# Low Complexity Video Compression for Fixed Focus Cameras

Kumar Manas, Mohit Jindal, and Preety Singh
The LNM Institute of Information Technology
Jaipur, Rajasthan, 302031, India
{kumarmanas.y17,mohitjindal.y17,preety}@lnmiit.ac.in

# Motivation

- Every new version of video coding algorithms uses more advanced algorithms to increase compression efficiency.  Hence computational complexity increases too.
- Due to the increasing cost of the CCTV cameras organisations are reluctant to install them in good numbers.

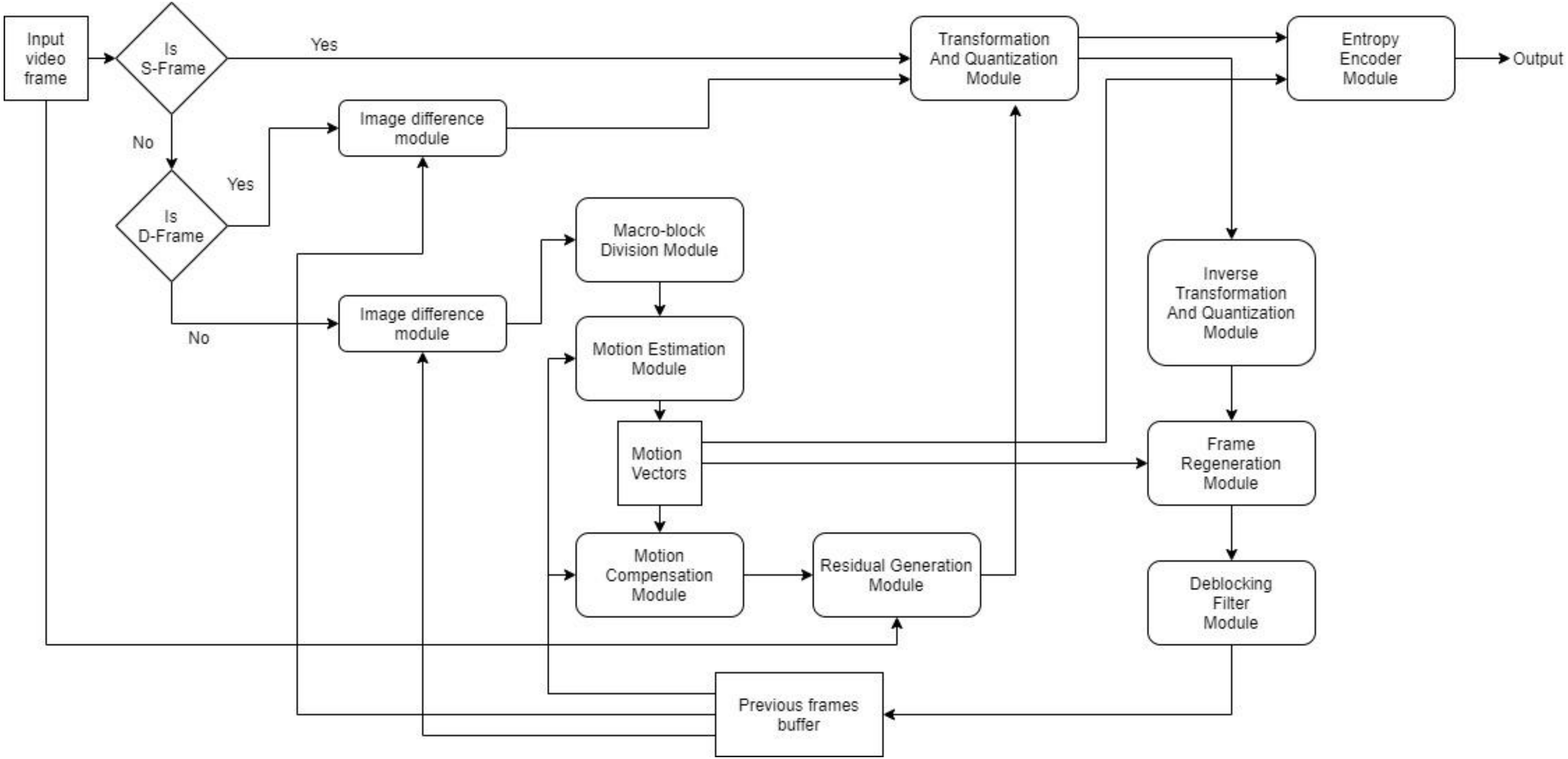
# Terminology

1. **S-frame**(Stable Frame) - Every 120th frame is an S-Frame. Encoded without prediction. Only Integer Transform and entropy encoding are performed for these frames. The interval of 120 was chosen after several experiments.
2. **D-frame**(Difference frame) - Every 10th frame is a D-Frame. It is not predicted, just a difference between this frame and the most recent S-Frame is encoded to compress the frame.
3. **P-frame**(Predicted frame) - These frames occur between two consecutive D-Frames. These are stored relative to the frame just preceding it.

# Proposed Methodology

- In the case of fixed focus cameras, difference in frames is only due to motion of objects in the frame.
