Compressed-Sensing-Enabled Video Streaming for Wireless Multimedia Sensor Networks

Abstract:

This article presents the design of a networked system for joint compression, rate control and error correction of video over resource-constrained embedded devices based on the theory of compressed sensing. The objective of this work is to design a cross-layer system that jointly controls the video encoding rate, the transmission rate, and the channel coding rate to maximize the received video quality. First, compressed sensing based video encoding for transmission over wireless multimedia sensor networks (WMSNs) is studied. It is shown that compressed sensing can overcome many of the current problems of video over WMSNs, primarily encoder complexity and low resiliency to channel errors. A rate controller is then developed with the objective of maintaining fairness among video streams while maximizing the received video quality. It is shown that the rate of compressed sensed video can be predictably controlled by varying only the compressed sensing sampling rate. It is then shown that the developed rate controller can be interpreted as the iterative solution to a convex optimization problem representing the optimization of the rate allocation across the network. The error resiliency properties of compressed sensed images and videos are then studied, and an optimal error detection and correction scheme is presented for video transmission over lossy channels.
**Architecture:**

![Architecture Diagram]

**Existing System:**

In existing layered protocol stacks based on the IEEE 802.11 and 802.15.4 standards, frames are split into multiple packets. If even a single bit is flipped due to channel errors, after a cyclic redundancy check, the entire packet is dropped at a final or intermediate receiver. This can cause the video decoder to be unable to decode an independently coded (I) frame, thus leading to loss of the entire sequence of video frames.

**Disadvantages:**

Instead, ideally, when one bit is in error, the effect on the reconstructed video should be unperceivable, with minimal overhead. In addition, the perceived video quality should gracefully and proportionally degrade with decreasing channel quality.
Proposed System:

With the proposed controller, nodes adapt the rate of change of their transmitted video quality based on an estimate of the impact that a change in the transmission rate will have on the received video quality. While the proposed method is general, it works particularly well for security videos. In addition, all of these techniques require that the encoder has access to the entire video frame (or even multiple frames) before encoding the video.

Advantages:

The proposed CSV encoder is designed to: i) encode video at low complexity for the encoder; ii) take advantage of the temporal correlation between frames.

Modules:-

1. CS Video Encoder (CSV)

The CSV video encoder uses compressed sensing to encode video by exploiting the spatial and temporal redundancy within the individual frames and between adjacent frames, respectively.

- Sensing the channel: those that have the cost of sensing channel have higher energy consumption and so they are not suitable for WMSNs.
- Using extra packets: Using retransmission time of dropped packets includes not only retransmission request but also transmission of dropped packet. These methods waste a great amount of energy for congestion detection in sensor nodes.
- Low cost: Some methods do not necessitate extra cost for congestion detection. These methods are the most suitable for congestion detection in WMSNs.
2. Rate Change Aggressiveness Based on Video Quality:
With the proposed controller, nodes adapt the rate of change of their transmitted video quality based on an estimate of the impact that a change in the transmission rate will have on the received video quality. The rate controller uses the information about the estimated received video quality directly in the rate control decision. If the sending node estimates that the received video quality is high, and round trip time measurements indicate that current network congestion condition would allow a rate increase, the node will increase the rate less aggressively than a node estimating lower video quality and the same round trip time. Conversely, if a node is sending low quality video, it will gracefully decrease its data rate, even if the RT T indicates a congested network. This is obtained by basing the rate control decision on the marginal distortion factor, i.e., a measure of the effect of a rate change on video distortion.

3. Video Transmission Using Compressed Sensing:
We develop a video encoder based on compressed sensing. We show that, by using the difference between the CS Samples of two frames, we can capture and compress the frames based on the temporal correlation at low complexity without using motion vectors.

4. Adaptive Parity-Based Transmission:
For a fixed number of bits per frame, the perceptual quality of video streams can be further improved by dropping error samples that would contribute to image reconstruction with incorrect information. Which shows the reconstructed image quality both with and without including samples containing errors? It assume that the receiver knows which samples have errors, they demonstrate that there is a very large possible gain in received image quality if those samples containing errors can be removed.

We studied adaptive parity with compressed sensing for image transmission, where we showed that since the transmitted samples constitute an unstructured, random, incoherent combination of the original image pixels, in CS, unlike traditional wireless imaging systems, no individual sample is more important for image reconstruction than any other sample. Instead, the number of correctly received samples is the only main factor in determining the quality of the received image.
System Configuration:

Hardware Required:

✓ System : Pentium IV 2.4 GHz
✓ Hard Disk : 40 GB
✓ Floppy Drive : 1.44 MB
✓ Monitor : 15 VGA color
✓ Mouse : Logitech
✓ Keyboard : 110 Keys enhanced
✓ RAM : 512MB

Software Required:

✓ O/S : Windows XP
✓ Language : C#.Net