Departamento de Engenharia Electrotécnica e de Computadores

Multimedia systems - M.EEC0057

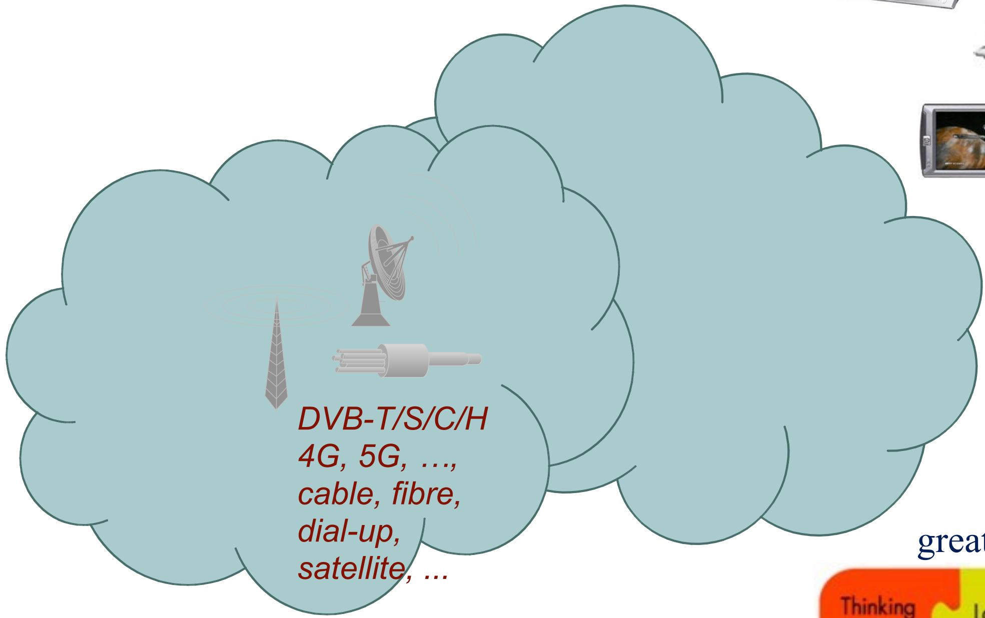
M.EEC 2023-2024

- Multimedia streaming challenges

Current landscape

great diversity of content sources, both professional and domestic

different access and core networks

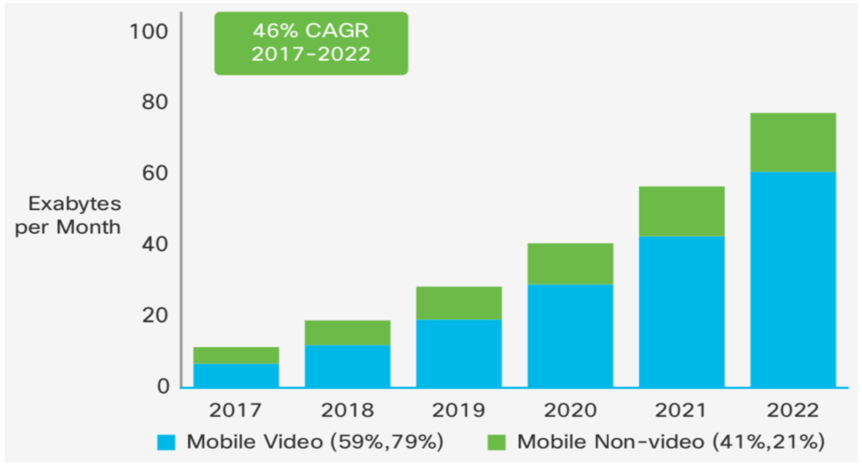


great diversity of client devices

great diversity of audiences

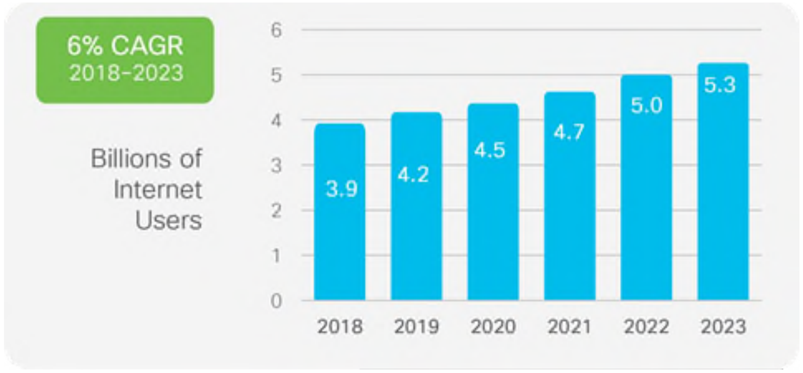
Situation and forecast

- exponential consumption of video on the Internet and mobile environments
  - in 2022, more than four-fifths of the world's mobile data traffic was video
  - following this trend, it is assumed that the IP video traffic will be more than 95% of the Internet traffic by 2030