- Exploiting this property, we have introduced two new modules in addition to the existing modules of the current coding standards. The two modules are:
  a. Image Difference Module (IDM)
  b. Macroblock Division Module (MDM)
- Also, we have introduced some changes in the Motion Estimation Module (MEM)

# Structure of Encoder



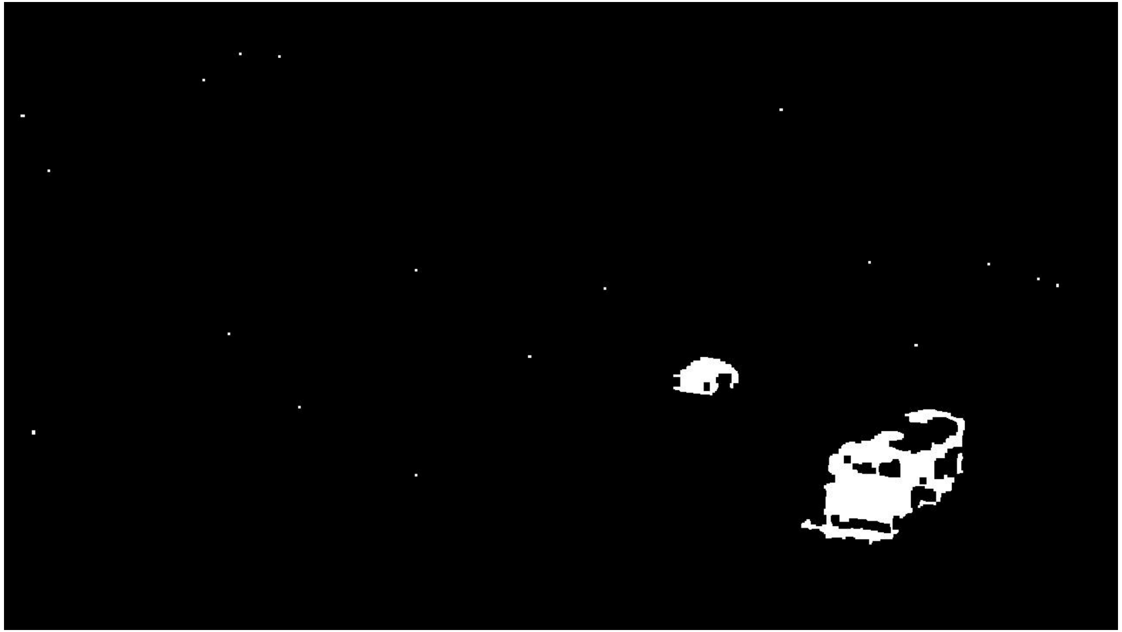
# Image Difference Module(IDM)

- The module performs pixel-to-pixel comparison of current frame and reference frame (in all three channels Y, Cb and Cr), marking the changed positions with value '1', else value '0', in a difference matrix M1.

# 2 consequent frames

Slight movement in the yellow bus and blue car

# Output of Image Difference Module



# MacroBlock Division Module(MDM)

- Values from M1 are used to predict the macroblock mode for each 16X16 macroblock in the frame.
- Every macroblock in M1 is visited with a 4X4 window.
- Depending on these values, that particular 4X4 block gets assigned a value `0' or `1'  in an intermediate matrix M2.
- Every 2X2 block of M2 is visited and sum of all its cells is stored in a variable **Total** and a new matrix M3 is created.
- Decision to merge sub-blocks is done on basis of **Total** and M3 values.

