Video Compression

INSTRUCTOR: YAN-TSUNG PENG

DEPT. OF COMPUTER SCIENCE, NCCU

CLASS 1

Syllabus

- Time: Thur (9:10 – 12:00)
- Location: 大仁200102
- Textbook: Video Codec Design: Developing Image and Video Compression Systems, Wiley
- TA: 陳子昊 magecliff96@gmail.com; Office Hours: 13-14:00, Location: 大仁200308
- References:

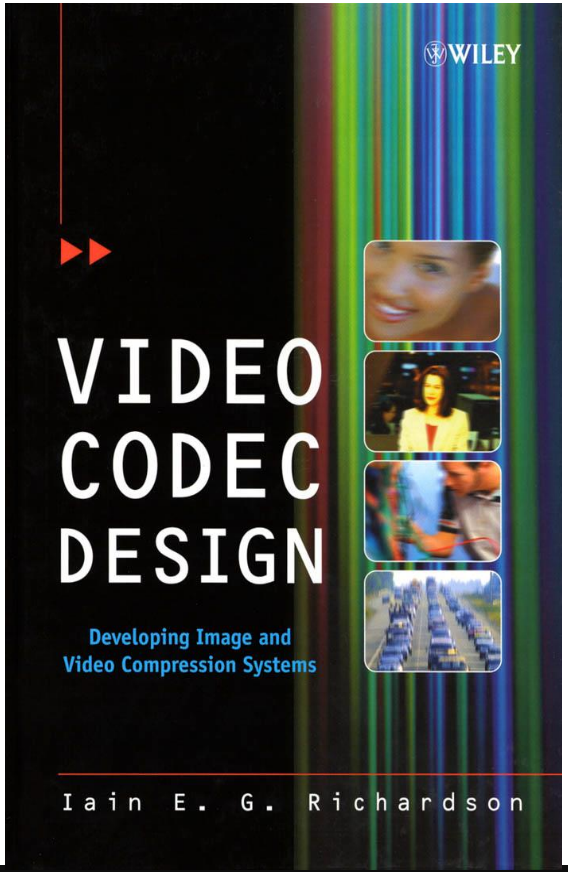
1. John Watkinson, “MPEG Handbook,” Focal Press, 2001
2. Gary J. Sullivan et al., “Overview of the High Efficiency Video Coding (HEVC) Standard,” IEEE TCSVT, 2012

Textbook

- Video Codec Design: Developing Image and Video Compression Systems, by Iain E. Richardson

Content List:

1. Introduction
2. Digital Video
3. Image and Video Compression Fundamentals
4. Video Coding Standards: JPEG and MPEG
5. Video Coding Standards: H.261, H.263, and H.26L
6. Motion Estimation and Compression
7. Transform Coding
8. Entropy Coding
9. Pre- and Post-processing
10. Rate, Distortion, and Complexity
11. Transmission of Coded Video
12. Platform
13. Video Codec Design
14. Future Developments



Topics

Content List:

Fundamental Concepts
1. Introduction
2. Digital Video
3. Image and Video Compression Fundamentals
4. Video Coding Standards: JPEG and MPEG
5. Video Coding Standards: H.261, H.263, and H.26L

Component Design
6. Motion Estimation and Compression
7. Transform Coding
8. Entropy Coding
9. Pre- and Post-processing

System Design
10. Rate, Distortion, and Complexity
11. Transmission of Coded Video
12. Platform
13. Video Codec Design
14. Future Developments

NCCU Moodle Platform

- **Materials:** Course slides
- **Grading:** Assessment criteria and grading details
- **Homework:** Submit reports via the course portal

Overview

- Recording videos, sharing them on social media, sending video messages, and streaming content on platforms like YouTube have become an integral part of our daily lives. None of this would be possible without **video compression**.
- This course provides students with a **fundamental understanding of video compression**, exploring the complete video coding workflow and key techniques. By the end of the course, students will gain insights into the most widely used video coding standards, **H.264** and **H.265**.

Goals

- Students will learn the fundamental theories of image and video compression. Additionally, they will be required to implement a simplified video codec as part of the coursework.

