially coherent detection to gain robustness against fast channel variations.

The rest of the paper is organized as follows. In Section 2, the previous system is described. Our contributions are detailed in Section 3. Section 4 presents the experimental results, while conclusions are summarized in Section 5.
ing signal is then fed to the OFDM modulator, which combines spatial and temporal techniques. The resulting video encoder compresses the input video down to 10 kbps coder and the OFDM modulator, along with their inverse

2. SYSTEM MODEL

The system consists of two main blocks: the video encoder and the OFDM modulator, along with their inverse functions at the receiver side (Fig. 1).

First, the MPEG-4 compression technique used in the video encoder compresses the input video down to 10 kbps by combining spatial and temporal techniques. The resulting signal is then fed to the OFDM modulator, which consists of a typical OFDM system with $K$ subcarriers. After scrambling and applying the channel coding, the data stream is serial-to-parallel converted into $K$ streams $d_k(n), k = 0, \ldots, K-1$. These bit streams are interleaved before passing through the IFFT (Inverse Fast Fourier Transform) block. The resulting signal is represented by

$$ u(t) = \sum_{n}^{K-1} d_k(n)e^{j\omega_k(t-nT')}g(t-nT) $$

where the following apply: $g(t)$ is a rectangular pulse in time with unit amplitude and duration $T$; $T' = T + T_g$, where $T_g$ is the guard interval; $\Delta \omega = 2\pi \Delta f$, where $\Delta f = 1/T$ is the subcarrier spacing; $\omega_k = \omega_0 + k\Delta \omega$, $k = 0, \ldots, K-1$ are the $K$ subcarriers, where $\omega_0$ is the frequency of the lowest subcarrier; and $d_k(n)$ is the symbol of the $k$ subcarrier at the $n$ OFDM block. After including the synchronization preambles, the modulated passband signal is given by $s(t) = Re\{u(t)e^{j\omega_0 t}\}$.

At the receiver side, the data is demodulated after applying an FIR filter and resampling the signal in order to compensate for the Doppler effect (see Section 3.1). Two types of detection algorithms are implemented: coherent detection, which uses adaptive channel estimation and relies on temporal coherence between adjacent OFDM blocks [7], and differential detection, which does not rely on the retrieval of phase and channel estimates, nor on coherence between adjacent OFDM blocks. Instead, it relies on the assumption that the channel response is similar between adjacent subcarriers, as the differential encoding is performed in the frequency domain. In addition to the simplicity of implementation of this technique, a significant advantage is that it can deal with rapid channel changes or high motion, which would likely cause a coherent detector to lose track of the channel.

3. CONTRIBUTIONS

Our work on improving the system involves two parts: the design of an efficient Doppler compensation technique, implemented in MATLAB, and the evaluation of the error resilient coding techniques offered by the MPEG-4 standard.

3.1 Doppler Compensation Algorithm

The Doppler effect is defined as the change in the period of a signal when there is a relative speed between the transmitter and receiver. The scaling factor applied to the signal is given by $a = v_t/c$, where $v_t$ is the relative speed between the transmitter and the receiver and $c$ is the velocity of waves in the medium. In an underwater wireless acoustic system, the speed of waves is around 1,500 m/s, resulting in a Doppler coefficient that can reach non-negligible values above $10^{-4}$.

The Doppler effect in the underwater channel is modeled by introducing the Doppler scaling factor in the multipath channel response, as

$$ c(\tau, t) = \sum_{p} A_p(t)\delta(\tau - \tau_p + a_p t) $$

where $A_p(t)$ are the paths' amplitudes, $\tau_p$ are the paths' delays, and $a_p$ are the Doppler factors of each path.

In order to illustrate the Doppler effect on our system, we focus on a single OFDM block. Let us assume the following over Equation (2): (i) all the paths have a similar Doppler factor: $a_p \approx a$, $\forall p$, and (ii) the nominal Doppler factor $a$, the path delays $\tau_p$ and the path gains $A_p(t)$ are nearly constant over the OFDM block duration. Following these assumptions, the received signal in passband is given by

$$ r(t) = \text{Re}\left\{ \sum_k d_k e^{j\omega_k (t+\tau + a t)} \times \left[ \sum_p A_p g(t + \tau + a t + \tau_p) e^{-j\omega_k \tau_p} \right] + n(t) \right\} $$

where $n(t)$ denotes additive noise. Note that the received signal experiences two effects: (i) it is scaled in time, from $T$ to $T(1 + a)$, causing possible desynchronizations at the receiver, and (ii) each subcarrier experiences a frequency shift $e^{j\omega_k t}$, which depends on the frequency value of the subcarrier. These shifts cause strong Inter-Carrier Interference (ICI) if compensation is not performed prior to OFDM demodulation.