source: Cisco Visual Networking Index: Global Mobile Data Traffic

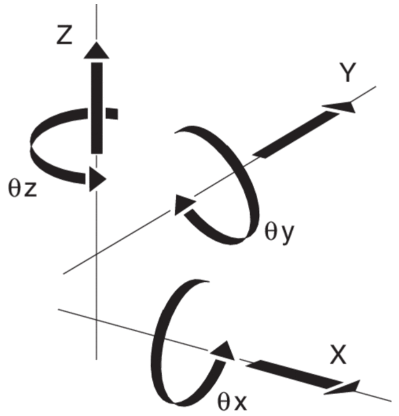
- exponential number of devices on the Internet and mobile environments
  - nearly two-thirds of the global population have Internet access (2023)



source: Cisco Annual Internet Report

Trends in networked multimedia

- Very high resolution media
- Beyond 2D
  - 3D, multi view, 360°, 6DoF
- Immersion
- Augmented reality
- Adaptability / Personalisation
  - Large amount of resources needed to support such advanced formats!



How to address challenges?

- compatibility of formats
    - through collaborative work on unified standards
        - involving players from all areas of the content value chain
    - this would partially addressed the negative aspect on users' frustration
- intelligent forms of using resources and adapting to dynamic changes
    - more efficient and flexible encoding schemes
    - cooperation between application and network
        - thus increasing the overall level of quality of experience for users
- intelligent forms of using freely available information
    - to be able to adapt and also meet users' expectations
        - through the use of sensors and data mining and artificial intelligence techniques to process historical and sensed data
            - in a non-intrusive way
            - without invading privacy
    - understand the context of usage or situation the user is in when consuming multimedia
    - taking profit of existing general-purpose and freely available frameworks

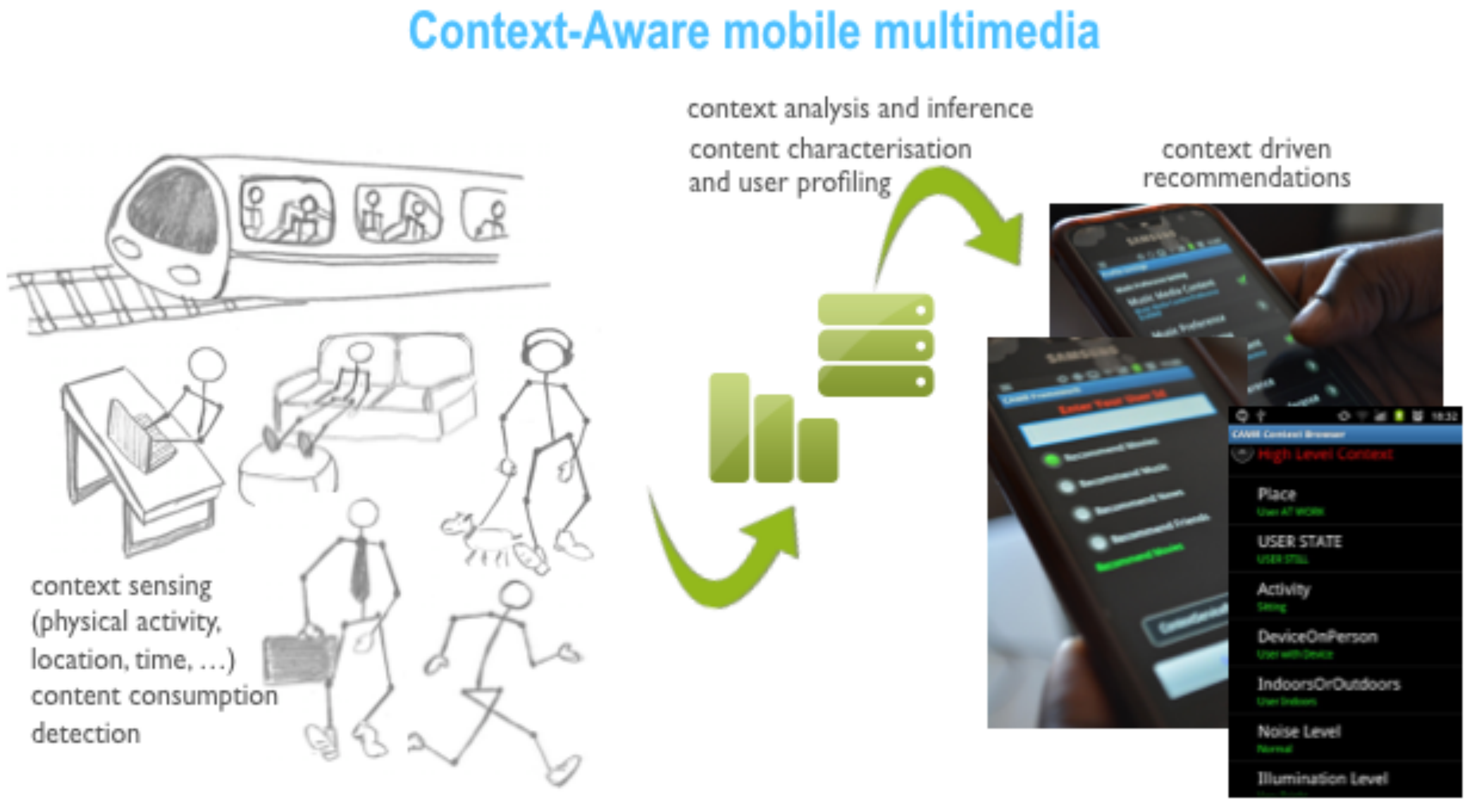
bringing intelligence into multimedia applications and services

