Video coding principles

Marco Cagnazzo

MN907 Compression
Plan

Introduction

The hybrid coder
  Temporal prediction
  The Group of Pictures (GOP)
  Operational video coding

MPEG standards
  MPEG-1
  MPEG-2
  MPEG-4
Plan

Introduction
The hybrid coder
MPEG standards
Video compression principles

- Spatial redundancy
  - Images are made up of homogeneous regions
- Time redundancy
  - Successive images in a video are similar each to the other
- A video compression algorithm must exploit both kind of redundancy
Video compression principles

Spatial redundancy
Video compression principles

Time redundancy
Video compression principles

General scheme of a video coder

Input: Compression Temporelle → Compression Spatiale → Buffer

Information de mouvement
Plan

Introduction

The hybrid coder
Temporal prediction
The Group of Pictures (GOP)
Operational video coding

MPEG standards
Prediction in video coding: DPCM

- Successive images are very similar
- Prediction: $\hat{f}_{n,m,k} = \tilde{f}_{n,m,k-1}$

Current image

Error

Previous image
Conditional replenishment

Prediction:

\[
\hat{f}_{n,m,k} = \begin{cases} 
  f_{n,m,k-1} & \text{si } |f_{n,m,k} - f_{n,m,k-1}| < \gamma \\
  0 & \text{otherwise}
\end{cases}
\]

Problem:

- **Side information**: one bit per pixel
- Using blocks of pixels the SI can be reduced
Conditional replenishment

Block similarity measure:

\[ d(B_1, B_2) = \sum_{p} |B_1(p) - B_2(p)|^k \]

Si \( d(B_k^{(p)}, B_h^{(p)}) < \gamma \)

- refine: prediction error is transmitted
- skip: no bit is transmitted

Si \( d(B_k^{(p)}, B_h^{(p)}) \geq \gamma \)

- new: the block is transmitted without prediction

How to set \( \gamma \) and the block size?
Motion estimation
Motion estimation

Introduction
The hybrid coder
MPEG standards

Temporal prediction
The Group of Pictures (GOP)
Operational video coding
Motion estimation

We compare $B_k^{(p)}$ and $B_{h}^{(p+v)}$
Motion estimation

- **ME Test:**
  \[ d(v) = d(B_k^{(p)}, B_h^{(p+v)}) \]

- **Estimated vector:**
  \[ v^* = \arg\min_v d(v) \]

- **Transmitted info:**
  \[ B_k^{(p)} - B_h^{(p+v)} \]

- The decoder reconstructs the prediction of \( B_k^{(p)} \) using the motion vectors and the reference image: this is the *motion compensation*.
Motion estimation

Cost function
Several choices are possible for $d(\cdot, \cdot)$:

- **SAD (Sum of Absolute Differences)**
  
  $$d(B_1, B_2) = \sum_{n,m} |B_1(n, m) - B_2(n, m)|$$

- **SSD (Sum of Squared Differences)**
  
  $$d(B_1, B_2) = \sum_{n,m} [B_1(n, m) - B_2(n, m)]^2$$

- **ZN-SSD (Zero-mean Normalized SSD)**
  
  $$d(B_1, B_2) = \frac{\sum_{n,m} [\overline{B}_1(n, m) - \overline{B}_2(n, m)]^2}{\sum_{n,m} \overline{B}_1^2(n, m) \sum_{n,m} \overline{B}_2^2(n, m)}$$
Prediction in video coding

Motion estimation regularization

- Vectors in homogeneous areas are chaotic
- A regularization term is added

\[ J(v) = d(v) + \lambda r(v) \]

- Estimated vector:

\[ v^* = \arg \min_v J(v) \]

- \( \lambda \) defines the trade-off between fidelity and regularity
- \( r(v) \): coding cost or smooth constraint
Prediction in video coding