Grading Policy

- Homework (30%) – Two to three assignments (programming + report)
- Midterm Exam(30%)
- Final Project (40%) – Paper presentation
- Class Participation (10%) – Quiz
  - Awarded to students who attend class and complete the quiz. **No exceptions.**

Grading Policy

- Homework must be completed individually
- You may use **C/C++, Python, Matlab**, etc.
- If plagiarism is detected, both the copier and the original author will **share the credit evenly**.
- Homework must include **both code and a report**.

- Late Submission Penalty:
  - Up to 3 days late: 70% of the points
  - Up to 6 days late: 50% of the points
  - 7 days or more: No points awarded

Final Presentation – Paper Study

- The **final presentation** will take place during **Week 18**.
- You are required to study a paper related to **video compression**, published in top conferences or journals.
- **Recommended Sources:**
  - **Top Conferences:**
    - CVPR, ICCV, ECCV
    - Data Compression Conference
    - IEEE International Conference on Image Processing (ICIP)
  - **Top Journals:**
    - IEEE Transactions on Circuits and Systems for Video Technology
    - IEEE Transactions on Image Processing
    - IEEE Transactions on Multimedia
- Choose a paper from these or similar reputable sources for your presentation.

Paper Presentation

- Please ensure your presentation covers the following key points:
  - **Introduction:**
    - Introduce the topic and explain its significance in the field.
  - **Related Work:**
    - Discuss previous approaches to the problem and how they attempted to solve it.
  - **Presented Method:**
    - Provide a detailed explanation of the presented method.
    - Highlight its **novelty**, the specific **problems it addresses**, and its **key contributions**.
  - **Experimental Results:**
    - Present results that support the method’s contributions and validate its effectiveness.
  - **Conclusions:**
    - Summarize the paper’s findings.
    - Share your personal insights, critiques, or thoughts on the work.

How to read a paper?

Abstract: rovide a concise overview of the work, highlighting its purpose and performance.

Introduction: Cover three key aspects:

1. Introduce the problem being addressed.
2. Discuss the history and evolution of related methods.
3. Explain the value and significance of the proposed method.

Related works: Review existing methods relevant to the proposed approach and discuss their connections and differences

Proposed method: the core section of your presentation. Offer a detailed explanation of the proposed method, including its structure, unique features, and how it improves upon previous approaches.

***Abstract*—We propose a weighted boundary matching error concealment method for HEVC. It uses block partition decisions to improve a common block matching algorithm that finds blocks with the best matched boundaries from the previous frame to conceal the currently corrupted blocks. The block partition decisions from the co-located block of the corrupted one are exploited. For each partition, a summed boundary weight is computed; the one with the highest weight is chosen to be concealed next. Experimental results show the proposed method performs better than conventional error concealment methods objectively and subjectively.**

***Index Terms*—Temporal error concealment, boundary matching algorithm, HEVC (High efficiency video coding), block partitions.**

I. Introduction

With the growing popularity of high resolution (HD) video, the High Efficiency Video Coding (HEVC) standard [1] has been recently developed jointly by the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG) standardization organizations. HEVC, evolved from H.264/MPEG-4 Advanced Video Coding (AVC) [2], aims to address two key issues: increased video resolution and increased use of parallel processing architectures. The HEVC standard is designed to increase coding efficiency for encoding video with higher resolutions and to introduce parallel decoding syntax to expedite the decoding process [3].

To achieve efficient compression for higher resolutions, HEVC provides coding units (CU) ranging from 64×64 to 8×8 and prediction units (PU) that can further split a CU down to 4×4 for finer video quality. Additionally, a CU can be partitioned asymmetrically to keep the shape of an object, such as $M/2 \times M$ or $M/4 \times 3M/4$, where M is 16 or larger for luma. However, since packet loss happens when video is transmitted through an unreliable network, HEVC coded videos are more vulnerable in transmission, as a lost packet with the same size may cause corruption of a larger region in HEVC than in H.264/AVC. Moreover, since HEVC suggests no error concealment (EC) method, it is crucial to develop effective EC for HEVC.

