Compression of Bayer based Endoscopic Images

In association with Cbyond
Keren Levinstein
Assaf Hallak
Supervisor: Rami Cohen
Winter 2010/2011
Problem Formulation

- Endoscopic images are taken by a miniature camera inside the human body
- Wireless transmission is desired
- Compression of the data is needed due to limited bandwidth of wireless channel
- Challenges:
  - Quality is crucial in medical imaging
  - Online compression is limited by energy and computational resources
  - The video is given in Bayer format:
    - each pixel contains information of only one color
    - Filter pattern is 50% green, 25% red and 25% blue
    - Full color pictures are obtained using de-mosaicing
De-mosaicing

- Digital image process used to reconstruct a full color image from a Bayer image
- De-mosaicing algorithms vary from closest neighbor interpolation to sophisticated adaptive algorithms
- The de-mosaicing algorithm used by Matlab (and Cbyond’s hardware):
  - Exploits the inter-channel correlation between the different colors channels
  - Uses simple linear operations and first order interpolation
  - A simple, low complexity algorithm with good performance

Bayer format image

De-mosaiced image
Data characteristics:

- Bayer format (8bpp)
- Current resolution 640x480, solution should hold for ~1 Megapixel
- 28.34 Frames per second
- Current bit rate is 66.416 Mbps
- Future planned bit rate is 226.72 Mbps
- Current system requirements – 60 Mbps

- Test set: 1 Minute sample movie of a simulated operation performed on a chicken

Assessment tools for the encoding algorithms
- Running time
- Bit rate
- Quality (YPSNR)
Popular encoders (designated for full color movies):

- **H.264:**
  - Best known encoder
  - Complicated encoding scheme
    - Uses motion estimation and integer DCT
    - Inter/Intra frame coding
    - Optimal encoding parameters are obtained using Lagrange multipliers
    - Due to its high complexity unusable in our case

- **M-JPEG:**
  - Encodes each frame using JPEG
  - Reasonable complexity
  - Temporal correlation is not exploited

- **JPEG-LS:**
  - Both lossless and near lossless modes
  - Efficient and has low complexity
  - Maximum allowed absolute error can be given as parameter
  - Temporal correlation is not exploited

- **We compare our suggested encoding methods to the encoders above**
  - Though they are not adjusted to Bayer format
Huffman Coding
• Prefix code
• Lossless
• Average word length close to optimal bound (entropy)
• Can be used in the end-point of any other compression

Huffman Coding, performance
• Running time:
  – Encoding time is linear in number of pixels
• Compression performance:
  – All colors together – 6.33 bpp, 52.62 Mbps
  – RGB separately – 6.29 bpp, 52.28 Mbps
  – Results for creating dictionaries online and offline are similar
• Quality: lossless
Block Truncation Coding

- The frame is divided into NxN blocks
- For each block:
  \[
  y(i, j) = \begin{cases} 
  1, & x(i, j) > \bar{x} \\
  0, & x(i, j) \leq \bar{x}
  \end{cases}
  \]
- The mean, variance and bitmap of each block are sent
- Decoding:
  \[
  a = \bar{x} - \sigma \sqrt{\frac{q}{m - q}}, \quad b = \bar{x} + \sigma \sqrt{\frac{m - q}{q}}
  \]
  Where \( m = N \times N \) and \( q = \# \) of pixels in block greater than the mean \( \bar{x} \)
  \[
  x(i, j) = \begin{cases} 
  a, & y(i, j) = 0 \\
  b, & y(i, j) = 1
  \end{cases}
  \]
- Total bits per block (8xNxN bits): 8 + 8 + NxN
  - Means, vars and bitmap can be Huffman encoded
- BTC is applied separately to each color
- Running time:
  - De/Encoding linear in number of pixels
Quantization

- Sample video histograms predicts good quantization results:
  - Quantizing only values greater than ~100 to minimize information loss
  - Max-Lloyd algorithm is used — converges to a local MSE minimum
  - Running time:
    - Offline: running time is irrelevant
    - Online: Max-Lloyd alg. can be stopped after a fixed time
    - De/Encoding linear in number of pixels
Intra differential coding
• Usually the subtraction of adjacent pixels results in data with lower entropy
• In Bayer format – subtraction is made for the same color

Inter differential frame coding
• Exploiting temporal correlation by subtracting successive frames
• Lower entropy data is obtained
• In order to prevent errors from affecting all future frames:
  – Choose each T frame to be a reset frame (Encoded as is)
  – Use DPCM

DPCM
• Motivation - Error in one pixel can drift in space (intra) or time (inter)
• A correction scheme is applied:
• Less efficient - the encoder has to contain a decoder
• Used in any lossy inter/intra encoder

Intra/Inter differential coding, performance
• Running time:
  – Huffman dictionaries built offline – to avoid sending the dictionary for each frame
  – De/Encoding linear in number of pixels
  – Inter requires one frame memory
• Disadvantage – unless DPCM is used, error in one pixel can drift in space (intra) or time (inter)
• Compression performance:
  – Intra – 3.15 bpp, 26.16 Mbps
  – Inter – 4.15 bpp, 34.44 Mbps
• Quality: lossless
Integer-DCT

- An efficient implementation of DCT
  - All arithmetic manipulations are done by multiplying/dividing by powers of 2
- The frame is divided into NxN blocks
- Each block is transformed to its corresponding spatial frequencies block
- Usually the energy is concentrated at the low frequencies coefficients
  - Quantizing the amplitude of high frequencies results in a lower entropy data
- The DC component has high inter/intra frame correlation
  - The difference can be encoded
- Integer DCT may perform better concatenated to the inter/intra-frame encoder

- Running time:
  - De/Encoding linear in number of pixels

- Compression performance:
  - For QP = 10 and normal frame coding – 2.15 bpp, 17.86 Mbps, YPSNR = 52.8 [dB]
  - Very similar results for inter/intra frame coding
**Rate control**

- An attempt to maximize picture quality vs. compression ratio trade off
- Each inter frame is divided into 8x8 blocks
- The variance is calculated for each block
- The 8x8 block is then encoded according to the following scheme:
  
  \[
  \sigma^2 < S_1 \quad \rightarrow \text{BTC encoding, 4x4 blocks, then Huffman} \\
  S_1 < \sigma^2 < S_2 \quad \rightarrow \text{BTC encoding, 2x2 blocks, then Huffman} \\
  \sigma^2 > S_2 \quad \rightarrow \text{Huffman encoding}
  \]

**Running time:**

The encoding technique for each block should be encoded as well

Linear in number of pixels

\[
S_1^2 = 3...20 \\
S_2^2 = 15...40
\]
Empirical results, summary

Conclusions:
- JPEG-LS shows the best results
- Combining several simple methods allow us to control the rate and quality of the compressed video to a satisfactory point
- Regular encoders produce good results for a Bayer video after color separation
- The desired compression rate and quality can easily be achieved with simple encoding methods