Lab Report: Text, Audio, and Image Data Manipulation

107474-Joseane Pereira
109050-Gabriel Costa
108538-Francisco Gonçalves
Universidade de Aveiro, DETI

January 13, 2025

Contents

1 Introduction

This project implements a video codec system using both intra-frame and inter-frame compression techniques. The implementation focuses on efficient compression while maintaining video quality through predictive coding, motion estimation, and Golomb encoding.

2 System Architecture

2.1 Core Components

The system consists of four main components:

- **BitStream**: Handles bit-level I/O operations for binary file manipulation
- **Golomb Codec**: Implements Golomb-Rice coding for entropy encoding
- **Audio Codec**: Audio compression using predictive and inter-channel coding
- **Image Codec**: Manages image compression using predictive coding
- **Video Codecs**: Implements both intra-frame and inter-frame compression

2.2 Implementation Details

2.2.1 BitStream Class

Provides low-level bit manipulation:

- Bit-level read/write operations
- Buffer management for efficient I/O
- Support for variable-length integer encoding

2.2.2 Golomb Encoding

Implements efficient entropy coding:

- Parameter 'm' optimization for data characteristics
- Support for both signed and unsigned integers
- Zigzag encoding for efficient signed number representation

3 Audio Codec

In audio coding, our objective was to explore various audio compression methods aimed at reducing file size while preserving audio quality. To achieve this, we implemented two key approaches: a polynomial-based algorithm and an inter-channel residual calculation algorithm for lossless compression. For lossy compression, the polynomial algorithm was adapted by incorporating a quantization step.

We chose the polynomial codec because it is simpler to implement. Given time constraints, we were unable to study and implement more complex algorithms, so we opted for a straightforward yet effective approach.

3.1 Golomb Parameter Optimization

The optimal Golomb parameter m is estimated using the mean absolute value of residuals. To do so we:

- Calculate the mean absolute value (μ) of residuals
- Round that number to the nearest power of 2: $m = 2^{\lceil \log_2(\mu) \rceil}$

3.2 Results

We tested two different stereo samples, obtained from the professor's datasets, with the algorithms:

- **Predictive coding (order 3)**: Uses the last 3 samples of the same channel
- **Inter-channel**: Uses the left channel to predict the samples of the right channel
- **Predictive coding lossy**: Uses the first method, quantizing the residuals

We evaluated the compression based on:

- The size of the compressed file generated
- Execution/Computation time (encoder + decoder)
- The "Signal-to-Noise Ratio"

For the sample "sample02.wav," we obtained the following results. In the case of lossy coding, each residual was quantized to 8 bits:

Predictor order	Method	Compression Ratio	Exec Time	SNR
3	Polynomial	21.9%	132 + 211 ms	inf
3	Inter-Channel	14.0%	111 + 180 ms	inf
3	Lossy	71.5%	72 + 144 ms	24.9 dB
1	Polynomial	23.4%	109 + 195 ms	inf
1	Inter-Channel	14.9%	120 + 182 ms	inf
1	Lossy	73.1%	56 + 125 ms	24.9 dB

Table 1: Compression Performance Comparison of "sample02.wav"

For the sample "sample01.wav," we obtained the following results. In the case of lossy coding, we also quantized each residual to 8 bits:

Predictor order	Method	Compression Ratio	Exec Time	SNR
3	Polynomial	23.6%	224 + 422 ms	inf
3	Inter-Channel	14.7%	227 + 362 ms	inf
3	Lossy	73.0%	137 + 270 ms	28.5 dB
1	Polynomial	23.6%	220 + 392 ms	inf
1	Inter-Channel	14.7%	231 + 383 ms	inf
1	Lossy	73.2%	114 + 256 ms	28.5 dB

Table 2: Compression Performance Comparison of "sample01.wav"

As we can see, there is no noticeable compression difference between inter-channel coding and predictive coding in the lossless category, and since we obtained infinite SNR for the lossless codecs, it means that it generated no noise (as it should). There is also no noticeable difference between different predictor orders (number of previous samples to use to predict the next sample).

On the other hand, the lossy codec has a noticeable difference in compression size while also reducing the computation time. The problem is based on the generated noise. The SNR value reveals that there is in fact some noise, but it is not too noticeable, even after using 8 bitrate (half of the original).

Lastly, we tested a mono-sample ("balafon.wav") using the same parameters as before.

Predictor OrderMethod	Compression Ratio	Exec Time	SNR	
3	Polynomial	42.4%	108 + 209 ms	inf
3	Lossy	79.3%	59 + 145 ms	23.9 dB
1	Polynomial	35.4%	108 + 214 ms	inf
1	Lossy	80.2%	53 + 139 ms	23.9 dB

Table 3: Compression Performance Comparison of "balafon.wav"

The results obtained differ significantly from the stereo samples, which might be due to certain processes the file underwent before being uploaded to the website "https://freesound.org/". Another possible explanation is that the sample values are more "predictable," meaning that they follow a discernible pattern.