- Recently, big companies have been investing on the development of Artificial Intelligence frameworks
    - making freely available Application Programming Interfaces (APIs)
        - enabling programmers to use built-in intelligent algorithms for different purposes
            - e.g., image, video and audio classification, object detection and context recognition
        - examples of such AI frameworks:
            - Google Cloud ML APIs
                - < https://cloud.google.com/products/ai>
            - Facebook AI and Machine Learning API
                - <https://ai.facebook.com>
                - <https://research.fb.com/category/applied-machine-learning/>
            - IBM Watson
                - <https://www.ibm.com/watson/)
            - Microsoft Cognitive Services API
                - <https://www.microsoft.com/cognitive-services/>

bringing intelligence into multimedia applications and services (2)

- innovative solutions for personalised access and consumption of multimedia content, through:
    - context awareness
        - sensing and reasoning using existing AI frameworks to understand consumption conditions / situations
    - content recommendation
        - based on results of analysing historical data (with existing AI frameworks and tools)
    - content adaptation
        - using newly defined standards, protocols and tools
            - HLS and MPEG-DASH
            - GPAC
            - SRT
    - immersive multi-view experiences
        - including advanced media formats
        - using emergent media standards together with content adaptation and context-awareness
        - devising new seamless interfaces, non-intrusive

Context-awareness, personalisation


Context-Aware mobile multimedia
context analysis and inference
content characterisation and user profiling
context driven recommendations
context sensing (physical activity, location, time, …) content consumption detection
High Level Context
Place
USER STATE
Activity
DeviceOnPerson
IndoorsOrOutdoors
Noise Level
Illumination Level

Requirements of streaming

- Temporal sequence
- Data integrity
    - In practice, video streaming can be expressed by a set of **timing requirements**
        - time interval Δ between images presentation (usually 40 ms) must be preserved
            - each image must be sent and decoded within its presentation time
    - the sequence of images must observe the following rules:
        - image n transmitted and decoded up to instant Tn
        - image (n+1) transmitted and decoded up to instant $Tn + \Delta$
        - image (n+2) transmitted and decoded up to instant $Tn + 2\Delta$
        - …
    - data that arrives after the indicated instant can no longer be used except to assist the decoding of other images
        - it depends on the compression scheme

Challenges faced

- **current networks** exhibit, in more or less degree,
  - bandwidth variability, **congestion, error rate**
    - **which may lead to delays,** delay **jitter, losses**
- this has implications on the ability to respect the timing constraints of multimedia applications
  - data may arrive too late to be used
  - lip-sync may not be possible
- as well as on other quality constrains that compromise Quality of Experience
  - long delays, leading to re-buffering during visioning
  - data may arrive with errors or not arrive at all, which may not be possible to recover

Trade-offs between transmission network and source coding

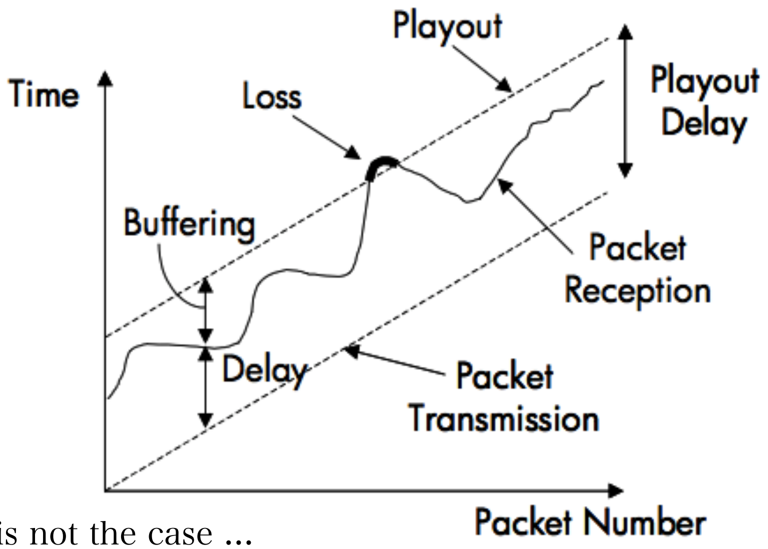
- The ultimate goal of source coding (compression)
    - achieving the lowest possible distortion for a given target bit rate
        - Minimise bit rate whilst maximising quality
- The goal of transmission networks
    - deliver reliable information at a rate that is as close as possible to the channel's full capacity
        - Efficient use of resources whilst proving a reliable service
- The ideal would be to work collaboratively
    - joint optimisation

