

Error Concealment and Post Processing for the Capsule Endoscope

Anna N. Kim^{1*} Eirik Jenssen Daling² Tor A. Ramstad² Ilangko Balasingham^{1,2}

¹Intervention Center

Oslo University Hospital and Institute of Clinical Medicine NO-0027, Oslo, Norway

²Dept. of Electronics and Telecommunications

Norwegian University of Science and Technology NO-7491, Trondheim, Norway

annak@ieee.org ramstad@iet.ntnu.no ilangkob@medisin.uio.no

ABSTRACT

We propose two receiver-end post processing schemes to improve the performance of capsule endoscope which utilizes in-body wireless communication. The first is tailored for our prior proposed source coder, and another has its roots in video processing and is not source coder dependent. Promising results are shown through simulation.

Categories and Subject Descriptors

H.4.3 [Information Systems Applications]: Communications Applications

Keywords

error detection, error concealment, inter-frame interpolation

1. BACKGROUND AND MOTIVATION

Ideally, digital transmission ought to be transparent. This means that bit stream entering the source decoder should contain as few errors as possible, if not none. To combat the errors the lossy wireless communication channels incur, the transmitter can incorporate for example error correction codes and set the transmit power sufficiently above the level of noise. Meanwhile, since wireless channels also often vary in time, designing the system for the worst case scenario can lead to unnecessary waste in limited resources. Adapting transmission to channel characteristics is one way of dealing with time varying channels. It however, may not be feasible for certain applications such as the capsule endoscope.

The capsule endoscope is a pill-sized in-body medical device which is designed to traverse the gastrointestinal (GI) tract. The miniature camera onboard the capsule takes photos along the way and transmits the images to on-body receivers wirelessly. The device is expected to operate 8 to

*This work was supported by the Research Council of Norway (NFR), under the project MELODY nr. 187857/S10

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...\$10.00.

10 hours and the available power is very much limited due to its restricted size. At the same time, the communication channel can vary both in time and in location, as the model presented in [2]. Power adaptation would require the capsule to also contain at least a simple receiver unit which leads to further expense in power; while error correction codes come at the cost of processing power and bit rate. Fortunately sources such as image and video are more tolerant to bit errors, compared to data files. In addition, contrary to the restriction on the transmitter side of the capsule endoscope, there is much more flexibility on the receiver side. Since there is no real-time playback requirement, the entire sequence of received images can be processed before analyzed by the medical personnel, without any concerns regarding complexity and power. In this paper, we propose a receiver based approach to combat bit errors resulted from transmission. Our contributions are two fold. For received images with limited bit errors, we detect and apply error concealment to improve the image quality. If the image frame contains too many bit errors to be concealed, we discard the frame and apply inter-frame interpolation for reconstruction.

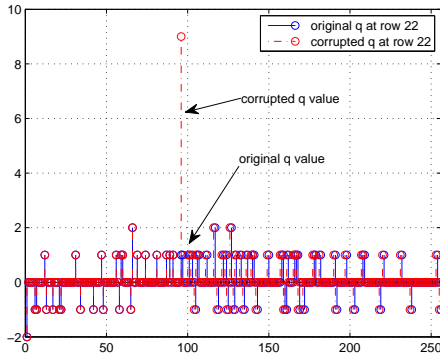
2. BIT ERRORS AND THEIR EFFECTS

The exact effect of transmission errors on reconstructed image is coder dependent. In [1], we proposed a low complexity, low bit rate image compression scheme for the capsule endoscope. The algorithm utilizes simple spatial domain decorrelation through prediction, combined with low-pass filtering and down-sampling to achieve high compression ratio at very good reconstruction quality. To further reduce the transmission rate, run-length coding is applied to the quantized prediction error, which contains consecutive zeros. So instead of encoding each zero level separately, the *number of zeros*, i.e. the run-length is encoded. The longer the run, the more savings in bit rate. Run-length coding is however susceptible to bit errors because the codeword length is not fixed. A single bit error may propagate and effect a string of decoded quantization levels. To improve robustness one can shorten the number of quantization levels to be encoded together, at the cost of higher bit rate. In addition, because of the inherent structure of predictive coding, the current reconstructed pixel value at the decoder relies on previously reconstructed pixels, error in one pixel value may also propagate.

