Some Principles Guiding the Design of Video Compression Algorithms

Yuriy A. Reznik
RealNetworks, Inc.

Outline

- Introduction
- Some facts from Human Visual System
- Some information-theoretic principles and building blocks of Video Compression algorithms
  - Encoding of simple stochastic processes
  - Prediction-based coding and Motion compensation
  - Quantization
  - Transform-based coding
- Q&A

2

Introduction

- Video Coding is a relatively new and amazingly interesting field for research;

- It is interdisciplinary in nature: based on facts from physics, cognitive psychology, neuro-science, statistics (information theory), and computer science;

- Nothing (with only few exceptions) is written on stones: extremely fast pace of the development; today's state of the art becomes obsolete tomorrow.
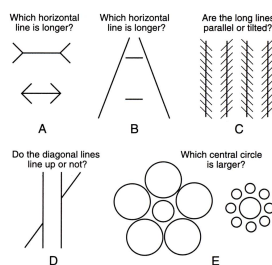
3

Some Facts from Human Visual System

- Perception of video is a very sophisticated neuro-biological process
  - first important results are obtained by Helmholtz (1850s)
  - many theories were proposed (structuralism, gelstaltism, ecological optics, etc.)
  - still an area of active research
- Few things that are useful for video coding:
  - Mach Bands
  - Spatial Sensitivity Thresholds
  - Mechanism of Color Vision
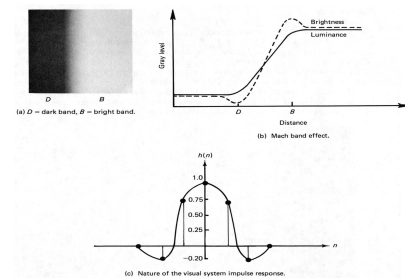 - Limits of Motion Perception

4

Visual Illusions

Which horizontal line is longer?   Which horizontal line is longer?   Are the long lines parallel or tilted?

A   B   C

Do the diagonal lines line up or not?   Which central circle is larger?

D   E

Visual illusions. Although they do not appear to be so, the two arrow shafts are the same length in A, the horizontal lines are identical in B, the long lines are vertical in C, the diagonal lines are collinear in D, and the middle circles are equal in size in E.
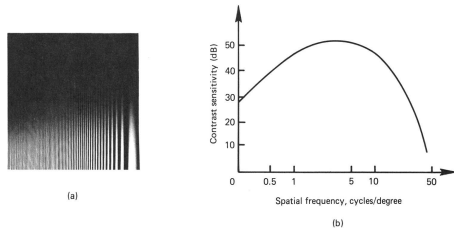
5

Mach Band Effect

(a) D = dark band, B = bright band.

(b) Mach band effect.

(c) Nature of the visual system impulse response.

6

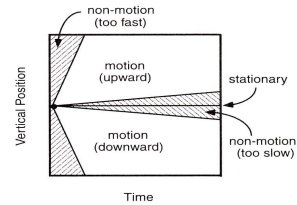## Spatial Sensitivity Thresholds

(a)

MTF of the human visual system. (a) Contrast versus spatial frequency sinusoidal grating; (b) typical MTF plot.
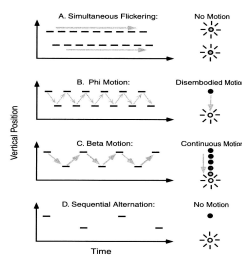
7

Perceiving Motion

The limits of motion perception. Observers experience motion as long as the object's motion is neither too slow (too shallow in space-time) nor too fast (too steep in space-time).

8

Wertheimer's Experiment (1912)

Phenomena of apparent motion. As the alternation rate between two stationary lights changes from very fast to very slow, observers perceive two simultaneously flickering lights (A), disembodied motion without intermediate position, called *phi motion* (B), continuous motion with intermediate positions, called *beta motion* (C), and sequential alternation between two unmoving lights (D).

9

Information-theoretic principles and techniques used in Video Codecs

- Things to follow:
  - Codes for simple stochastic processes
  - Prediction-based coding and Motion compensation
  - Quantization
  - Transform-based coding

10

Codes for simple stochastic sources

- Bernoulli (memoryless) source *S*:

  $A = \{a_1, ..., a_m\}$ - alphabet

  $P = \{p_1, ..., p_m\}, 0 \le p_i \le 1, \sum p_i = 1$ - probabilities of its symbols

  $h = -\sum p_i \log p_i$ - entropy of this source

- Binary prefix code for *S:*

  $f : S \to B \subset \{0,1\}^*$ (*B* is decipherable*)*

- Example:

  $A = \{a,b,c\}; P = \{1/2, 1/3, 1/6\}$

  $h = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{3}\log\frac{1}{3} - \frac{1}{6}\log\frac{1}{6} \approx 1.459$

  Code: $B = \{0, 01, 11\}$

  Average code length:

  $C = \frac{1}{2} \cdot 1 + \frac{1}{3} \cdot 2 + \frac{1}{6} \cdot 2 = 1.5$
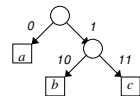
11

Redundancy of (simple prefix) Codes:

- Shannon source coding Theorem:

  $\forall f : R(f,S) \ge 0, \; R(f,S) = C(f,S) - h$

- Classic codes for Bernoulli source:
  - Shannon code $f_S : \; R(f_S, S) \le 1$ - simple
  - Gilbert-Moore code $f_{GM} : \; R(f_{GM}, S) \le 2$ - order preserving
  - Huffman code $f_H : \; R(f_H, S) < 1$ - optimal

- Problem:
  - How to improve the performance of Huffman encoding when the cardinality of source alphabet is small?

12

Block Codes for Bernoulli Source

- Input string: $S_i^A = a_{i_1} a_{i_2} ... a_{i_k}, a_{i_{k+1}} a_{i_{k+2}} ... a_{i_{2k}}, a_{i_{2k+1}} a_{i_{2k+2}} ... a_{i_{3k}} ...$
- New input string: $S_i^C = c_{i_1} c_{i_2} c_{i_3} ...$
- New source S^n:
  - $C = A^n$ - alphabet
  - $\Pr(c_i) = p_1^{r_1(c_i)} ... p_m^{r_m(c_i)}$ - probabilities of its symbols, where
    - $r_j(c_i)$ is the number of symbols a_j in c_i
  - $h(S^n) = n\,h(S)$ - entropy of this source
- Redundancy rate of Huffman block code:
  $$R(f_H, S^n) < \frac{1}{n}$$
- Simple recipe for good compression:
  - make n (block size) large!

13

What happens if we do not know probabilities of (Bernoulli) Source?

- Possible solutions:
  - count frequencies on the fly – too slow
  - use fast adaptive algorithms (LZ, move-to-front, splay-trees, etc) – less efficient
  - use universal codes

14

Universal Block Codes

- Invented by B. Fitingof in 1965 (!), complete characterization is given by L. Davisson in 1973.
- Simple (Lynch-Davisson) universal code:
  - consider n-symbols sample x from a source over $A = \{0,1\}$
  - let $r_1(x)$ be the number of symbols '1' in x;
  - the code consist of a $\lceil \log n \rceil$-bit prefix transmitting $r_1(x)$, and a $\left\lceil \log \binom{n}{r_1(x)} \right\rceil$-bit suffix transmitting the position of x in a group of $\binom{n}{r_1(x)}$ n-symbols strings with $r_1(x)$ '1's.
- Redundancy rates of universal block codes:
  $$R(f_U, S^n) = O\left(\frac{\log n}{n}\right)$$

15

Universal Codes for Monotonic Sources

- Monotonic (memoryless) source S:
  - $A = \{a_1, ..., a_m\}$ - alphabet
  - $1 > p_1 \geq p_2 \geq ... \geq p_m > 0$ - the only known property of its probabilities
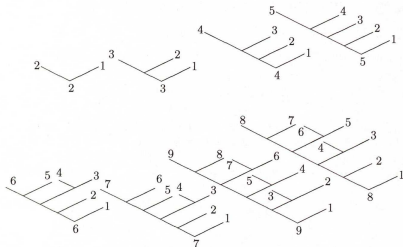- B. Ryabko, 1979: There exists a code, such that for **any** m-ary monotonic source S:
  $$R(f_M, S) = O(\log \log m)$$
- For block monotonic codes:
  $$R(f_M, S^n) = O\left(\frac{\log \log n}{n}\right)$$

16

Examples of Universal Monotonic Codes

Cases when $m = 2...9$:

17

Some Other Types of Sources:

- Fixed-order Markov sources
  - can be decomposed in a system of Bernoulli sources (main challenge is the amount of memory needed to maintain the states of the encoders)
  - there exists a universal code (V.Trofimov, 1974)
- Unknown order Markov sources – can be handled using twice-universal codes (B. Ryabko, 1984)
- Finite memory tree sources – can be handled using context tree weighting (CTW) technique (F.Willems, Yu.Shtarkov, Tj.Tjalkens, 1995)