Challenges faced - delay and jitter

- end-to-end delay may vary from packet to packet - **jitter**
  - real problem, as the rate of presentation of images should always be the same (after the pre-roll delay, where initial data is buffered)
    - it may occur the case when the decoder needs information from an image and that data has not yet arrived
      - **buffer underflow**
    - or the decoder is receiving more data than the amount it can actually process
      - **buffer overflow**
  - this causes a picture degradation referred to as **"jerkiness"**
    - the movement in the video sequence is ruined
    - the decoder may have to skip an image or to delay the presentation of an image (thus "freezing" the previous image)

Challenges faced on the network - delay and jitter

- How to compensate jitter?
  - jitter may be attenuated through the use of a playout buffer
    - it compensates the variations of the delay (i.e., the jitter) but it introduces an additional delay
  - if a maximum bound of the jitter is assured by the network, then a conveniently sized buffer will solve the problem



    - but many times this is not the case …

Challenges faced - errors and losses

- depending on the type of network, **different errors and losses** may occur
    - in wired networks such as Ethernet, it is usual that entire packets are lost
    - in wireless networks, there are single bit errors or bursts of errors
- losses have a direct negative impact on the quality of the recovered pictures
    - to fight this impact, the application may implement error control mechanisms (**channel coding** and **source coding** approaches)

        (1) forward error correction (FEC)

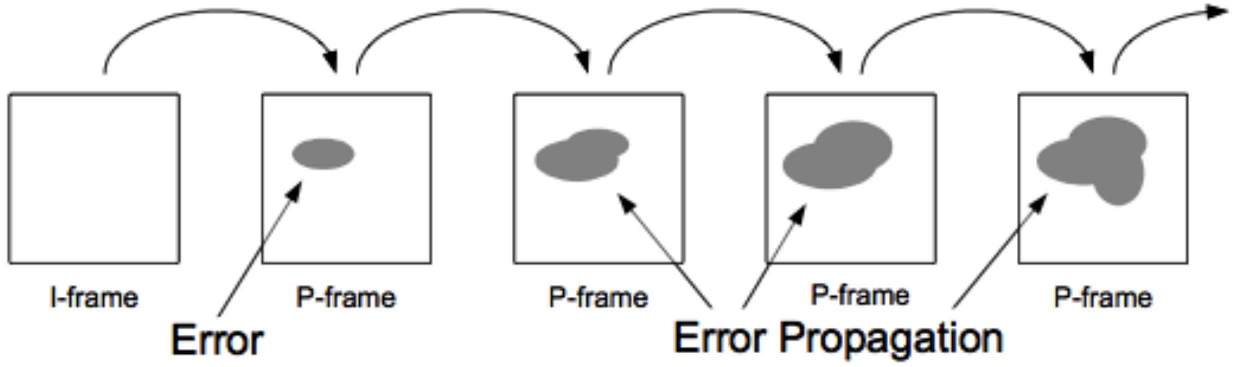
        (2) retransmissions

        (3) error concealment

        (4) error-resilient video coding

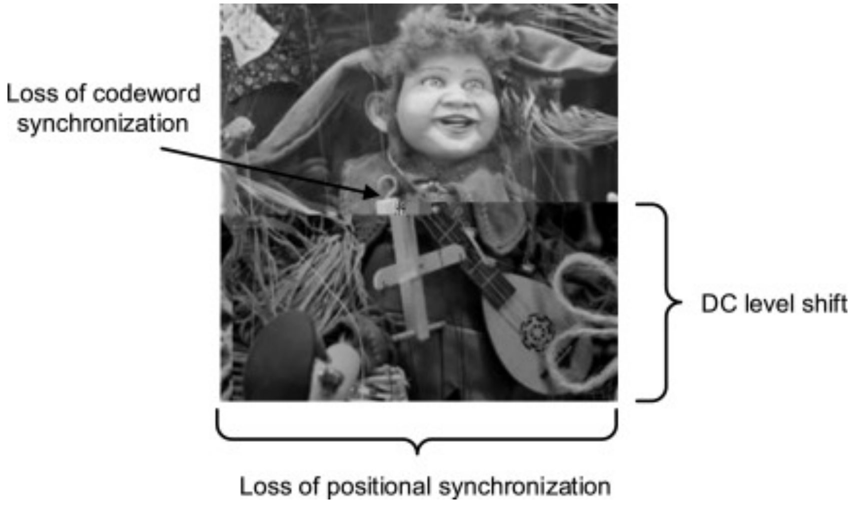
Effects of errors and losses

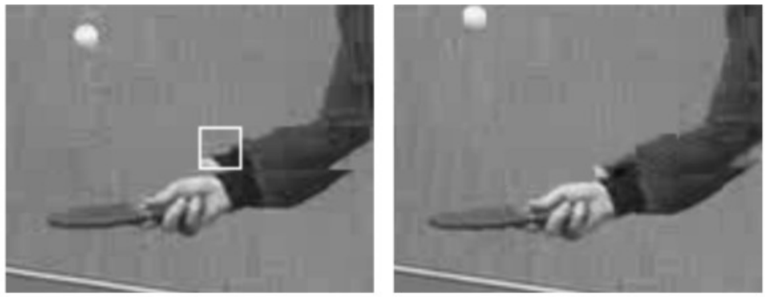
- most popular compression schemes are based on
  - **prediction with motion compensation**, DCT or other spatial transform, entropy coding with **variable length codes**
- in this kind of schemes one of the most common problems is
  - missing areas of images
  - propagation of errors
  - loss of VLC synchronisation