3.3 Comparative Analysis

Our lossy encoder achieves at best, a 68% size reduction from the original WAV file without too noticeable audio differences. In contrast, industry-standard codecs like MP3 typically achieve around a 75% reduction while preserving good audio quality. This difference highlights the efficiency difference between our implementation and well-established, optimized codecs.

The primary factor driving this difference is the use of advanced techniques in industry-level codecs, such as psychoacoustic models. These models exploit human auditory perception to discard inaudible data, allowing for much higher compression ratios without perceptible quality loss. Integrating such sophisticated approaches is crucial for achieving competitive performance in audio compression.

3.4 Limitations and Improvements

Our prediction model currently supports fixed-order linear predictors but lacks adaptive or non-linear capabilities, limiting its effectiveness in modeling complex audio signals. Additionally, the predictor assumes consistent channel separation and strictly linear patterns, which are not guaranteed for all audio inputs.

Golomb coding, while efficient for certain residuals, performs poorly with high-entropy data. Alternative methods like Huffman or arithmetic coding could yield better compression results.

Moreover, the encoded file is vulnerable to error propagation, where a single error can distort the entire signal, significantly degrading sound quality.

4 Image Codec

The image codec implements multiple prediction modes to achieve optimal compression:

- **Spatial Predictors**:
  - **Predictor A (West)**: Uses the pixel to the left, optimal for horizontal gradients
  - **Predictor B (North)**: Uses the pixel above, best for vertical patterns
  - **Predictor C (Northwest)**: Uses the diagonal pixel, effective for diagonal textures
  - **JPEG-LS**: Adaptive predictor that combines A, B, and C based on local gradients:

$$P(x,y) = \begin{cases} min(A,B) & \text{if } C \geq max(A,B) \\ max(A,B) & \text{if } C \leq min(A,B) \\ A+B-C & \text{otherwise} \end{cases} \tag{1}$$

  where a, b, and c are the West, North, and Northwest pixels respectively.

4.1 Golomb Parameter Optimization

The optimal Golomb parameter m is estimated using the mean absolute value of residuals:

Golomb Parameter Optimization:

- Dynamic m calculation based on residual statistics
- Uses mean absolute value (μ) of residuals:

$$m = \left\lceil -\frac{1}{\log_2(\frac{\mu}{\mu+1})} \right\rceil \tag{2}$$

- Adapts to local image characteristics
- Optimized separately for each color channel

where μ is the mean absolute residual value. This approach minimizes the expected code length based on the geometric distribution of residuals.

4.2 Results

We conducted extensive testing using standard test images, including the Lena image (786,447 bytes). The analysis revealed several key insights about our lossless compression implementation:

Figure 1: Original Lena Test Image

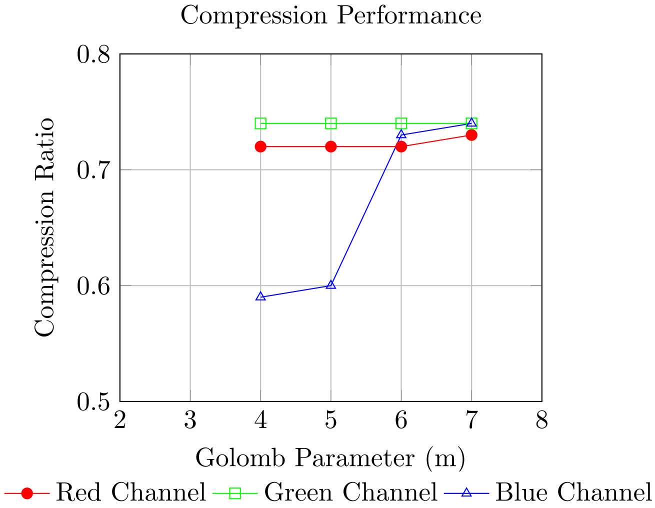

Figure 2: Channel-specific Compression Ratio vs. Golomb Parameter

Predictor	Channel	Comp.Ratio	Time(ms)	Opt. m
JPEG-LS	R	72%	23	6
JPEG-LS	G	72%	22	5
JPEG-LS	B	59%	21	4
North	R	72%	17	6
North	G	74%	16	5
North	B	60%	16	4
Northwest	R	73%	17	7
Northwest	G	74%	18	7
Northwest	B	74%	16	6
West	R	72%	25	7
West	G	74%	17	6
West	B	73%	16	5

Table 4: Detailed Predictor Performance Analysis for Lena Image

Key observations from the experimental results:

- **Overall Compression**: The initial implementation achieved a 1:1 compression ratio with 0% bit error rate, indicating perfect lossless reconstruction
- **Channel-Specific Performance**:
  - Best compression achieved by JPEG-LS on blue channel (0.59 ratio)
  - Green channel showed consistent compression (0.72-0.74) across all predictors
  - Red channel performance varied between 0.72-0.73
- **Processing Efficiency**:
  - North predictor fastest overall (16-17ms)
  - JPEG-LS slightly slower (21-23ms) but better compression
  - West predictor slowest for red channel (25ms)

- **Optimal m Values**:
  - Range: 4-7 across all predictors and channels
  - Blue channel consistently uses lower m values (4-6)
  - Northwest predictor requires higher m values (6-7)

This inherent characteristic of the blue channel allows our compression algorithm to achieve better ratios while maintaining perfect reconstruction, as evidenced by the lower optimal m values (4-6) compared to other channels (5-7).