18

3

Prediction-based Coding

- Consider a sequence of symbols $X = x_1 x_2 x_3 \ldots,\ x_i \in A$ produced by a stochastic source.
- If we can find a predictor: $y : A^n \to A$ (n – order of the predictor) such that a sequence of the residual values $\Delta = \delta_1 \delta_2 \delta_3 \ldots$ where $\delta_t = x_t - y(x_{t-n-1}, \ldots, x_{t-1})$ are mutually independent and $|\delta_k|$ has (at least) a monotonic distribution,
- Then, we arrive at the following coding scheme:
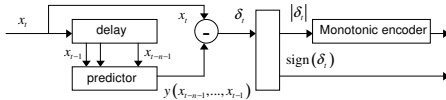
19

Linear Prediction

- We are trying to construct a predictor in a form $y(x_{t-n-1}, \ldots, x_{t-1}) = \sum_{i=1}^{n} a_i x_{t-i}$, and we need to find coefficients a_i.
- Our goal is to make sure that $\delta_t = x_t - y(x_{t-n-1}, \ldots, x_{t-1})$ are independent from $x_{t-n-1}, \ldots, x_{t-1}$
- So (at least) we must require $E(\delta_t x_{t-i}) = 0,\ i = 1, \ldots, n$
- Hence: $E(\delta_t x_{t-i}) = E\left(\left(x_t - \sum a_j x_{t-j} \right) x_{t-i} \right) = E(x_t x_{t-i}) - \sum a_j E(x_{t-j} x_{t-i})$
$$= E(x_t X) - R_X A = 0$$
where $A = [a_1 \ldots a_n]^T$, $X = [x_{t-1} \ldots x_{t-n-1}]^T$, $R_X = E\{X X^T\}$ (covariance)
- Which yields a solution:
$$A = E(x_t X) R_X^{-1}$$
- Such a prediction technique is called Linear Prediction

20

Motion Compensation

- Same idea as in predictor-based encoder (n=1)
- Block-based MC:
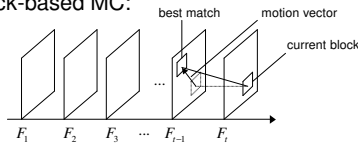
- To be encoded:
  - motion vectors
  - residual information in predicted blocks

21

More Accurate Motion Compensation

- Multiple-frame based (n>=2)
  - polynomial motion models
- Tracking shape invariants:
  - translations (done now)
  - rotations
  - dilations
- Shape-based (v.s. block-based)
- 3D-model based

22

Rate-Distortion Function

- Consider two (correlated) Bernoulli sources S and $\hat{S}$;
- The Mutual Information between S and $\hat{S}$ is given by:
$$I(S; \hat{S}) = \sum_i \sum_j \Pr(a_i, \hat{a}_j) \log \frac{\Pr(a_i, \hat{a}_j)}{\Pr(a_i) \Pr(\hat{a}_j)}$$
- The average distortion of S when presented by $\hat{S}$:
$$d(S; \hat{S}) = \sum_i \sum_j \Pr(a_i) \Pr(\hat{a}_j \mid a_i) d(a_i, \hat{a}_j)$$
- The Information Rate Distortion Function:
$$R^{(I)}(D) = \min_{\Pr(\hat{a}_j \mid a_i),\ d(S; \hat{S}) \le D} I(S; \hat{S});$$
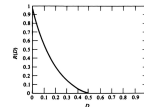- Shannon RD Theorem:
$$\forall f :\ R(f, S, D) \ge R^{(I)}(D);$$
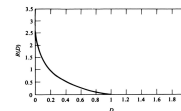
23

Examples of Simple RD Functions

- Binary Bernoulli source and Hamming distortion:
$$R(D) = \begin{cases} h(p) - h(D), & 0 \le D \le \min(p, 1-p), \\ 0, & D > \min(p, 1-p). \end{cases}$$


- Gaussian Source and square error distortion:
$$R(D) = \begin{cases} \dfrac{1}{2} \log \dfrac{\sigma^2}{D}, & 0 \le D \le \sigma^2, \\ 0, & D > \sigma^2. \end{cases}$$

24

Quantization

- For memoryless sources the problem of finding an optimal quantizer can be solved numerically using Lloyd algorithm (1957)
- Modifications:
  - Block-based (vector) quantization
  - Entropy constrained
- Video codecs typically use simple linear quantizers, sometimes in combination with compandors:

$$x \rightarrow \boxed{G} \rightarrow \boxed{\text{UNIFORM QUANTIZER}} \rightarrow \boxed{G^{-1}} \rightarrow \hat{x}$$
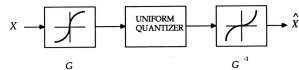
25

Transform-Based Coding

- Consider a sample vector *x* produced by a stationary process X (with finite second moments)

  $R_x = E\{xx^T\}$ - autocorrelation matrix

- We want to find a matrix T (transform), such that the vector

  $y = Tx$ - is decorrelated.