**We have come up with some codewords** which will help us in encoding the information of macroblocks and their sub blocks in lesser space

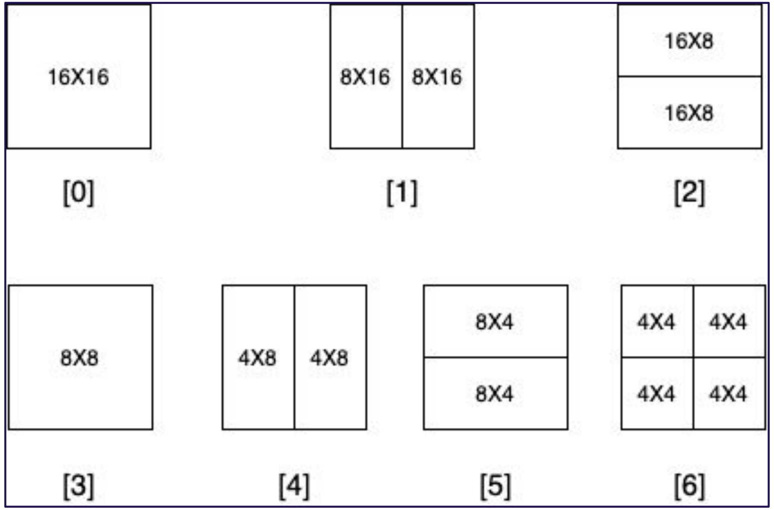



**Fig: Codewords**

| Codeword | Block Dimensions | Blocks count |
|---|---|---|
| 0 | 16x16 | 1 |
| 1 | 8x16 | 2 |
| 2 | 16x8 | 2 |
| 3 | 8x8 | 4 |
| 4 | 4x8 | 2 |
| 5 | 8x4 | 2 |
| 6 | 4x4 | 4 |

**Table: Code-words for different block**

Let’s take a 16x16 Macroblock from the Image Difference Matrix, to understand better