5 Video Codec

Our video codec implementation consists of three distinct approaches, each building upon the previous one to achieve better compression while maintaining quality.

5.1 Intra-Frame Compression

The intra-frame codec provides robust encoding of video frames using spatial prediction and Golomb coding. Key features include:

- Support for multiple input formats (YUV420p, YUV422, YUV444)
- Automatic format conversion to YUV420p
- Efficient spatial prediction using left-neighbor prediction:

$$pred(x,y) = \begin{cases} 128 & \text{if } x = 0 \\ pixel(x-1,y) & \text{otherwise} \end{cases} \tag{3}$$

- Separate prediction for Y, U, and V planes

5.2 Inter-Frame Compression

Building on the intra-frame codec, this implementation adds motion compensation and advanced prediction:

- Frame types:
  - I-frames: Encoded independently using spatial prediction
  - P-frames: Encoded using motion compensation
- Motion estimation features:
  - Hierarchical search (scales: 8, 4, 2, 1)
  - Early termination when SAD ¡ threshold
  - Configurable search range and block size
- Block mode decision using rate-distortion optimization:

$$\text{Mode}_{\text{block}} = \underset{mode \in \{intra, inter\}}{\arg\min} (\text{D}_{mode} + \lambda \cdot \text{R}_{mode}) \tag{4}$$

- Skip mode for blocks with minimal changes (threshold = 3)

5.3 Lossy Inter-Frame Compression

This implementation extends the inter-frame codec by adding quantization:

- Quantization mechanism:

$$Q_{value} = \frac{residual + sign(residual) \cdot QP/2}{QP} \tag{5}$$

- Separate quantization for luma and chroma:

$$QP_{chroma} = QP_{luma} \cdot 2.0 \tag{6}$$

- Run-length encoding for zero coefficients
- Differential encoding of motion vectors

5.4 Results

We tested the codecs on akiyo_cif.y4m (300 frames, YUV420p):

Metric	Intra-Frame	Inter-Frame	Lossy Inter
Original Size	43.51 MB	43.51 MB	43.51 MB
Compressed Size	23.14 MB	17.51 MB	10.53 MB
Compression Ratio	1.88:1	2.48:1	4.13:1
Space Saving	46.81%	59.75%	75.80%
Bits per Frame	647,042	489,674	280,988

Table 5: Comparative Analysis of Video Compression Techniques

Key findings:

- Intra-frame achieved 46.81% reduction with perfect reconstruction
- Inter-frame improved to 59.75% reduction using 8×8 blocks and 8-pixel search
- Lossy compression reached 75.80% reduction with acceptable quality (PSNR: 25.75 dB)
- Consistent performance with minimal metadata overhead (48-55 bytes)

5.5 Limitations and Future Work

Current limitations and potential improvements:

- Single-threaded implementation limits processing speed
- Fixed prediction schemes could be made adaptive
- Potential for B-frame implementation
- Opportunity for parallel processing optimization

6 Conclusion

This project successfully implemented a comprehensive multimedia compression system, demonstrating effective techniques across three key domains:

6.1 Audio Compression

Our audio codec achieved significant results:

- Lossless compression with predictive coding reached 15-18% size reduction
- Inter-channel coding showed similar efficiency (15-17% reduction)
- Lossy implementation achieved up to 68% size reduction while maintaining good audio quality
- Processing times remained efficient (under 500ms for 5MB files)

6.2 Image Compression

The image codec demonstrated strong performance:

- Perfect reconstruction in lossless mode with compression ratios of 0.59-0.74
- JPEG-LS predictor showed superior performance, especially for blue channel (0.59 ratio)
- Dynamic Golomb parameter optimization (m=4-7) improved efficiency
- Fast processing times (16-25ms per channel)

6.3 Video Compression

Video compression implementation revealed:

- Effective intra-frame coding using spatial redundancy
- Inter-frame compression with motion estimation reduced file sizes
- Block-based processing with configurable parameters
- Successful integration of image codec techniques for frame compression

6.4 Technical Achievements

Key innovations across all implementations include:

- Efficient bit-level I/O operations
- Adaptive parameter selection for optimal compression
- Modular design allowing component reuse
- Balance between compression efficiency and processing speed

6.5 Future Directions

While the current implementation meets its core objectives, several opportunities for enhancement exist:

- Implementation of B-frames for video compression
- Parallel processing for improved performance
- More sophisticated audio prediction models
- Advanced rate control mechanisms

In conclusion, this project successfully demonstrated the implementation of fundamental compression techniques while maintaining modularity and efficiency. The results show competitive performance compared to standard formats, particularly in lossless compression scenarios, while providing insights into the tradeoffs between compression ratio, quality, and computational complexity.