21 - S1 - Q1

Q: (a) A row-column decomposition method $\longrightarrow$?

Solution: 不会

(b) 让帧某些宏块坏，发运动向量修，

draw diagram

$\boxed{\text{Motion estimation}}$

$\downarrow$

$\boxed{\text{Motion Vectore}}$

$\downarrow$

$\boxed{\text{Motion Compensation}}$

$\downarrow$

$\boxed{\text{I - Frame}}$

思考 74 秒 CHATGPT o1

Answer to 1(a):

To compute the two-dimensional transform of the given image block using the transform matrix 'A' via the row-column decomposition method, we'll follow these steps:

1. **Compute $Y = A \times X$:**

   First, multiply the transform matrix A with the image block X.

   Given:

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 \\ 1 & -1 & -1 & 1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} \end{bmatrix}$$

$$X = \begin{bmatrix} 20 & 20 & 20 & 20 \\ 20 & 20 & 20 & 20 \\ 20 & 35 & 35 & 35 \\ 20 & 35 & 35 & 35 \end{bmatrix}$$

   Compute $Y = A \times X$:

$$Y = \begin{bmatrix} 80 & 110 & 110 & 110 \\ 0 & -22.5 & -22.5 & -22.5 \\ 0 & 0 & 0 & 0 \\ 0 & 7.5 & 7.5 & 7.5 \end{bmatrix}$$

   This is computed by performing matrix multiplication, where each element of Y is calculated as $Y_{i,j} = \sum_{k=1}^{4} A_{i,k} \times X_{k,j}$.

2. **Compute $T = Y \times A^T$:**

   Next, multiply the result Y with the transpose of the transform matrix A^T.

   Compute $T = Y \times A^T$:

$$T = \begin{bmatrix} 410 & -30 & -30 & -15 \\ -67.5 & 22.5 & 22.5 & 11.25 \\ 0 & 0 & 0 & 0 \\ 22.5 & -7.5 & -7.5 & -3.75 \end{bmatrix}$$

   This is calculated by multiplying Y and A^T where each element of T is computed as $T_{i,j} = \sum_{k=1}^{4} Y_{i,k} \times A_{j,k}$.

Final Transformed Matrix T:

$$\begin{bmatrix} 410 & -30 & -30 & -15 \\ -67.5 & 22.5 & 22.5 & 11.25 \\ 0 & 0 & 0 & 0 \\ 22.5 & -7.5 & -7.5 & -3.75 \end{bmatrix}$$

Answer to 1(b):

In MPEG-2 encoding, motion vectors can be included with I-frames to assist in error recovery when macroblocks are damaged during transmission. Here's how motion vectors are obtained in an I-frame:

1. **Motion Estimation on I-Frame:**
   - **Input I-Frame:** The current I-frame is input into the encoder.
   - **Reference Frames:** Previous frames (which could be earlier I-frames or P-frames) are stored in the reference frame buffer.
   - **Motion Estimation Unit:** The encoder performs motion estimation by comparing macroblocks of the current I-frame with those in the reference frames to find similar blocks.
   - **Calculation of Motion Vectors:** For each macroblock, the displacement between the matching blocks is calculated, resulting in motion vectors.
2. **Inclusion of Motion Vectors:**
   - The motion vectors are encoded and included in the bitstream along with the I-frame data.
   - These vectors provide additional information that can be used at the decoder to reconstruct damaged macroblocks using data from reference frames.
3. **Diagram of the Process:**

   Explanation of Functional Blocks:
   - **Input I-Frame:** The current frame being encoded.
   - **Reference Frame Buffer:** Stores previous frames used for motion estimation.
   - **Motion Estimation Unit:** Compares macroblocks from the input I-frame with those in the reference frames to find motion vectors.
   - **Motion Vectors:** The output from the motion estimation unit, representing the movement of macroblocks.
   - **DCT and Quantization:** The I-frame undergoes DCT transformation and quantization for compression.
   - **Entropy Encoder:** Encodes the quantized coefficients and motion vectors into a compressed bitstream.
   - **Output Bitstream:** The final encoded data sent over the transmission channel.
4. **Use of Motion Vectors for Error Recovery:**
   - If a macroblock in the I-frame is damaged, the decoder uses the motion vectors to locate the corresponding macroblock in the reference frame.
   - The missing or damaged macroblock is reconstructed using the data from the reference frame, guided by the motion vector.

Summary:

By including motion vectors with I-frames, the MPEG-2 encoder enhances error resilience. The motion vectors allow the decoder to effectively recover and reconstruct damaged macroblocks, ensuring better video quality even in the presence of transmission errors.