There is little literature on HEVC error concealment. In [4], [5], motion vector (MV) extrapolation based on various CU partition decisions is applied to conceal whole frame loss or CU loss. The CU partitions in a lost frame are evaluated and extrapolated according to the MV correlation from the co-located CUs in [4]. In [5], the extrapolated and overlapped partition size is used to decide an extrapolated MV for a lost CU. Ref. [6] uses co-located CU partitions and MVs in the previous frame to recover the lost CU. The partitions are merged and the MVs are refined based on the residual energy for motion compensation. However, all of them fail to consider the spatial smoothness of the lost CUs, often resulting in boundary misalignments and degrading visual quality.

In this paper, we improve the traditional boundary matching algorithm (BMA) by using partition decisions. We adopt the co-located partition decisions from the previous frame for lost CUs because it is observed that the CU depths and the PU partitions show strong temporal correlations between the previous and current frame [7], [8]. Since each partition often represents an object or block segmentation, we perform a weighted BMA for each partition separately to not only maintain spatial smoothness but also to recover the objects in the lost regions more precisely.

The rest of this paper is organized as follows: The proposed method is described in Section II. The subjective and objective experimental results are shown in Section III. Finally, Section IV summarizes the conclusions.

II. Proposed Method

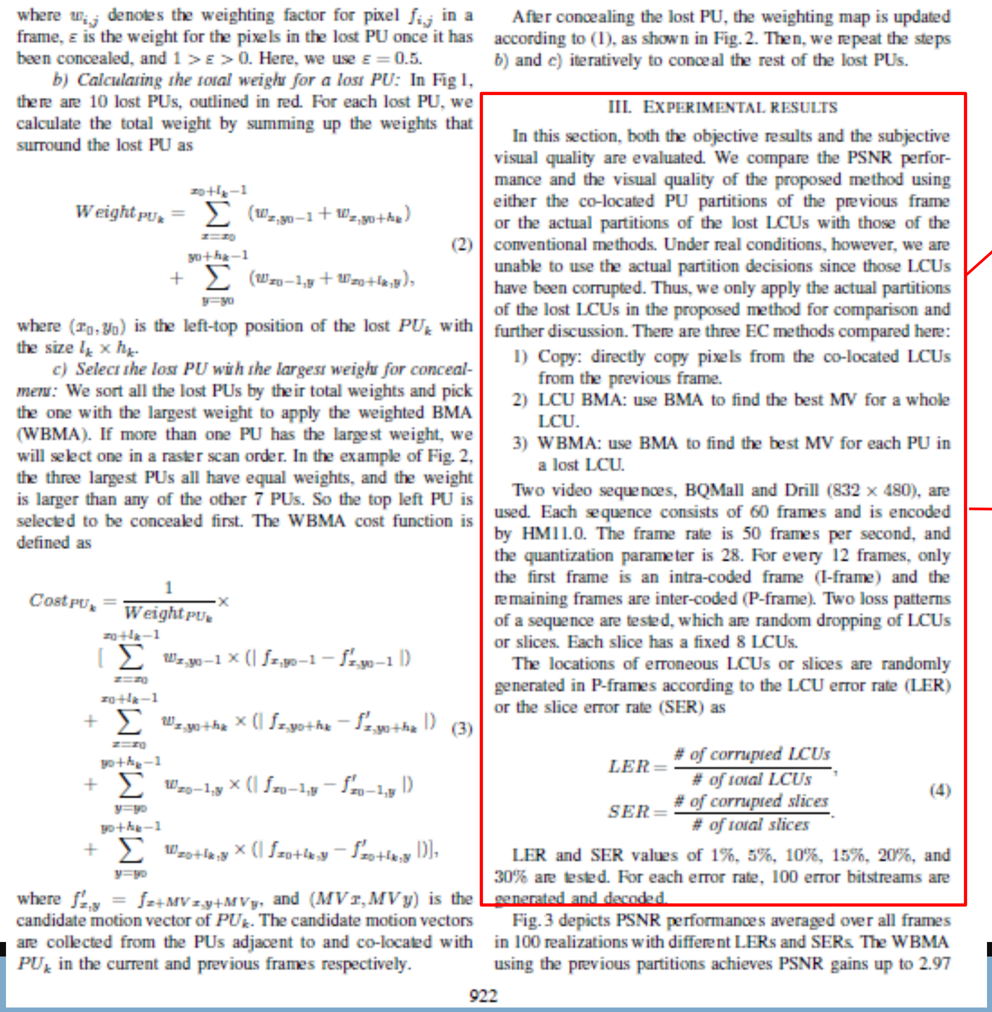
In this section, we present a weighted boundary matching EC method based on the block partition decisions from the previous frame. For each partition, the block matching algorithm is performed separately to conceal the lost partition according to the summed boundary weight. In a frame with lost LCUs, each lost LCU will be concealed sequentially.