In order to compensate for the Doppler effect, we have implemented an algorithm based on [2] which utilizes the preambles of the transmitted signal, inserted before and after an arbitrary number of OFDM blocks (one OFDM frame, Fig. 2). The algorithm has two stages:

1. For each OFDM frame, and via synchronization with two adjacent preambles, the receiver estimates the time duration of one frame as $T_{rx}$. The time duration of this frame at the transmitter side is $T_{tx}$, which is a known value at the receiver. By comparing $T_{rx}$ with $T_{tx}$, the receiver infers how much the signal has been compressed or dilated by the channel:

$$ T_{rx} = T_{tx}(1 + \hat{a}) \quad \Rightarrow \quad \hat{a} = T_{rx}/T_{tx} - 1 $$

This requires a nearly constant speed between adjacent preambles. If a sudden change in speed occurs,
this algorithm will not be able to track the Doppler factor. Instead, the estimated Doppler factor will be the average along each OFDM frame, degrading the performance of the receiver. However, this should not be a big issue for typical AUV motion during inspection procedures, which are slow and steady.

2. The signal is resampled by the estimated factor \(1 + \hat{a}\):

\[
\tilde{z}(t) = \tilde{r}(t/(1 + \hat{a}))
\]

The goal is to make \(\hat{a}\) as close as possible to \(a\). If they are not exactly equal, there will remain a residual shift in each subcarrier, denoted by \(\epsilon = \frac{\hat{a} - a}{1 + \hat{a}}\). If coherent detection is utilized at the receiver, the residual frequency shift can affect the detection process unless it is compensated through dedicated Doppler tracking [6]. However, if differential detection is used, the residual frequency shift does not affect the detection process because it is similar between adjacent subcarriers, thus satisfying the assumption that the channel response does not vary drastically between consecutive subcarriers. Our new system matches the use of differential detection with an effective Doppler compensation scheme, making it more robust against motion variations in the temporal domain.

### 3.2 MPEG-4 Error Resilience Tools

In order to minimize the impact of transmission errors and increase the portion of corrupted bit streams that can be correctly decoded, the MPEG-4 standard offers a set of error-robust coding techniques. They are integrated with the core encoding algorithm and have minimal impact on the coding efficiency. Since they include a trade-off between robustness and efficiency, they are meant to be used together with, not in replacement of, conventional techniques such as channel coding. In this scenario, the overhead introduced by the heavy use of channel coding techniques is reduced by allowing them to leave residual errors in the data stream. The error resilience efficiency is maintained by allowing the error resilience tools to handle such residual errors. See [3] for further information.

The Packet-based Resynchronization and Data Partitioning are the specific tools that have been included in this system.

#### 3.2.1 Packet-Based Resynchronization

Packet-Based Resynchronization is used to re-establish synchronization between the decoder and the bit stream after an error has been detected in the decoding process. This is achieved by means of resynchronization markers, implemented as unique codes (i.e., a sequence of bits that cannot be emulated by any code used by the encoder). At the encoder side, these markers are inserted into the bit stream prior to transmission. When an error is detected at the receiver side, the decoder searches for the next resynchronization marker. Once it is found, synchronization is reestablished, allowing correct decoding of the remaining bits.

This resynchronization mode is usually enhanced by dividing each video frame into independent resynchronization packets in order to avoid error propagation from one packet to another (Fig. 3). In addition to inserting resynchronization markers at the beginning of each packet, the encoder removes all dependencies that exist from data belonging to consecutive video packets by inserting additional information in the header of each packet. This technique assures that even if one video packet is corrupted by transmission errors, all of the remaining packets can be decoded.

#### 3.2.2 Data Partitioning

After detecting an error in the bit stream and resynchronizing to the next resynchronization marker, the decoder has isolated the erroneous data between the two resynchronization markers. Typical video decoders discard all erroneous data because the motion, shape and texture information (which define a video frame) are coded together. Hence, when the decoder detects an error, regardless of whether the error occurred in the motion, shape or texture part, all the data in the video needs to be discarded.

MPEG-4 allows the user to combine Packet-Based Resynchronization with the Data Partitioning tool, which divides and rearranges the elements in each video packet into two groups, according to their importance in the decoding process. The most important syntax elements are the motion and shape information, which are placed in the first partition of the packet (Fig. 4). The second partition contains the texture information, which is considered to be less important for the decoding algorithm. These partitions are separated by a Motion Boundary Marker (MBM). If the first partition is recovered, it is possible for the decoder to reconstruct the whole packet, even if the second partition is damaged or lost due to transmission errors.

#### 3.2.3 Improvement Demonstration

In order to compare the visual quality of videos encoded with and without error resilience tools, we have chosen the DCT-based Video Quality Metric (VQM) [9] as a quantitative metric, as it exploits the property of human visual perception. The VQM indicates how much an image has been visually distorted: the higher the VQM, the higher the distortion and the worse the quality; the lower the VQM, the lower the distortion and the better the quality. In our study, we assume a maximum distortion of VQM=2, which proved to be sufficient to provide an acceptable visual quality.

The video used in our tests was taken from the National Oceanic and Atmospheric Administration (NOAA) website and contains a footage of a ROV (Remotely Operated Vehicle) mission during the discovery of the Titanic shipwreck. After the video signal is compressed and modulated, it is passed through a simulated multi-path time-invariant channel. The modulation parameters are the same ones used in the underwater experiments conducted in [5].