CHATGPT 4o

1. (a) Two-Dimensional Transform Using Row-Column Decomposition Method

The matrix A provided is a 4x4 transformation matrix:

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/2 & -1/2 & -1 \\ 1 & -1 & -1 & 1 \\ 1/2 & -1 & 1 & -1/2 \end{bmatrix}$$

The image block given in Figure 2 is:

$$I = \begin{bmatrix} 20 & 20 & 20 & 20 \\ 20 & 20 & 20 & 20 \\ 20 & 35 & 35 & 35 \\ 20 & 35 & 35 & 35 \end{bmatrix}$$

To compute the 2D transform, we use the row-column decomposition method. The 2D transform of matrix I is computed by applying the 1D transform A first to the rows, and then applying the transform to the columns of the intermediate result.

Step 1: Apply A to the rows of I.

We perform matrix multiplication for each row of I with matrix A. Let's compute the transformed matrix after applying A to the rows.

$$T_{\text{rows}} = A \cdot I$$

Matrix multiplication is carried out row-wise:

1. First row transformation:

$$[1, 1, 1, 1] \cdot [20, 20, 20, 20] = 1(20) + 1(20) + 1(20) + 1(20) = 80$$

2. Second row transformation:

$$[1, 1/2, -1/2, -1] \cdot [20, 20, 20, 20] = 1(20) + 1/2(20) - 1/2(20) - 1(20) = 0$$

3. Third row transformation:

$$[1, -1, -1, 1] \cdot [20, 20, 20, 20] = 1(20) - 1(20) - 1(20) + 1(20) = 0$$

4. Fourth row transformation:

$$[1/2, -1, 1, -1/2] \cdot [20, 20, 20, 20] = 1/2(20) - 1(20) + 1(20) - 1/2(20) = 0$$

Thus, the first row of the transformed matrix becomes: $[80, 0, 0, 0]$.

Repeating the process for the remaining rows of the image block I:

1. Applying the first row of A to the second row of I:

$$[1, 1, 1, 1] \cdot [20, 20, 35, 35] = 1(20) + 1(20) + 1(35) + 1(35) = 110$$

2. Applying the second row of A to the second row of I:

$$[1, 1/2, -1/2, -1] \cdot [20, 20, 35, 35] = 1(20) + 1/2(20) - 1/2(35) - 1(35) = -12.5$$

3. Applying the third row of A to the second row of I:

$$[1, -1, -1, 1] \cdot [20, 20, 35, 35] = 1(20) - 1(20) - 1(35) + 1(35) = 0$$

4. Applying the fourth row of A to the second row of I:

$$[1/2, -1, 1, -1/2] \cdot [20, 20, 35, 35] = 1/2(20) - 1(20) + 1(35) - 1/2(35) = 2.5$$

Thus, the second row of the transformed matrix becomes: $[110, -12.5, 0, 2.5]$.

Proceeding similarly for the remaining rows, we get the intermediate transformed matrix after row transformations.

$$T_{\text{rows}} = \begin{bmatrix} 80 & 0 & 0 & 0 \\ 110 & -12.5 & 0 & 2.5 \\ 110 & -12.5 & 0 & 2.5 \\ 80 & 0 & 0 & 0 \end{bmatrix}$$

Step 2: Apply A to the columns of T_{rows}.

We now multiply matrix T_{rows} with the transpose of matrix A.

This step follows a similar matrix multiplication process, which leads to the final 2D transformed matrix.

---

1. (b) Diagram and Explanation for Motion Vector Estimation in I-Frame

In MPEG-2 encoding, motion vectors are typically used for predicting macroblocks in P-frames and B-frames. However, in cases where certain macroblocks in an I-frame (intra-frame) are damaged during transmission, motion vectors can be used to retrieve the missing details by exploiting the spatial redundancy between macroblocks in adjacent I-frames.

The process for obtaining motion vectors in an I-frame involves several key steps:

1. **Motion Estimation Block**: This block identifies the displacement of blocks between successive frames. For I-frames, the damaged macroblocks are compared with neighboring macroblocks or from adjacent frames to estimate motion vectors.
2. **Motion Compensation Block**: Once the motion vectors are obtained, they are used to compensate for the missing blocks by filling in the damaged macroblocks with information from nearby blocks or previous frames.
3. **Residual Coding Block**: Any remaining difference between the compensated blocks and the actual data is encoded using transform coding techniques, such as DCT.

The functional blocks diagram:


```lua
+-------------------------+
| Motion Estimation       |
+-------------------------+
         |
         v
+-------------------------+     +-------------------------+
| Motion Vectors          | -->| Motion Compensation     |
+-------------------------+     +-------------------------+
                                    |
                                    v
                              +------------------+
                              | I-Frame          |
                              +------------------+
```

Motion estimation compares the I-frame's blocks with reference data to generate motion vectors, which are then used in motion compensation to correct damaged macroblocks, thus maintaining image quality during transmission errors.