Examples

Reference image

Current image
Prediction in video coding

Examples

Current image

Difference image
Prediction in video coding

Estimated vectors

Rate(MV): 1668 bits – PSNR(Pred): 24.51 dB – Time: 7.6 s

Non-regularized MVF

Regularized MVF

Institut Mines-Telecom
Prediction in video coding

Estimated vectors

Non-regularized MVF

Regularized MVF
Prediction in video coding

Estimated vectors

Regularized MVF, motion-compensated image

Regularized MVF, compensation error
Prediction in video coding

Estimated vectors

- Regularized MVF, motion-compensated image
- Regularized MVF, compensation error

Motion-compensated image
MC error
Prediction in video coding

Estimated vectors

Rate(MV): 7938 bits − PSNR(Pred): 26.53 dB − Time: 26.6 s

Non-regularized MVF

Regularized MVF
Prediction in video coding

Estimated vectors

Non-regularized MVF

Regularized MVF
Estimated vectors

Motion-compensated image

Regularized MVF, motion-compensated image

Regularized MVF, compensation error
Prediction in video coding

Estimated vectors

Regularized MVF, motion-compensated image

Regularized MVF, compensation error
Prediction in video coding

Estimated vectors

Non-regularized MVF

Regularized MVF

Rate(MV): 368 bits – PSNR(Pred): 22.52 dB – Time: 2.9 s
Prediction in video coding

Estimated vectors

Non-regularized MVF

Regularized MVF
Prediction in video coding

Estimated vectors

Regularized MVF, motion-compensated image

Regularized MVF, compensation error
Prediction in video coding

Estimated vectors

Regularized MVF, motion-compensated image

Motion-compensated image

MC error

Regularized MVF, compensation error
Prediction in video coding

Examples

Reference image

Current image
Prediction in video coding

Examples

Current image

Difference image
Prediction in video coding

Estimated vectors

Non-regularized MVF

Regularized MVF

Rate(MV): 2300 bits – PSNR(Pred): 23.03 dB – Time: 7.4 s
Prediction in video coding

Estimated vectors

Non-regularized MVF

Regularized MVF

Rate(MV): 2300 bits − PSNR(Pred): 23.03 dB − Time: 7.4 s

Rate(MV): 2253 bits − PSNR(Pred): 22.85 dB − Time: 8.9 s
Prediction in video coding

Estimated vectors

Regularized MVF, motion-compensated image

Regularized MVF, compensation error
Prediction in video coding

Estimated vectors

Regularized MVF, motion-compensated image

Motion-compensated image

MC error

Regularized MVF, compensation error

Video coding principles
Motion estimation

Search strategy: complexity/effectiveness trade-off

Let $n$ be the search window side

- **Full search** method: All the $n^2$ vectors are tested
- **Cross search** method: First horizontal vectors are tested; then the vertical component is found; total, $2n$ vectors
- **Log search** method: Nine positions $\{0, \pm(2^m - 1)\}^2$ are tested; the search window is then centered on the best position and the search step is halved to $2^{m-1} - 1$ pixels. The number of tests is $\approx 8 \log_2 n$
- **Diamond search** method: Eight directions are tested, but the step is reduced only when the center position has been chosen
Motion estimation

Summary

- Very effective for video temporal prediction
- Used in virtually all video encoders
- Trade-off: precision - coding cost - complexity
- Design choices:
  - Cost function (SAD, SSD, regularization, ...)
  - Motion support (shape and size of blocks, search area, ...)
  - Search strategy (Full-search, Log, Diamond, ...)

Very effective for video temporal prediction

Used in virtually all video encoders

Trade-off: precision - coding cost - complexity

Design choices:

- Cost function (SAD, SSD, regularization, ...)
- Motion support (shape and size of blocks, search area, ...)
- Search strategy (Full-search, Log, Diamond, ...)
Frame types

- Frames I (Intra coded)
- Frames P (Predictive)
- Frames B (Bi-directional)

I and P Frames: Anchor Frames (AF)
Group of Pictures

- Frames organized into GOP (Group of Pictures)
- First image: Intra
- Structure:
  - interval between I frames
  - interval between AFs
I Frames

- Encoded independently from others
- JPEG-like coding
- Low complexity, low coding rate
- Used for:
  - Fast forwards
  - Random access
  - Error robustness
P Frames

- Prediction from previous AF
- High Complexity (ME)
- High compression ratio
B Frames

- Predicted from both previous and next AF
- Very high complexity (double ME)
- Very high compression ratio
Frame coding order

I → AF → Frames B → AF → Frames B . . .
Delay?