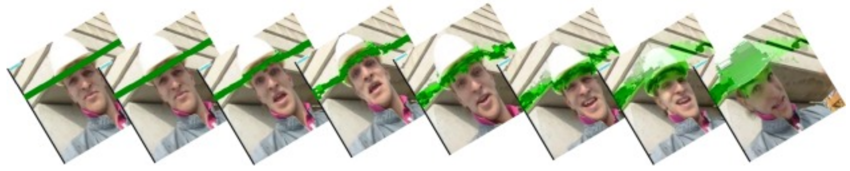


Effects of errors and losses (2)

- Effects of transmission errors


Loss of codeword synchronization
DC level shift
Loss of positional synchronization





Challenges faced - errors and losses (2)

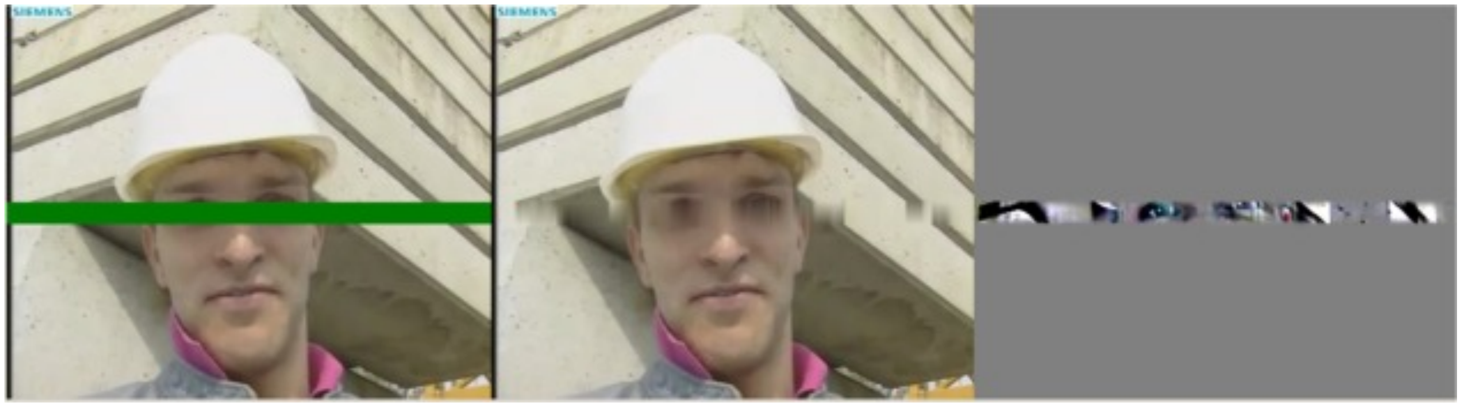
- how to fight errors and losses locally?
  - (1) through channel coding techniques
- FEC adds redundancy to the bit stream, the additional bits being used to recover the errors
  - for ex., Read-Solomon (RS) block codes
    - for each block of K packets or bits it generates N packets or bits (N>K)
    - the N-K packets are redundant packets or bits
    - if at least K packets or bits are correctly received, then the code is able to recover all the errors in the remaining N-K packets or bits
    - increases the bit rate by a factor of N/K

Challenges faced - errors and losses (3)

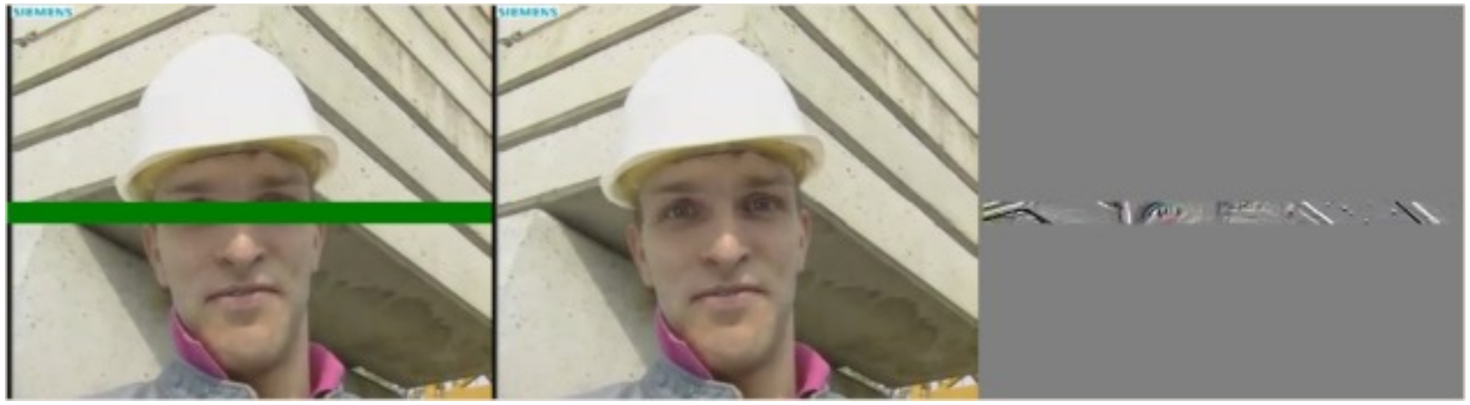
- how to fight errors and losses locally?
  - (2) through source coding techniques
- **error concealment** estimates the amount of lost information to conceal or hide the error (decoder side)
  - it takes advantage of the spatial and temporal redundancy that exists in the video signal
  - part of the existing correlation is explored during the encoding process to achieve compression
  - un-explored correlation can be used to predict the loss of data in the decoder
    - spatial and/or temporal interpolation or extrapolation is used to estimate the lost data
      - when image blocks or samples are missing due to transmission errors, the decoder can estimate them based on surrounding received samples
      - using inherent correlation among spatially and temporally adjacent samples