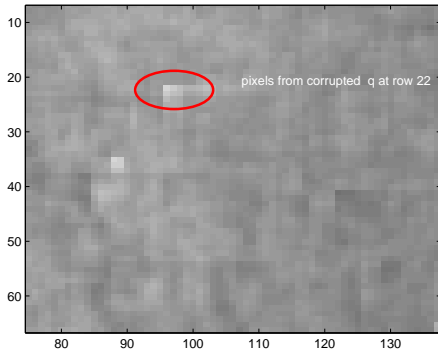
We begin our analysis by first analyzing the impact of bit

errors on the reconstructed image. The bit stream produced by the source coder is the output of the run-length coder. Here each codeword is either the number of consecutive zeros, or the value of the nonzero quantization level. Depending on the location of the bit error within the codeword, it has different impact at the decoder. This is best understood by inspecting the decoded quantization level, compared to the error-free original.

As shown in Figure 1(a), we see that a single bit error in the nonzero quantization level q resulted in a large difference in the decoded quantization level. This error in the reconstructed prediction error leads to a reconstructed pixel with a value that also differs greatly with its neighbours. Because of the following pixels are also reconstructed using the error pixel value, the effect of the bit error propagates in both horizontal and vertical direction, as shown in Figure 1(b). When the bit error is present in the *number* of consecutive



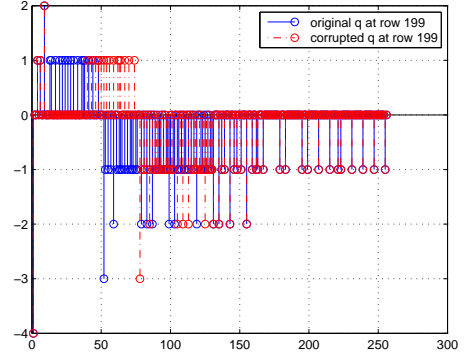
(a) Nonzero q value contains bit error



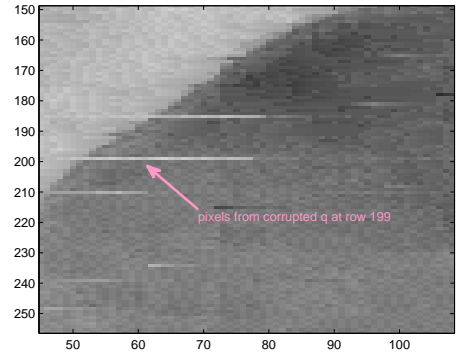
(b) Resulting reconstructed image

Figure 1: Effect of Bit Errors in Nonzero Value

zeros, its impact is somewhat different. Figure 2 gives one such example. We can see that because the number of zero run is wrong, the position of the nonzero quantization level is shifted. This error also propagates further down the other quantization levels. On the pixel level, the effect appear as a straight line which intensity does not taper off as quickly as the nonzero value errors, since error is enforced when the nonzero quantization level appears at the wrong location.



(a) Run-length of zero contains bit error



(b) Resulting reconstructed image

Figure 2: Effect of Bit Errors in Run-length

3. ERROR DETECTION AND CONCEALMENT

To identify the pixel location of the quantization level that contains bit errors, we start with detecting the 90° corners appearing on the reconstructed image. Note that all error concealment process takes places before the low-pass and upsampling step since error will get further smeared from interpolation.