The main advantage of using error resilience tools in our...
responds to a Doppler coefficient of receivers in all the experiments is around 10 cm/s, which correspond to a relative speed of \( v_r = a \cdot c_{\text{water}} = 47 \text{ cm/s} \approx 1 \text{ knot} \).

Initially, we will consider that the system shows a correct performance if it meets the maximum thresholds for Mean Square Error (MSE) and Bit Error Rate (BER) set in [5], which are \( \text{MSE}_{\text{max}} = -5 \text{ dB} \) and \( \text{BER}_{\text{max}} = 10^{-3} \). However, our efforts demonstrate that these requirements can be relaxed when we utilize MPEG-4 error resilience tools.

### 4. EXPERIMENTAL RESULTS

The experimental tests were conducted using an in-air testbed, which was developed to enable efficient testing in an accessible environment, emulating underwater acoustic propagation more accurately than simple software simulations.

The deployed system at the transmitter side consists of a laptop, an audio interface and a speaker. The receiving elements are two microphones, which record the signals and send them to the receiver laptop through another audio interface. The received signals are combined using Maximal-Ratio Combining (MRC) [4].

The relative speed between the transmitter and the receivers in all the experiments is around 10 cm/s, which corresponds to a Doppler coefficient of \( a = 0.1/c_{\text{air}} = 3 \cdot 10^{-4} \).

#### 4.1 Experiment 1: Uniform Speed

In this experiment, the transmitter is moved with reasonably well controlled uniform speed towards the receivers, followed by deceleration and a final static period.

Without the Doppler compensation algorithm (Fig. 6), the average BER —indicated with a dashed line— remains around 0.5 when there is motion between the transmitter and the receiver. The MSE curve also shows high values during the motion period. As a result, the received video could not be decoded.

When the Doppler compensation algorithm is applied, the BER shows a high peak at block 15, caused by the short and fast motion applied to the transmitter when it was raised from its initial position. Since the Doppler factor varied during 1 ms (much less than the duration of one OFDM frame), the algorithm was not able to detect this short period of motion. However, the damaged data in the affected OFDM frame rendered just a few damaged video frames, without affecting the visual quality of the rest of the video. It must be also noted that the blocks going from 55 to 65 show a higher MSE than when the Doppler compensation is not applied. This demonstrates that occasionally the algorithm may not accurately identify the correct Doppler factor (in this case, \( a = 0 \)), leading to incorrect or unnecessary resampling.

The average BER and MSE along the rest of the signal remain around 9 \( 10^{-3} \) and \( -4 \text{ dB} \), respectively. As these values are above the maximum permitted thresholds, the video should present high levels of distortion or could even be undecodable. The simplest solution to improve these results would be to increase the amount of channel coding, which in turn would decrease the data rate of the system. However, even with such levels of BER, the video decoder was able to decode the video with almost no distortion (VQM=1.1).

This is explained by the use of MPEG-4 error resilience tools at the video decoder, which overcame those residual errors that the channel coding could not correct. This allows the system to operate with higher BER while still maintaining acceptable visual quality. The same experiment was conducted without applying MPEG-4 error resilience tools to the compressed video. As expected, the received video presented higher levels of visual distortion, quantified with a VQM of 3.2.

#### 4.2 Experiment 2: Changing the Speed

In this experiment, the transmitter is moved at uniform speed towards the receivers, followed by a decrease in velocity to a full stop. Next, the transmitter is moved at uniform speed in the opposite direction, away from the receivers.

When the Doppler compensation algorithm is not applied (Fig. 7), the system fails as in Experiment 1. As expected, the received video was rendered undecodable.

When we apply the Doppler compensation algorithm, the Doppler factor is correctly tracked and compensated for,
The testbed demonstrates the applicability of the presented approach to noisy wireless underwater channels, by providing the OFDM signal with an efficient motion compensation scheme and offering additional protection to the received video over error bursts. However, the Doppler compensation algorithm should be improved so that the Doppler factor is accurately calculated and we can avoid incorrect or unnecessary resampling.

Our experiments also demonstrate that the bit rate capacity supported by the system (10 kbps and higher depending on acoustic channel conditions) is large enough to handle the bit rates of the compressed video files (10 kbps). Future work will focus on transforming the system design into a real-time implementation.

As a companion to this short paper, a demonstration of video transmission over an in-air acoustic link will be submitted to WUWNET’11.

6. ACKNOWLEDGMENTS

This work was supported in part by the MIT Sea Grant College Program under NOAA Grants NA06OAR4170019 and NA10OAR4170086; the MIT SG Project Number 2009-R/RCM-26; the Chevron MIT Energy Initiative program, projects titled “Acoustic Communications for Ultra-Deepwater Oilfield Applications” (01/01/2010 - 12/31/2011) and “Development of a stationary single input/output communication station using OFDM Technology” (01/01/2011 - 12/31/2011); and ONR grants number N00014-10-1-0576 and N00014-07-1-0202. The authors would also like to thank Rameez Ahmed from Northeastern University (Boston, US) for conducting the in-air experimental tests.

7. REFERENCES