For a lost Largest Coding Unit (LCU), we use the partition decisions from the co-located LCU. The proposed weighted BMA includes the following steps:

a) Constructing an initial weighting map: Fig. 1 shows the initial constructed weighting map for a frame with a lost LCU. For a $m \times n$ frame $F \in R^{m \times n}$ with lost LCUs, the weighting map $W \in R^{m \times n}$ is defined as

$$w_{i,j} = \begin{cases} 1, & \text{if } f_{i,j} \text{ is correctly received,} \\ \varepsilon, & \text{if } f_{i,j} \text{ is concealed,} \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

How to read a paper?



where $w_{i,j}$ denotes the weighting factor for pixel $f_{i,j}$ in a frame, ε is the weight for the pixels in the lost PU once it has been concealed, and $1 > \varepsilon > 0$. Here, we use $\varepsilon = 0.5$.

b) Calculating the total weight for a lost PU: In Fig 1, there are 10 lost PUs, outlined in red. For each lost PU, we calculate the total weight by summing up the weights that surround the lost PU as

$$Weight_{PU_k} = \sum_{x=x_0}^{x_0+l_k-1} (w_{x,y_0-1} + w_{x,y_0+h_k}) + \sum_{y=y_0}^{y_0+h_k-1} (w_{x_0-1,y} + w_{x_0+l_k,y}), \quad (2)$$

where (x_0, y_0) is the left-top position of the lost PU_k with the size $l_k \times h_k$.

c) Select the lost PU with the largest weight for concealment: We sort all the lost PUs by their total weights and pick the one with the largest weight to apply the weighted BMA (WBMA). If more than one PU has the largest weight, we will select one in a raster scan order. In the example of Fig. 2, the three largest PUs all have equal weights, and the weight is larger than any of the other 7 PUs. So the top left PU is selected to be concealed first. The WBMA cost function is defined as

$$Cost_{PU_k} = \frac{1}{Weight_{PU_k}} \times \Big[\sum_{x=x_0}^{x_0+l_k-1} w_{x,y_0-1} \times (| f_{x,y_0-1} - f'_{x,y_0-1} |) + \sum_{x=x_0}^{x_0+l_k-1} w_{x,y_0+h_k} \times (| f_{x,y_0+h_k} - f'_{x,y_0+h_k} |) + \sum_{y=y_0}^{y_0+h_k-1} w_{x_0-1,y} \times (| f_{x_0-1,y} - f'_{x_0-1,y} |) + \sum_{y=y_0}^{y_0+h_k-1} w_{x_0+l_k,y} \times (| f_{x_0+l_k,y} - f'_{x_0+l_k,y} |)\Big], \quad (3)$$

where $f'_{x,y} = f_{x+MVx,y+MVy}$, and (MVx, MVy) is the candidate motion vector of PU_k. The candidate motion vectors are collected from the PUs adjacent to and co-located with PU_k in the current and previous frames respectively.

After concealing the lost PU, the weighting map is updated according to (1), as shown in Fig. 2. Then, we repeat the steps *b)* and *c)* iteratively to conceal the rest of the lost PUs.