Details

Base

Gop n

Gop n+1

Institut Mines-Telecom

Video coding principles
The hybrid video encoder

- Macroblock-based coding
- Coding modes
  - **Intra**: No temporal prediction, transform-based coding
  - **Inter**: ME/MC-based temporal prediction, transform coding
  - **Direct**: Motion vector inferred from neighbors; no residual coding
- Lossless
The hybrid video encoder

Coding performance examples

Distortion vs. Rate

- Direct
- ME/MC Inter16
- CR16
- ME/MC Inter8
- CR8
- Intra
- Lossless
The hybrid video encoder

Coding mode selection

- Goal: minimizing $D$ for a given $R$:

\[ D = \sum_{k=1}^{K} D_k(i_k, Q) \quad R = \sum_{k=1}^{K} R_k(i_k, Q) \]

- The quantization step is given $Q$

- The set of modes $i = \{i_k\}_{k=1}^{K}$ must be chosen such that we minimize:

\[ J(i, Q, \lambda) = \sum_{k=1}^{K} D_k(i_k, Q) + \lambda \sum_{k=1}^{K} R_k(i_k, Q) \]
The hybrid video encoder

Coding mode selection

- Joint minimization over $i$ is way too complex
- A sub-optimal minimization is preferable
- For a MB $k$, we choose the mode such that we minimize:

$$J_k(i_k, Q, \lambda) = D_k(i_k, Q) + \lambda R_k(i_k, Q)$$

- That is, we minimize *separately* each term of the sum giving $J$
- The selected mode depends on $Q$ and $\lambda$
The hybrid video encoder

Coding mode selection

- quantization step $Q$ is considered as an input
- For each $Q$ (rate) there exists an optimal $\lambda$ value, which is determined empirically
  - MPEG-2: $\lambda = aQ^2 + b$
  - H.264: $\lambda = c2^{dQ+e}$
- With this $\lambda$, minimizing $J_k$ amounts to find a line in the RD plane
The hybrid video encoder

Example of performance

\[ D + \lambda_2 R = J_2 \]
\[ D + \lambda_1 R = J_1 \]
The hybrid video encoder

\[ f_k \oplus e_k \rightarrow \text{DCT} \rightarrow \text{Q} \rightarrow \text{VLC} \]
The hybrid video encoder
The hybrid video encoder

\[ f_k \rightarrow e_k \rightarrow \text{DCT} \rightarrow \text{Q} \rightarrow \text{VLC} \rightarrow \text{Q}^* \rightarrow \text{IDCT} \rightarrow \tilde{e}_k \]
The hybrid video encoder

Introduction
The hybrid coder
MPEG standards
Temporal prediction
The Group of Pictures (GOP)
Operational video coding

The hybrid video encoder

JPEG coder

\[ f_k \rightarrow e_k \rightarrow \text{DCT} \rightarrow Q \rightarrow \text{VLC} \]

\[ Q^* \rightarrow \text{IDCT} \rightarrow \tilde{e}_k \rightarrow \text{Frame Buffer} \]

\[ \tilde{f}_k \rightarrow \hat{f}_k \]
The hybrid video encoder

- Introduction
- The hybrid coder
- MPEG standards
- Temporal prediction
- The Group of Pictures (GOP)
- Operational video coding

**Diagram:**

- **JPEG coder**
- **DCT**
- **Q**
- **VLC**
- **Q**
- **IDCT**
- **MC**
- **Frame Buffer**

Symbols:
- $f_k$
- $e_k$
- $\hat{f}_k$
- $\tilde{e}_k$
- $\tilde{f}_k$
The hybrid video encoder

- Introduction
- The hybrid coder
- MPEG standards
- Temporal prediction
- The Group of Pictures (GOP)
- Operational video coding

Diagram:

- ME
- DCT
- Q
- VLC
- IDCT
- Q'
- Frame Buffer
- MV
- MC
- f_k
- e_k
- \( \hat{f}_k \)
- \( \hat{\epsilon}_k \)
- \( \tilde{f}_k \)
The hybrid video encoder

**Introduction**

- The hybrid coder
- MPEG standards

**Operational video coding**

- ME
- DCT
- Q
- VLC
- Control
- Channel Buffer
- IDCT
- Frame Buffer
- MC
- JPEG coder
The hybrid decoder

Asymmetrical scheme!
Plan

Introduction

The hybrid coder

MPEG standards
  MPEG-1
  MPEG-2
  MPEG-4
Video standards

Timeline:
- 1980: H.261
- 1990: H.263, H.263++, MPEG-2, H.262
- 2010: HEVC, 3D-VC

Standards:
- MPEG-1
- MPEG-2
- MPEG-4
- H.262
- H.264
- H.263++
- H.264/SVC
- H.264/MVC
- 3D-VC
- HEVC

Organizations:
- ITU-T
- ISO/IEC
- Joint
- The standard only defines the bitstream syntax and the decoder behavior
- Goal: interoperability, competition
## MPEG-1 standard

- Developed in 1988-1992
- Parts
  1. Systems
  2. Video
  3. Audio
  4. Conformance test
  5. Software simulation
MPEG-1 standard

Part 2 (Video)

- Hybrid coder with ME/MC
- Input: max $720 \times 576$ pixel @ 30 fps
- Rate $\leq 1.86$ Mbps (VHS quality)
- Asymmetric applications: VoD, video CD, videogames

Features

- Image types
- Sub-pixel ME/MC
Standard MPEG-1

Sub-pixel ME/MC

- Physical motion does not necessarily correspond to pixel grid
- Interpolation to improve precision
- Further complexity increase
- Rate-distortion improvement
Standard MPEG-1

Sub-pixel ME/MC

$v=(5.5,2.5)$

$v=(5,3)$
MPEG-2 standard

- Developed in 1990-1994
- Parts
  1. Systems
  2. Video
  3. Audio
  4. Conformance test
  5. Software simulation
MPEG-2 standard

- Hybrid coder
- Rate \( \leq 15 \) Mbps (HDTV)
- Profiles and levels
- Interlaced video support
- Scalability support
# MPEG-2 standard

## Profiles and levels

<table>
<thead>
<tr>
<th>Level</th>
<th>width [pixel]</th>
<th>height [pixel]</th>
<th>frame rate [frame/s]</th>
<th>bit rate [Mbps]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>352</td>
<td>288</td>
<td>30</td>
<td>4</td>
</tr>
<tr>
<td>Main</td>
<td>720</td>
<td>576</td>
<td>30</td>
<td>15</td>
</tr>
<tr>
<td>High-1440</td>
<td>1440</td>
<td>1152</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>High</td>
<td>1920</td>
<td>1152</td>
<td>60</td>
<td>80</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Profile</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>No scalability; interlaced video; no B-frames</td>
</tr>
<tr>
<td>Main</td>
<td>Simple + B-frames</td>
</tr>
<tr>
<td>SNR scalable</td>
<td>Main + Two or three quality scalability levels</td>
</tr>
<tr>
<td>Spatial scalable</td>
<td>SNR + Two or three resolution scalability levels</td>
</tr>
<tr>
<td>High</td>
<td>Space + Oversampled chroma</td>
</tr>
</tbody>
</table>
**MPEG-2 standard**

Profiles and levels

<table>
<thead>
<tr>
<th>Level</th>
<th>Simple</th>
<th>Main</th>
<th>SNR scalable</th>
<th>Spatial scalable</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Main</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td>•</td>
</tr>
<tr>
<td>High-1440</td>
<td>•</td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>High</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td>•</td>
</tr>
</tbody>
</table>
MPEG-2 standard

Scalability

*Encode once, decode many!*

- Bitstream made up of:
  - One *base* layer
  - One or more *enhancement* layers
- The base layer can be decoded alone
- The enhancement improves quality or resolution...
- ... but cannot be decoded alone
- A client may demand the base layer only or base+enhancement
Scalability