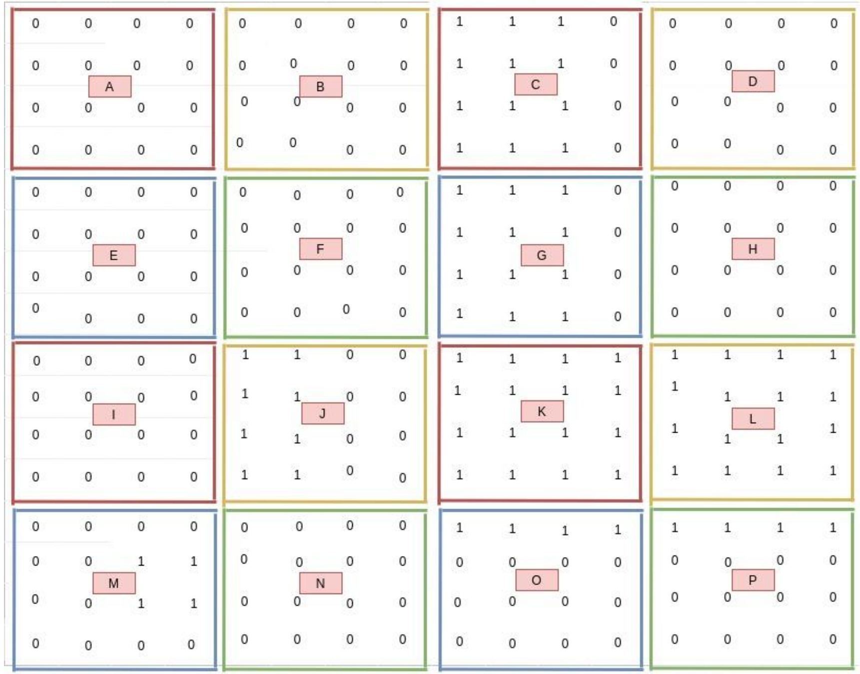



**Fig: A 16x16 macroblock of Image Difference Matrix M1**

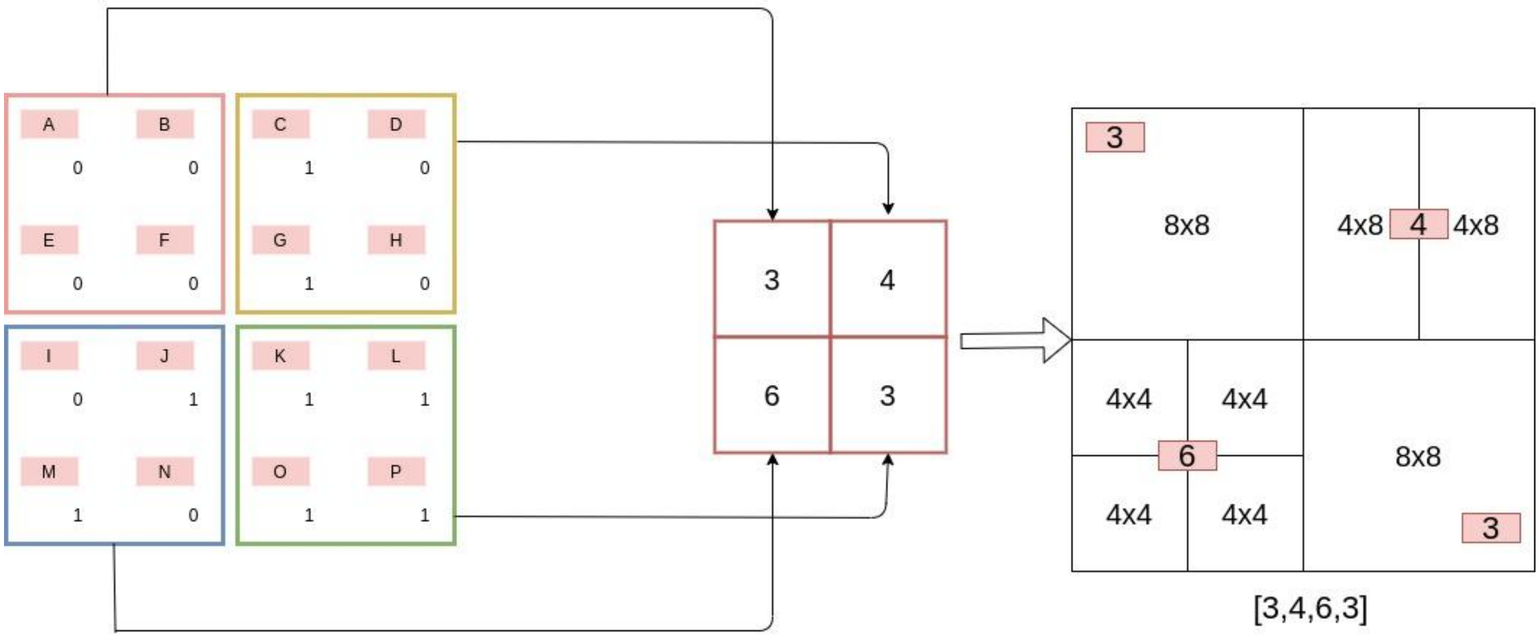



Fig: Matrix M2 (Left) Matrix M3 (Center) and Macroblock Modes (Right)

# Motion Estimation Module(MEM)

- It uses the macroblock modes list obtained from MDM and determines motion vectors for all macroblocks.
- If there are no changes in that block, (0,0) is directly assigned as its motion vector.
- Else, Three Step Search Algorithm (TSS) is applied for block matching to determine motion vectors for that block.

# How we reduced time complexity!

- In standard video codecs, macroblock division is performed by finding motion vectors for all possible combinations of macroblock modes and the mode with least error is used.

- The proposed algorithm determines macroblock modes using image difference technique without computing motion vectors for every possible combination of macroblock modes.