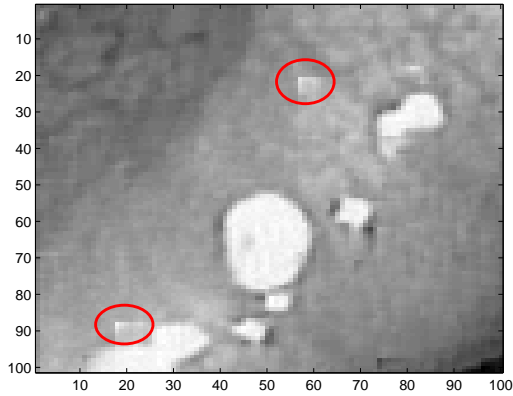
We apply a simple line detector to the image using kernel K_h^+ , where $^+$ indicates it is for detecting bright lines over dark background and $_h$ means the line is horizontal:

$$K_h^+ = \begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} \quad (1)$$

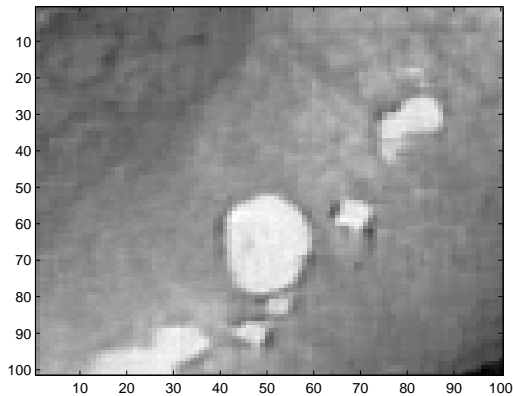
For darker lines against a bright background, noted by $K_{h,v}^-$, we can simply negate the kernel matrix. The vertical lines can be found by applying kernel $K_v^+ = \{K_h^+\}^T$, where T is the matrix transpose operator. The intersecting point of the horizontal and vertical lines is then the pixel where its corresponding q is erroneous. There is however, a simpler way of finding the pixel location. Because encoding is performed row-by-row, error propagation in the horizontal direction is a lot more prominent compared to the vertical direction. We can therefore only run the horizontal line de-

tector and determine the pixel location by finding the quantization level which has the maximum value of the line. This simplified strategy turned out to be very effective.

The error concealment process deals with the two types of bit errors separately. For bit errors in the nonzero quantization levels, we copy the quantization level from the previous neighbouring pixel (the one to the left). We then re-run the decoder with the error concealed quantization levels. The result is shown in Figure 3. We can see that both the ver-



(a) Luminance Channel Corrupted by Bit Errors



(b) After Error Concealment

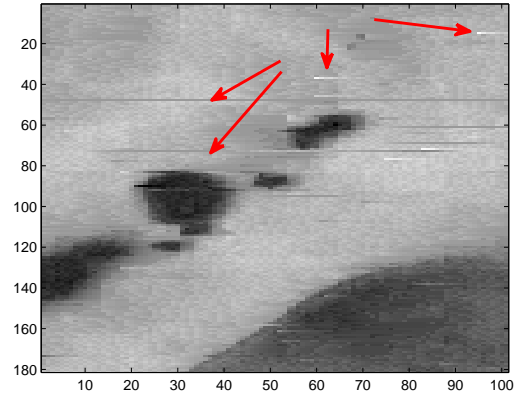
Figure 3: Quantization Level Concealment Example

tical and horizontal lines forming the square edge have disappeared after prediction with the concealed quantization level.

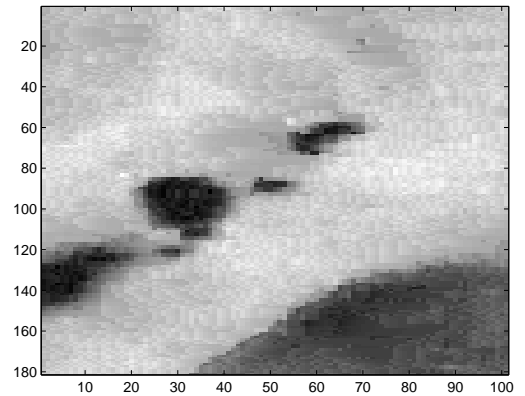
When bit errors are present in the runs of zeros, copying the neighbouring q is no longer sufficient, because replacing one value does not have effect on errors that propagated over many pixel locations. This type of bit error effect is often present in the chrominance components since they usually have longer runs of zeros.