Error Concealment results

- Spatial error concealment



- Temporal error concealment



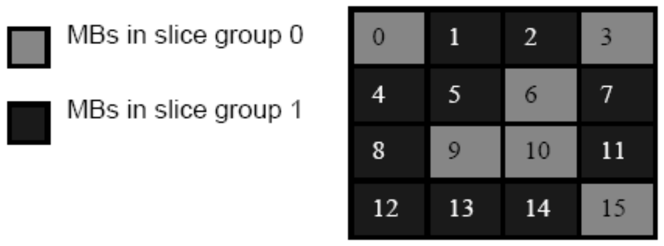
Challenges faced - errors and losses (4)

- how to fight errors and losses?
    - (2) through source coding techniques
- **error resilient video coding** incorporates in the coding algorithm itself, mechanisms that are robust to errors (encoder side)
    - In MPEG-2 video
        - possibility of using prediction through “motion vectors” within an I image
    - Tools for robustness in h.264 / MPEG-4 part 10
        - Flexible Macroblock Ordering (FMO)
        - Arbitrary Slice Ordering (ASO)
        - Data Partitioning (DP)
        - Redundant Slices (RS)

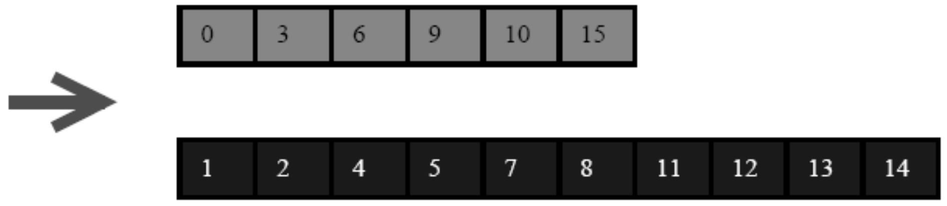
Techniques for error robustness in the source

- Flexible Macroblock Ordering (FMO)
  - MBs are scrambled in time, so that MBs that are originally adjacent are not transmitted consecutively in time
    - If a group of MBs is lost or arrives with errors, it will be easier to recover that information by using MBs that were transmitted in another instant in time but that were originally co-located in time, therefore having the probability of being similar
- Arbitrary Slice Ordering (ASO)
  - Groups of slices are sent in an arbitrary order (defined, out-of-order), different from the normal scanning order
- Data Partitioning (DP)
  - separates elements of the coded bit stream codificado in different packets according to their importance thereby enabling "unequal error protection" (UEP)
- Redundant Slices (RS)
  - an additional representation of an image region with lower quality is sent

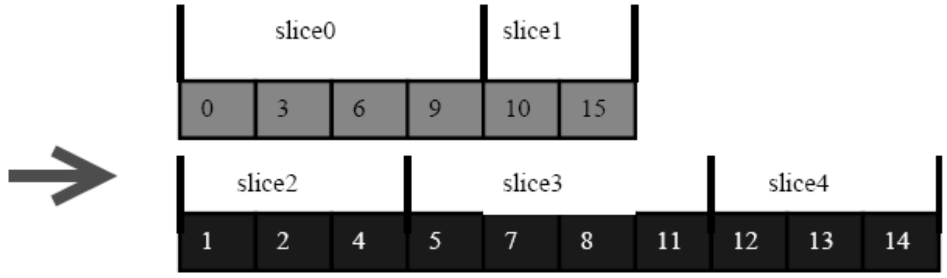
FMO and ASO - arranging MBs in “Slice Groups”