III. EXPERIMENTAL RESULTS

In this section, both the objective results and the subjective visual quality are evaluated. We compare the PSNR performance and the visual quality of the proposed method using either the co-located PU partitions of the previous frame or the actual partitions of the lost LCUs with those of the conventional methods. Under real conditions, however, we are unable to use the actual partition decisions since those LCUs have been corrupted. Thus, we only apply the actual partitions of the lost LCUs in the proposed method for comparison and further discussion. There are three EC methods compared here:

1) Copy: directly copy pixels from the co-located LCUs from the previous frame.
2) LCU BMA: use BMA to find the best MV for a whole LCU.
3) WBMA: use BMA to find the best MV for each PU in a lost LCU.

Two video sequences, BQMall and Drill (832×480), are used. Each sequence consists of 60 frames and is encoded by HM11.0. The frame rate is 50 frames per second, and the quantization parameter is 28. For every 12 frames, only the first frame is an intra-coded frame (I-frame) and the remaining frames are inter-coded (P-frame). Two loss patterns of a sequence are tested, which are random dropping of LCUs or slices. Each slice has a fixed 8 LCUs.

The locations of erroneous LCUs or slices are randomly generated in P-frames according to the LCU error rate (LER) or the slice error rate (SER) as

$$LER = \frac{\#\ of\ corrupted\ LCUs}{\#\ of\ total\ LCUs}, \quad SER = \frac{\#\ of\ corrupted\ slices}{\#\ of\ total\ slices}. \quad (4)$$

LER and SER values of 1%, 5%, 10%, 15%, 20%, and 30% are tested. For each error rate, 100 error bitstreams are generated and decoded.

Fig. 3 depicts PSNR performances averaged over all frames in 100 realizations with different LERs and SERs. The WBMA using the previous partitions achieves PSNR gains up to 2.97

922

Experimental Results: about experiment design

1) describing experiment settings
2) introducing experiment data and performance metrics
3) listing all the compared methods
4) demonstrating objective and subjective assessment
5) discussing ablations and limitations

Conclusion: summarizing the paper by

1) reciting the proposed design and contributions (novelty)
2) describing the performance
3) proposing possible future directions and extensions

Course Topics

- Introduction & syllabus
- Mathematical background
- Human visual system, color space, and video formats
- Predictive coding - spatial prediction
- Predictive coding - temporal prediction (motion estimation and compensation)
- Midterm
- Transform coding and quantization
- In-loop filter and entropy coding
- Rate-distortion optimization
- Deep Learning based Video Compression
- Final Presentation