Because the exact location of the bit error is not known, it is not possible to determine the actual correct number of zeros. For error concealment in this case, we work directly on the reconstructed image before upsampling. We replace the pixel segments of straight lines by copying the line above

or below. Note also that because the encoding process of the chrominance components only concerns pixels to its left, there is no propagation of errors in the vertical direction. Figure 4 gives an example of such concealment. The arrows in the top figure point to some of the corrupted pixel lines.



(a) Chrominance Channel Corrupted by Bit Errors



(b) After Error Concealment

Figure 4: Pixel Level Concealment Example

4. INTER-FRAME INTERPOLATION

When error concealment is no longer sufficient for obtaining a suitable image reconstruction quality, we apply inter-frame interpolation. This is done by exploiting the temporal correlation between the images frames, which is a common approach in processing video streams. The entire sequence of our received images can be seen as a video stream encoded frame-by-frame. In the rest of the paper, we use the term image and video interchangeably. Note that unlike that was described in the previous section, this approach is not tied to specific source coder. We briefly present the approach here and the interested reader is referred to [3] for further details. Denote the current frame to be interpolated as frame \hat{f}_n . Given the previous and future frames f_{n-1} and f_{n+1} , we first apply motion estimation to extract the motion vectors. This is followed by motion compensation. Both these pro-

cedures are bi-directional. For motion estimation, we apply so-called *block-matching motion estimation* BME [6], [4]. If we divide the video frame into macro blocks of certain size, we can use the *motion vector* to describe the displacement of these blocks in a frame before or after. Let $\hat{B}_{i,j}$ denote the i, j -th macro block of the current frame. For each block $\hat{B}_{i,j}$, we perform a bi-directional search within a specified window size N_w . Our goal is to find the best motion vector between the previous frame f_{n-1} and future frame f_{n+1} , passing through the block. Mathematically, we have [4]:

$$\mathbf{v}_{\hat{B}_{i,j}} = \arg \min_{\mathbf{v}} \sum_{\mathbf{s} \in \hat{B}_{i,j}} |f_{n-1}(\mathbf{s} - \mathbf{v}) - f_{n+1}(\mathbf{s} + \mathbf{v})| \quad (2)$$

where $\mathbf{v}_{\hat{B}_{i,j}}$ is the motion vector, and $\mathbf{s} = (i, j)$ is the coordinate for a pixel in the frame. To further improve the estimated motion vector field, we can correct erroneous motion vectors using vector median filtering [5].

To generate the actual interpolated frame, we apply bi-directional motion compensation:

$$\hat{f}_n(\mathbf{s}) = \frac{1}{2} \left(f_{n-1}(\mathbf{s} - \mathbf{v}_{\hat{B}_{i,j}}) + f_{n+1}(\mathbf{s} + \mathbf{v}_{\hat{B}_{i,j}}) \right) \quad (3)$$

Since block-based motion estimation and compensation often lead to blocking artifacts, one way to overcome the problem is to apply *overlapped block motion compensation* (OBMC) [4]. In OBMC, the pixel value of the current frame is generated, using not only motion vector of the block it belongs to but also taking into account the neighbouring blocks' motion vectors. The number of neighbouring blocks considered varies depends on implementation. In this paper we use the simple approach of considering only the three adjacent macro-blocks.

