Homework #3 – Motion Estimation & Compensation

314552036 周子翔

Objective

The goal of this homework is to implement and analyze motion estimation (ME) and motion compensation (MC) algorithms for video compression. Specifically:

1. Implement full search block matching for motion estimation with different search ranges ([±8], [±16], [±32]).
2. Implement motion compensation to reconstruct the frame and calculate residuals.
3. Compare the results (PSNR and runtime) for different search ranges.
4. Implement a three-step search algorithm and compare its results with the full search algorithm.

Implementation Details

1. Motion Estimation (ME)

- **Algorithm**: Full Search Block Matching
- **Parameters**:
    - **Block Size**: 8x8
    - **Search Range**: [±8], [±16], [±32]
- **Method**:
    - Divide the frame into non-overlapping blocks of size 8x8.
    - For each block in the current frame, search for the best matching block in the reference frame within the specified search range.
    - Use Mean Squared Error (MSE) as the matching criterion.
- **Output**:
    - **Motion Vectors**: Displacement values (dx, dy) for each block.
    - **Residuals**: Difference between the original block and the reconstructed block.

2. Motion Compensation (MC)

- **Purpose**: Reconstruct the current frame using motion vectors and the reference frame.
- **Method**:
    - Use motion vectors to fetch the best matching blocks from the reference frame.
    - Combine these blocks to reconstruct the current frame.
- **Output**:
    - **Reconstructed Frame**: Approximated version of the current frame.

3. PSNR Evaluation

- **Purpose**: Measure the quality of the reconstructed frame.
- **Method**:
    - Calculate the Peak Signal-to-Noise Ratio (PSNR) between the original frame and the reconstructed frame.
- **Output**: PSNR value in dB.

4. Runtime Comparison

- **Purpose**: Compare the computational efficiency of different search ranges.
- **Method**:
    - Measure the execution time for motion estimation and motion compensation using Python's time module.
- **Output**: Runtime statistics.

Results

```
Search Range: 8
Full Search Time: 1.5155 seconds
PSNR: 23.15 dB
Search Range: 16
Full Search Time: 5.8053 seconds
PSNR: 25.92 dB
Search Range: 32
Full Search Time: 20.9858 seconds
PSNR: 30.25 dB
```

1. search range 8:

2. search range 16:

3. search range 31:

Conclusion

1. **Search Range**:
    - Increasing the search range improves PSNR but increases runtime significantly.
    - A balance between search range and runtime must be considered for practical applications.
2. **Three-Step Search**:
    - Three-step search is faster than full search but may result in slightly lower PSNR.
    - It is suitable for scenarios where computational efficiency is critical.
3. **Residuals**:
    - Residuals highlight the areas where motion compensation fails to perfectly reconstruct the frame.