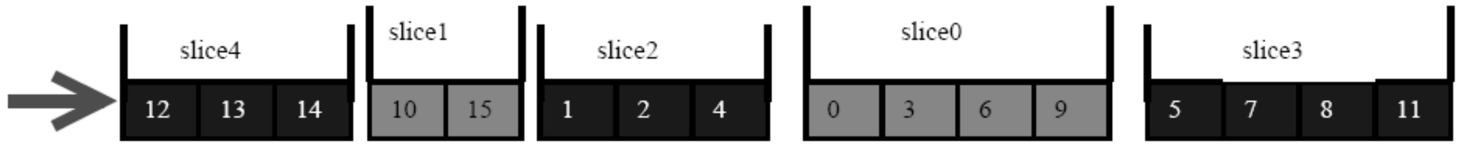
example: picture with 16 MBs and 2 slice groups 0 (grey) and 1 (black): numbers 0 and 1 are randomly assigned to MBs



MBs are grouped orderly into the slice groups according to the number assigned to them

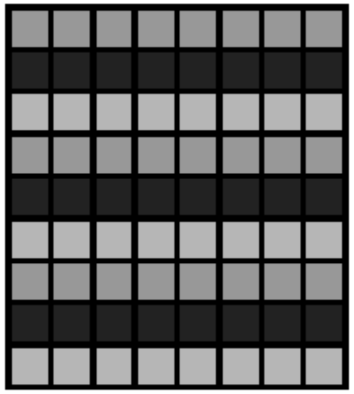


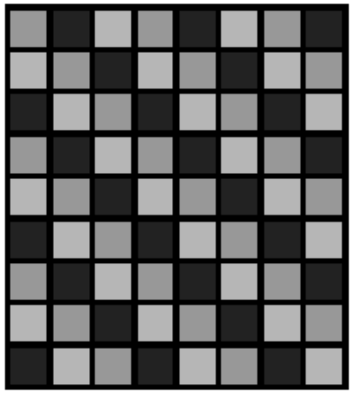
MBs in a same slice group can be divided into multiple slices

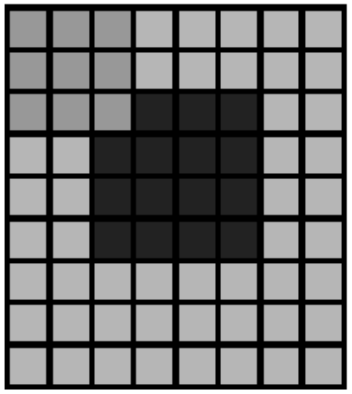


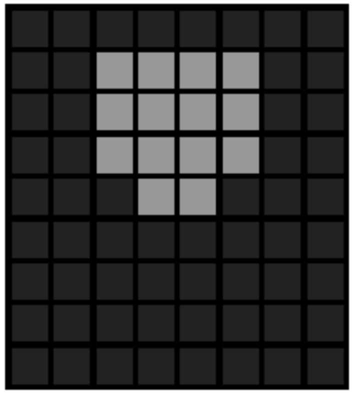
Slices then can be transmitted in an abitary order in the bitstream