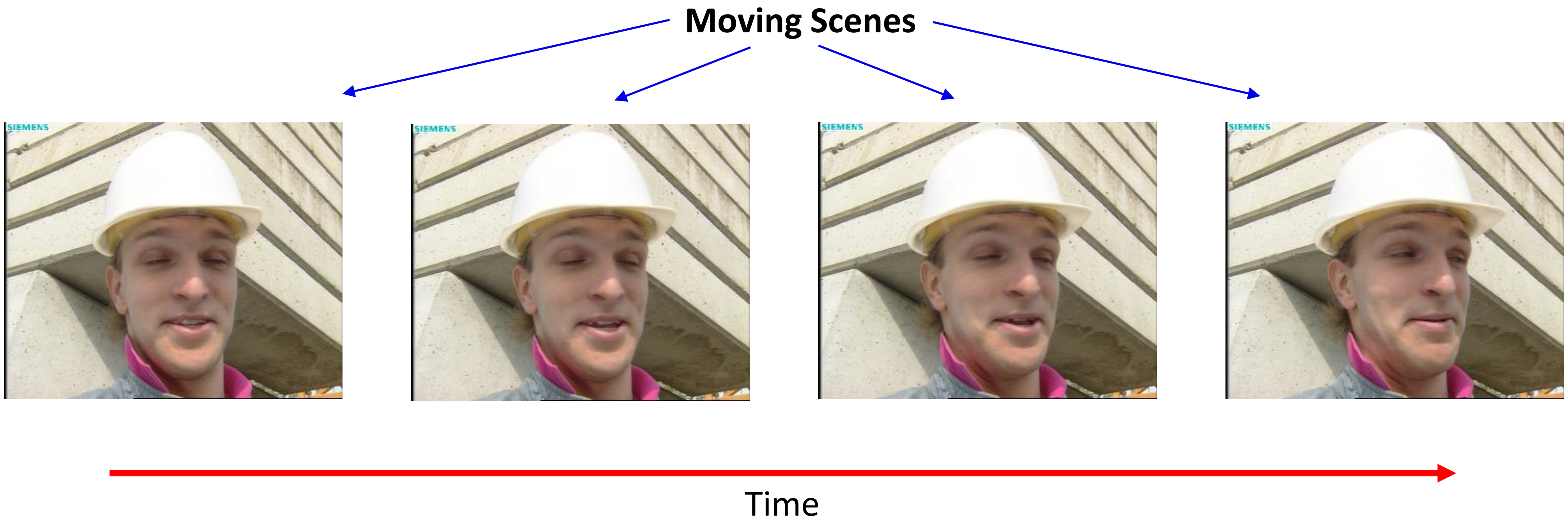
Math Background

- Linear Algebra
- Probability
- Fourier Transform

Introduction to Video Compression

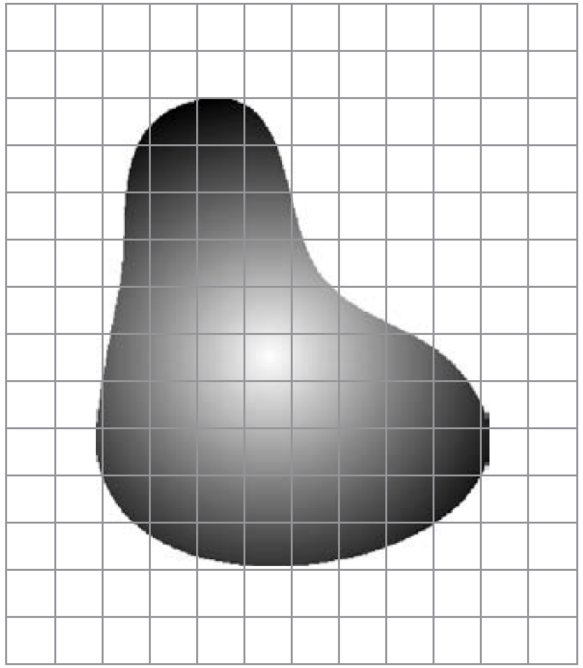
What is a Video

Definition: a video consists of continuous video frames (images) that are usually correlated both spatially and temporally