- Courbe limite
- Codeur scalable souhaité
- Codeur non scalable, un seul layer à 200 kbps
- Codeur non scalable, un seul layer à 900 kbps
- Codeur scalable, deux layers à 400 et 800 kbps
Scalability: example

Video distribution without scalability
Scalability: example

Video distribution with scalability
MPEG-2 standard

SNR scalability: encoder

- DCT coefficients refinement
- *Drift* of the base layer
- Good quality of the enhanced layer
MPEG-2 standard

SNR scalability: decoder

- No drift control
- The same MVF is used at both layers
MPEG-2 standard

Resolution scalability: encoder

\[ x - e_E \rightarrow \text{DCT} \rightarrow Q_E \rightarrow \text{VLC} \]

\[ e_B - \rightarrow \text{DCT} \rightarrow Q_B \rightarrow \text{VLC} \]

\[ W \rightarrow \text{MC} \rightarrow \text{Frame Buffer} \]

\[ \hat{f}_B \rightarrow \text{IDCT} \rightarrow \text{Frame Buffer} \]

\[ \hat{f}_E \rightarrow \text{IDCT} \rightarrow \text{Frame Buffer} \]
MPEG-2 standard

Resolution scalability: encoder

- Double loop: no drift
- Input video is filtered and subsampled
- Enhanced level prediction is a weighted sum of:
  - The interpolated base-layer image
  - ME/MC prediction
- The weight is changed per-MB, and its value is encoded in the bitstream
MPEG-2 standard

Resolution scalability: decoder

\[ \hat{f}_E = Q_E \cdot B + \hat{f}_E \]

\[ \hat{f}_B = Q_B \cdot E + \hat{f}_B \]

\[ \text{Frame Buffer} \]
MPEG-2 standard

Time scalability

![Diagram showing the concept of time scalability in MPEG-2 standard]
MPEG-2 standard

Hybrid scalability, 1/2

- SNR + spatial
  1. SDTV/CIF, low quality
  2. HDTV/SDTV, low quality
  3. HDTV/SDTV, high quality
MPEG-2 standard

Hybrid scalability, 2/2

- spatial + temporal
  1. SDTV interlaced
  2. HDTV interlaced
  3. HDTV progressive

- SNR + temporal
  1. HDTV interlaced, low quality
  2. HDTV interlaced, high quality
  3. HDTV progressive, high quality
Le standard MPEG-4

- Developed in 1993-1998
- Parts
  - 5 main parts (as MPEG-1 et 2)
  - 18 additional parts
  - E.g. MPEG4/part 10 is H.264/AVC
Standard MPEG-4

Features

- Hybrid coder
- Interactivity
  - Bitstream manipulation without transcoding
  - Hybrid coding of natural and synthetic data
  - Improved random access
- Compression
  - Improved RD performance
- Universal access
  - Error robustness
  - Object-based scalability
Standard MPEG-4

Object-based representation

- Audiovisual object (AVO)
  - Several AVOs encoded in different bitstreams
  - Audio (mono, stereo, synthetic, ... ) and/or video part (natural, synthetic, ... )
- Several AVOs make a AV scene
- MPEG-4 defines syntax scene description
Standard MPEG-4

Audiovisual scene

Synthetic BG

Still Image

Audio object

Visual object

AV scene

AV object

Audiovisual scene

MPEG standards

Introduction

The hybrid coder

MPEG-1

MPEG-2

MPEG-4

Audiovisual scene
Standard MPEG-4

Visual coding

Video object coding
Mesh object coding
Model-based coding
Still texture coding
Video object coding

A *video object* (VO) is the succession of *video object planes* (VOP), made up of:

- Motion
- Texture
- Contours (Shape)
Standard MPEG-4

Sprite coding
Standard MPEG-4

Scalability

- Frame-rate and resolution: as MPEG-2
- Quality: *fine grain scalability* (bit-plane coding)
- Object scalability: the scene can be composed with a subset of available objects