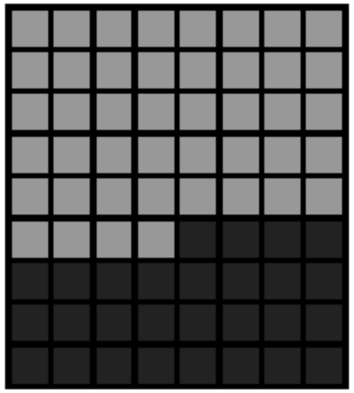
Maps for arranging groups of slices

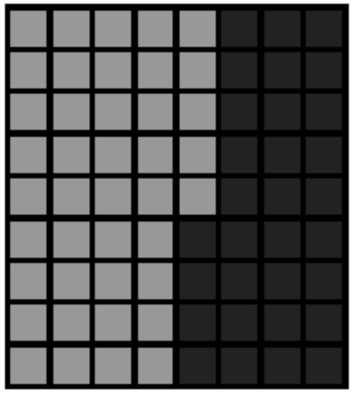

0: interleaved

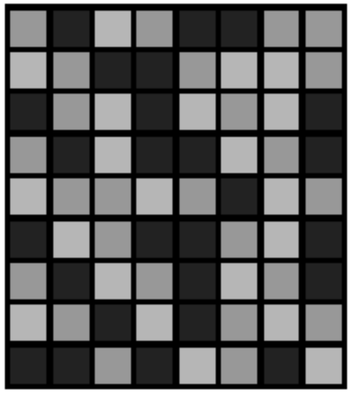

1: dispersed


2: foreground with left-over


3: box_out


4: raster-scan


5: wipe


6: expicit

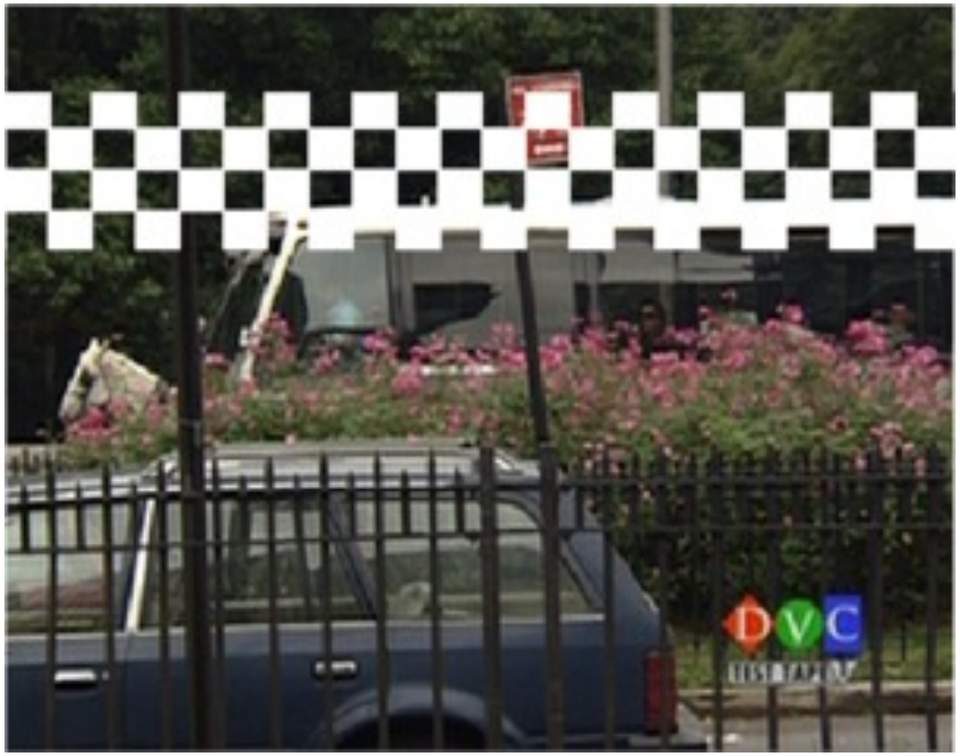
▨ MBs in slice group 0

■ MBs in slice group 1

□ MBs in slice group 2

Results of robustness tools

- FMO and ASO to enable error recovery



- the distribution of data facilitates the interpolation and the recovery of the lost data

Challenges faced - errors and losses (5)

- **error resilient video coding** schemes can limit the extent of error propagation by
  - carefully designing both the predictive coding loop and the variable length coder
    - reversible VLC (RVLC)
      - exhibits both prefix and suffix properties
      - the codewords are decodable in both directions (forward and backwards)
      - if the sync word is lost, backward decoding can be performed.
  - adding redundant/duplicate information that can be used by the decoder in case of missing data
    - “Spare pictures”, redundant slices, redundant MVs, periodic synchronisation words
      - when the motion compensation reference image is lost, decoders may use a spare image that resembles the actual reference picture
      - …
- the decoding process becomes less susceptible to errors
  - erroneous or missing bits in a compressed stream will no longer have a disastrous effect in the reconstructed video quality
- the downside: encoders become less efficient in terms of compromise quality-bit rate

Bottom-line …

- The goal
    - To provide Quality of Experience (QoE)
- Solutions
    - Local solutions (as just seen - channel and source coding)
    - General, end-to-end solutions
        - Network resources management
        - Receiver buffer management
        - Adaptive Streaming

Best-effort networks and dynamic bandwidth congestion

- in best-effort shared networks, available bandwidth between two peers is usually unknown and variable
    - it depends on the number of connected users and on the volume of traffic they are generating
        - it is not possible to predict congestion periods
        - congestion leads to losses and bandwidth availability variations
    - if the server sends packets at a rate higher than the availability, packets will be lost and consequently quality will drop
    - if the server sends packets at a rate lower than the availability, then it is not maximising the use of resources or the compromise "quality-bit rate"
- Instead of best-effort, QoS may be guaranteed by the network
    - Resources are specifically assigned to a certain service/stream
    - But this is expensive and may be inefficient

Alternative for providing QoE in best effort networks - adaptive streaming

- instead of allocating dedicated network resources to mm applications
  - which may prove to make an inefficient use of network resources or not possible to be implemented
- it is the application that adapts to the dynamic conditions of the network
  - may request initial minimum QoS conditions
- **different paradigm**
  - **instead of being the network to accommodate totally the requirements of the application**
  - **it is the application that accommodates to the availability of the network**
- even though compromises can be established
  - such as allocating minimum resources to guarantee a minimum level of quality
- there are some existing standardised solutions specific for adaptive streaming
  - HLS, DASH, MMT, SRT
  - others proprietary (Microsoft Smooth Streaming, Adobe HTTP Dynamic Streaming)

HTTP adaptive streaming

- existing HTTP adaptive streaming technologies (aka HAS) share the same approach of
    - generating multiple variations of the content
    - fragmenting it in chunks
    - assigning to the client the freedom of independently download, rearrange, and decode such chunks
- it enables the distribution of the same content among several sources to support scalability, error resilience, bandwidth congestion
- it enables providing dynamic adaptation of the multimedia presentation according to various transmission parameters
- DASH is the first attempt to standardise the concept of media segmentation for dynamic HTTP streaming so as to support wider interoperability among different vendors' applications and devices

Further Topics

- HAS
  - Details on the HLS protocol
  - Details on the MPEG-DASH standard
- Details on MPEG MMT