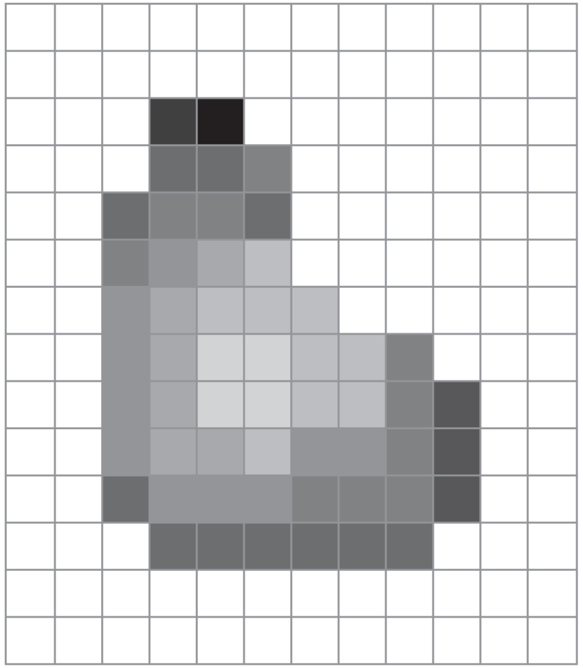


Digital Video

Spatial sampling – Frame Resolution



Continuous frame



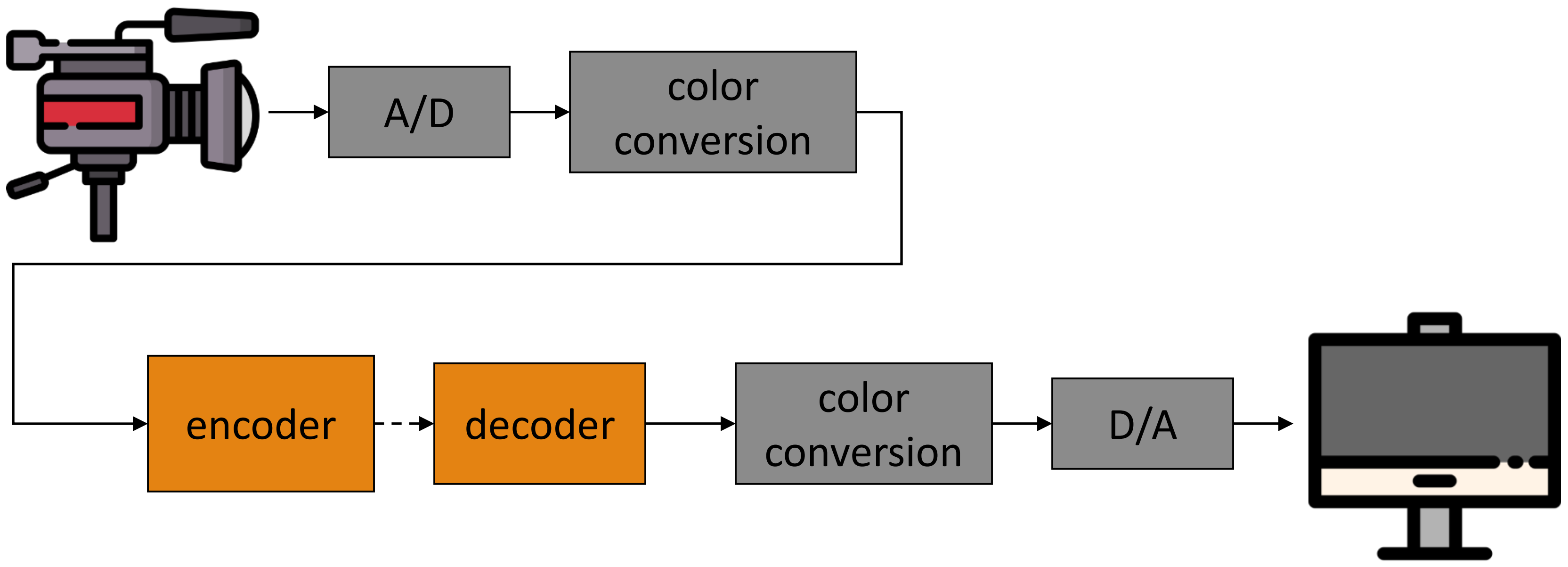
Sampling & Quantization

Temporal sampling – Frame Rate (frame per second, fps)

30 fps

15 fps

Video Systems



Why do we need compressions?

- Why compressions
  - storage requirements
  - bandwidth requirements (storage, network)
- Objective of Compression
  - removing information redundancy from data to reduce its size necessary to reproduce the original data while maintaining required levels of coded “quality”
  - processing delay
  - implementation complexity

Compression vs Redundancy

- Data vs Information
  - Definition: The same amount of information can be represented by various amounts of data
  - Data is a means to convey information
- Irrelevant or repeated information contains redundant data
- Define Relative Data Redundancy R