- In other words, we are looking for solution of:

$$R_y = E\{yy^T\} = E\{Txx^TT^T\} = TR_xT^T = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix}$$

26

Karhunen-Loeve Transform:

- The last equation has an immediate solution in the form: $T = [u_i ... u_n]$

- Where u_i are the eigen vectors of the covariance matrix: $R_x u_i = \lambda_i u_i$

- This is a well-known Hotelling (or Karhunen-Loeve) transform.

- Problems:
  - how to obtain the covariance matrix?
  - what if it cannot be easily estimated?

27

KLT Approximations

- In a case of a Markov-1 process with transitional probability *p*, Ahmed and Flickner (1982) have established convergence (when $p \rightarrow 1$) of KLT to
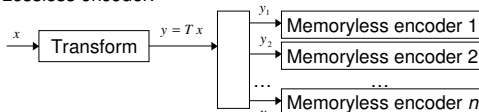
$$t_{uv} = \sqrt{\frac{2}{n}} k_u \cos\left(\frac{\pi u (v+1/2)}{n}\right); k_u = \begin{bmatrix} \frac{1}{\sqrt{2}}, & u=0 \\ 1, & u \geq 1 \end{bmatrix}; u,v = 0,...n-1;$$

- which is a DCT-II transform.

- There have been reported several other asymptotic results (e.g. convergence to DFT) when *n* is large.

28

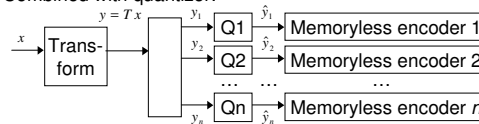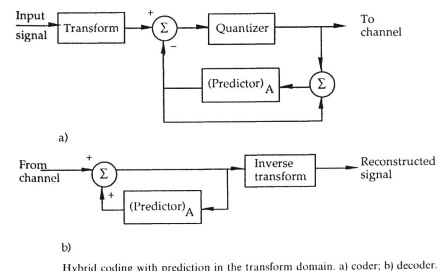## Simple Transform-Based Encoders

Lossless encoder:

$$x \rightarrow \boxed{\text{Transform}} \xrightarrow{y = Tx} \begin{cases} y_1 \rightarrow \boxed{\text{Memoryless encoder 1}} \\ y_2 \rightarrow \boxed{\text{Memoryless encoder 2}} \\ \quad\vdots \\ y_n \rightarrow \boxed{\text{Memoryless encoder } n} \end{cases}$$

Combined with quantizer:

$$x \rightarrow \boxed{\text{Trans-form}} \xrightarrow{y = Tx} \begin{cases} y_1 \rightarrow \boxed{Q1} \xrightarrow{\hat{y}_1} \boxed{\text{Memoryless encoder 1}} \\ y_2 \rightarrow \boxed{Q2} \xrightarrow{\hat{y}_2} \boxed{\text{Memoryless encoder 2}} \\ \quad\vdots \\ y_n \rightarrow \boxed{Qn} \xrightarrow{\hat{y}_n} \boxed{\text{Memoryless encoder } n} \end{cases}$$

29

Hybrid Encoder with Prediction in Transform Domain

a)

b)

Hybrid coding with prediction in the transform domain. a) coder; b) decoder.
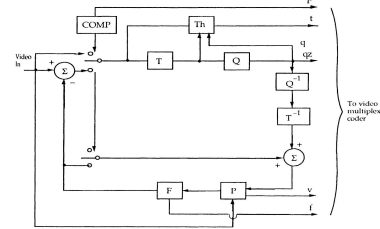
30

Hybrid Encoder with Prediction in Spatial or Temporal Domain
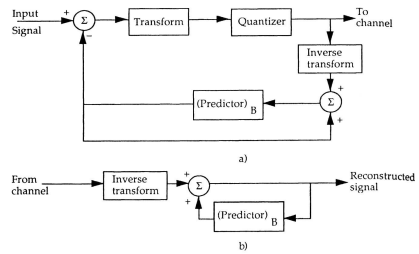
Hybrid coding with prediction in the spatial or temporal domain.

Hybrid Transform/DPCM Video Encoder (H.26*, MPEG1-4)

Hybrid transform/DPCM encoder [LBR-37, LBR-40]. The transform and its inverse refer to the 2D 8×8 DCT. COMP = comparator for intra/inter; Th = threshold; T = transform; Q = quantizer; P = picture memory with motion-compensated variable delay; F = loop filter; p = flag for intra/inter; t = flag for transmitted or not; q = quantizing index for transform coefficients; qz = quantizer indication; v = motion vector; f = switching on/off of the loop filter.

Questions?