To evaluate the performance of the proposed scheme, we carry out the following experiment. We take the entire reconstructed image sequence at the decoder which have gone through the error detection and concealment process, and remove every second frame. We then perform inter-frame interpolation to reconstruct the missing frames using the method described above. For each interpolated frame, we calculate the composite PSNR (CPSNR) of each frame:

$$CPSNR = 10 \log_{10} \frac{3 * 255^2}{\sum_{k=1}^3 \sum_{i=1}^M \sum_{j=1}^N [R_{i,j,k} - I_{i,j,k}]^2}, \quad (4)$$

where R is the intentionally discarded reconstructed image frame, acting as reference, I is the corresponding interpolated frame, M, N are image dimensions and k represent the luminance and chrominance channels. The video sequences we use for the experiment are GI tract videos taken using *wired* endoscope, available from <http://gastrolab.net>. This is due to the fact that the current off-the-shelf capsule endoscopes can not deliver images of sufficient quality and frame rate. The wired endoscope video we use are 401×241 at 24 frames per second (fps). The block size and search window size are set to 32 pixels. We present the results in Figure 5. We can see that results from video sequence 1 is quite good. The CPSNR varies around 40dB which is generally regarded as high quality reconstruction. By comparison, inter-frame interpolation from sequence 2 has considerably lower CPSNR. This is due to the fact that sequence 1 has relatively slow motion, while sequence 2 contains much faster motion from the wired endoscope being pulled through the GI tract. Although the capsule endoscope in general moves

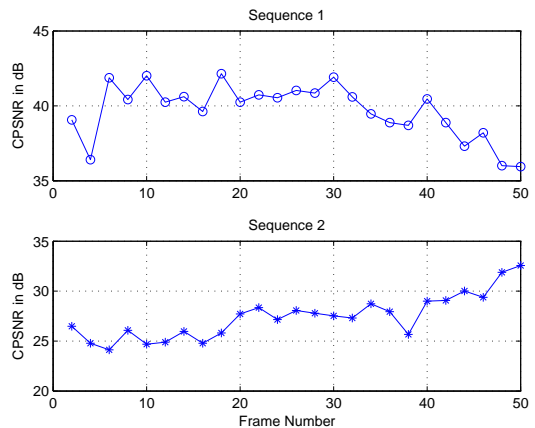


Figure 5: Quality of the Interpolated Frames

rather slowly, abrupt motion is still possible when the capsule is tumbling. High frame rate is therefore essential to ensure better estimation of motion vectors. Another source of distortion in sequence 2 is the reflection from the light source. Unlike motion, the light reflections depend on the inner structure and texture of GI tract. How to more accurately estimate the location of the reflection in the interpolated frame is a problem that remains to be solved.

5. CONCLUSIONS

Because of the restrictions in size and available power, it is often difficult to incorporate transmitter based schemes to improve robustness of the capsule endoscope. To circumvent the problem, two receiver based post processing algorithms are proposed in this paper. The first is a simple error detection and concealment method, which is designed for the source coder we proposed earlier. The second is a inter-frame interpolation scheme that uses temporal dependencies between the frames to construct a discarded frame. Simulation results showed that both approach are effective for the application considered.

6. REFERENCES

- [1] A. N. Kim, T. A. Ramstad and I. Balasingham, "Very Low Complexity Low Rate Image Coding for the Wireless Endoscope", *Proc. of ISABEL* 2011.
- [2] S. Støa, R. Chavez-Santiago, I. Balasingham, "An Ultra Wideband Communication Channel Model for the Human Abdominal Region", *Proc. of IEEE Globcom* 2010.
- [3] E. J. Daling, "Reduction of Power Consumption in Video Communication based on Low Frame Rate Transmission and Decoder Frame Interpolation". Master thesis, Dept. of Electronics and Telecomm. NTNU, June 2011.
- [4] B.-D. Choi, J.-W. Han, C.-S. kim and S.-J. Ko, "Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation", *IEEE Circ. & Sys. for Video Tech.* vol. 17, pp 407– 416, Apr. 2007.
- [5] J. Astola, P. Haavisto and Y. Neuvo, "Vector median filters", *Proc. of the IEEE* vol.78, pp 678–689, 1990.
- [6] D. Wang, A. Vicent, P. Blanchfield and R. Klepko, "Motion-compensated frame rate up-conversion – Part II: new algorithms for frame interpolation", *IEEE Trans. on Broadcasting*, vol 56, nr. 2, pp 142–149, 2010