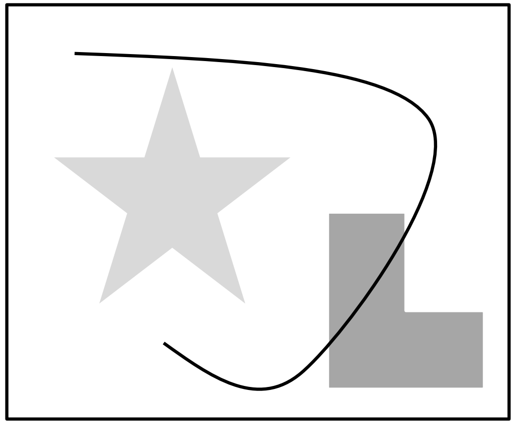
$$R = 1 - \frac{1}{C},$$

where $C = \frac{b}{b'}$, called Compression Ratio

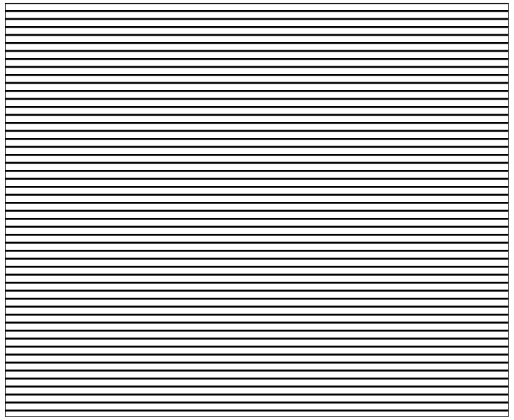
b and b' represent the numbers of bits in two representations (Ori and Compressed) of the same information

For example, if $C = 10, R = 0.9$, indicating 90% of the data in Ori is redundant

Redundancy



Coding Redundancy

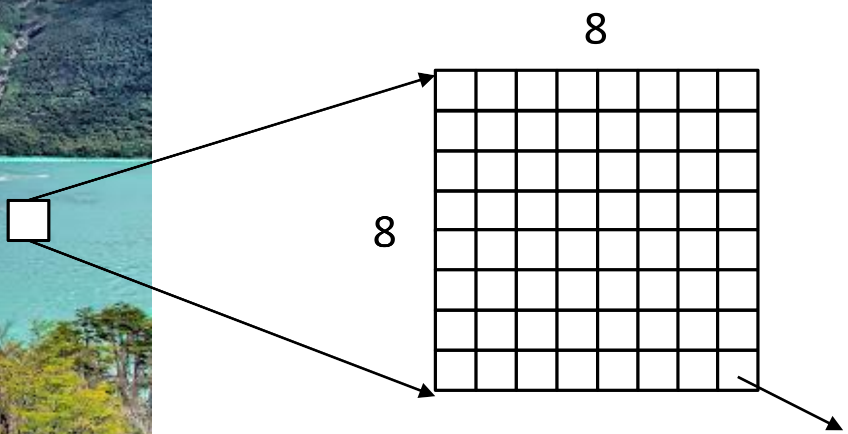


Spatial Redundancy

Irrelevant Redundancy

Temporal Redundancy

If we don’t compress it



[Assume RGB color space used]

A **pixel** consists three color components:
R, G, and B (red, green, and blue),
each of which takes 8 bits.

$3 \times 8 = 24$ (bits/pixel)

Assume each component takes 8 bits

2-hour 30 fps 4K uncompressed video = $\underbrace{4096*2160}_{\text{4K}}*\underbrace{24}_{\text{24 bit}}*\overbrace{30}^{\text{30 fps}}*\underbrace{7200}_{\text{total seconds}} \approx$ 5 TByte

Compression Ratio vs Video Quality

High Quality
Low Compression Ratio

Low Quality
High Compression Ratio

Video Codecs Used by Major Platforms

- YouTube
  - VP9, AV1 (Gradual adoption, In 2018, YouTube began deploying AV1)
- Facebook & Instagram
  - H.264 (Primary codec), AV1 (Planned adoption)
- TikTok
  - H.264 (Standard codec)
- **General Trends**
  - **VP9 & AV1**: Used for improved compression and streaming efficiency.
  - **H.264**: Still widely used due to broad compatibility.

https://en.wikipedia.org/wiki/AV1

Comparison among H.264, VP9, and AV1

Feature	H.264	VP9	AV1
Release Year	2003	2013	2018
Compression Efficiency	Baseline	~50% better than H.264	~30% better than VP9
Licensing	Royalty-based	Royalty-free	Royalty-free
Hardware Support	Universal	Moderate	Emerging
Encoding Complexity	Low	Medium	High
Typical Use Cases	HD streaming, social media	4K streaming, YouTube	4K/8K streaming, future platforms

What You Will Learn

- Fundamental Compression Techniques:
  - Gain a deep understanding of core compression methods for images, videos, and beyond.
- Hands-On Experience:
  - Develop practical skills by implementing functional units within a video codec.