# Results

# Dataset used

- VIRAT video Dataset
- 1280 x 720
- 30 frames per second
- Videos from fixed focus cameras at parking lots, restaurants, institution campus, etc.
- Average compression 90.19 %

# Compression Results

| Video Sample | Raw Video Size [MB] | Compressed Video Size [MB] | Compression [%] |
|---|---|---|---|
| 1 | 48918 | 4158.03 | 91.5 |
| 2 | 7595.5 | 873.97 | 88.34 |
| 3 | 29666.4 | 3079.37 | 89.62 |
| 4 | 33927 | 2123.83 | 93.74 |
| 5 | 43395 | 7281.68 | 83.22 |
| 6 | 54835.5 | 2856.93 | 94.79 |
| 7 | 39844.5 | 3446.55 | 91.35 |
| 8 | 45130.8 | 4995.98 | 88.93 |

# Computational Complexity

Computational complexity of determining macroblock mode in the present algorithms is:

$$C_{standard} = N \times \sum_{P} \sum_{SB} \{(dimension\ of\ SB) \times N_{ME}\}$$
$$= N \times 256 \times N_{PC} \times N_{ME}$$

where, $N$ = number of macroblocks per frame
$P$ = Possible combination
$SB$ = Sub-block
$N_{ME}$ = number of iterations required for motion estimation of a sub-block
$N_{PC}$ = number of possible combinations of macroblock modes for a $16 \times 16$ macroblock

The average value of $N_{ME}$ is 25 and value of $N_{PC}$ is 259 [5]. Hence, the average value of $C_{standard} = 256 \times N \times 259 \times 25 = 1657600N$.

Computational complexity of determining macroblock mode in the proposed method,

$$\begin{aligned} C_{proposed} &= (dimensions\ of\ a\ macroblock) \times (dimensions\ of\ matrix\ M2) \times N \\ &= (16 \times 16) \times (4 \times 4) \times N = 4096N \end{aligned}$$

where, $N$ = number of macroblocks per frame

Ratio of computational complexities $= \frac{C_{standard}}{C_{proposed}} = \frac{1657600}{4096} =$ 405:1 (approx.)

# Conclusion

The use of **Image Difference Module** and **Macroblock division module** in the proposed framework greatly reduces the complexity of the process of **determining appropriate macroblock modes** for each 16x16 macroblock in every frame.

From experimental results we conclude:

1. Reduction in computational complexity by a **factor of approx. 400 times**
2. **Compression levels of 90%** on average

# Literature Survey


- T. Wiegand, G. J. Sullivan, G. Bjontegaard and A. Luthra, "Overview of the H.264/AVC video coding standard," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 7, pp. 560-576, July 2003, doi: 10.1109/TCSVT.2003.815165.
- VCodex H.264 documentation - https://www.vcodex.com/h264-resources/.
- Ze-Nian Li, Mark S. Drew, Jiangchuan Liu “Fundamentals of Multimedia“.
- Iain E. G. Richardson The Robert Gordon University, Aberdeen, UK “H.264 and MPEG-4 Video Compression Video Coding for Next-generation Multimedia”.
- L.C.Manikandan, Dr.R.K.Selvakumar, IEEE Senior Member “A Study on Block Matching Algorithms for Motion Estimation in Video Coding” International Journal of Scientific & Engineering Research, Volume 5, Issue 7, July-2014.

# Literature Survey


- K. Panusopone and K. R. Rao, "Efficient motion estimation for block based video compression," 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, Munich, 1997, pp. 2677-2680 vol.4, doi: 10.1109/ICASSP.1997.595340.
- D. Marpe, H. Schwarz, and T. Wiegand, "Context-adaptive binary arithmetic coding in the H.264/AVC video compression standard," IEEE Trans. Circuits Syst. Video Technol., vol. 13, pp. 620–636, July 2003.
- W. Hassen and H. Amiri, "Block Matching Algorithms for motion estimation," *2013 7th IEEE International Conference on e-Learning in Industrial Electronics (ICELIE)*, Vienna, 2013, pp. 136-139, doi: 10.1109/ICELIE.2013.6701287.
- L. Czuni, G. Csaszar and A. Licsar, "Estimating the Optimal Quantization Parameter in H.264," 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, 2006, pp. 330-333, doi: 10.1109/ICPR.2006.502.

# Thank You