Introduction to Video Processing & Applications

1

1

A Brief History

- Born of Television (1920s)
- Cable TV system (1968)
- Video games (1970s)
- All-digital HDTV (1990s)
- Video streaming (2000s)
- Everyday video transmission through internet and wireless networks (20??)

2

2

Why Video?

- The magic of Tele-Vision
  - Our vision capability is extended in **space**

You don't need to travel to north pole to watch polar bears

3

3

Why Video? (Cont'd)

- Our vision capability is extended in **time**
  - If time can be reversed, I will not need a Gigabyte hard-drive to store the moments of how a baby is growing
- The fundamental interplay between **time** and **motion**
  - We measure time by the motion of material things
  - Motion offers a new horizon for us to understand the world

4

4

Importance of Motion

- Our HVS routinely perceives and interprets motion (neurobiology)
- Functional MRI (fMRI)
  - By measuring the increase in blood flow to the local vasculature that accompanies neural activity in the brain, fMRI studies brain function instead of anatomy
- Gait-based biometrics
  - The characteristics of an individual's walk

5

5

Diversity of Motion






6

6

Motion in Video

- It is not an arbitrary concatenation of images, but a sequence of images carrying a *coherent* interpretation of natural scene
  - Ordering is important
  - Sampling rate is important
  - The role of a single frame is less important due to the masking effect of HVS

7

7

How to Understand Video?

- Understand the source
  - How to model the motion of a camera? (relatively easy)
  - How to model the motion in the real world? (notoriously difficult)
- Understand the mechanism of time-varying image formation model
  - Two sides: geometric and photometric

8

8

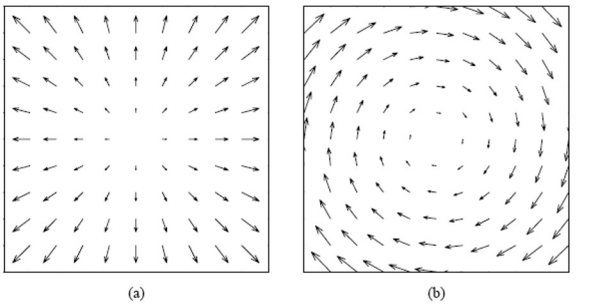
Camera Motion

- How many scene changes?
- Within each scene, what kind of camera motion do you see?
  - camera panning
  - zoom in/out
  - combination

9

9

2-D Motion Corresponding to Camera Motion



(a) Camera zoom

(b) Camera rotation around Z-axis (roll)

10

10

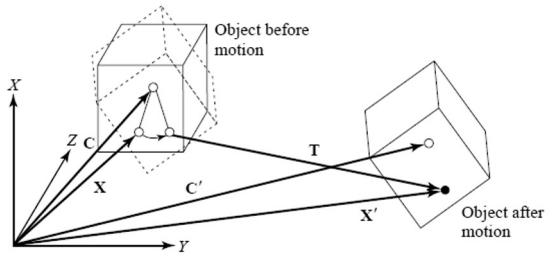
Real-world Motion

- Every motion you observe for a day
  - Can you classify them into a few simple classes?
    - Rigid motion vs. deformable motion
  - IF you observe multiple motions at the same time, how about the spatial relationship among different moving objects?
    - Overlapping vs. non-overlapping

11

11

Rigid Object Motion



Rotation and translation wrp. the object center :

$$\mathbf{X}'=[\mathbf{R}](\mathbf{X}-\mathbf{C})+\mathbf{T}+\mathbf{C};\quad [\mathbf{R}]:\theta_x,\theta_y,\theta_z;\quad \mathbf{T}:T_x,T_y,T_z$$

12

12

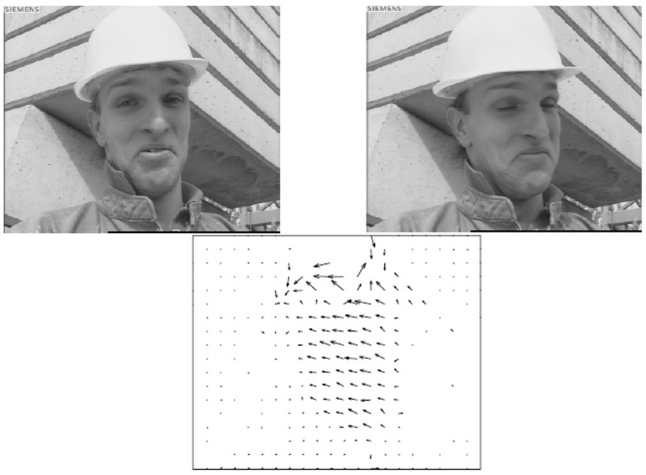
Flexible Object Motion

- Two ways to describe
  - Decompose into multiple but connected rigid sub-objects
  - Global motion plus local motion in sub-objects
  - Ex. Human body consists of many parts each undergo a rigid motion

13

13

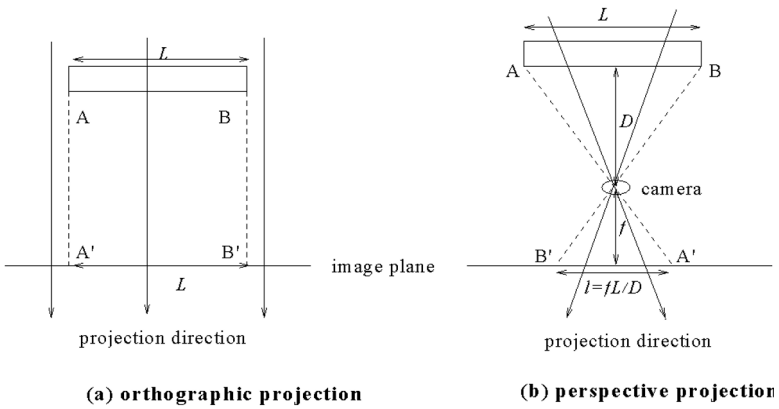
An Example



14

14

Geometric Image Formation Models



(a) orthographic projection (b) perspective projection

15

15

Photometric Image Formation Models

- Modeling surface reflectance function
- Modeling illumination condition
  - Light source location and intensity
- Modeling the photometric impact of 3D motion

16

16

Photometric Image Formation Models

- For natural images we need a light source *(λ: wavelength of the source)*
  - $E(x, y, z, \lambda)$: incident light on a point *(x, y, z world coordinates of the point)*
- Each point in the scene has a reflectivity function.
  - $r(x, y, z, \lambda)$: reflectivity function
- Light reflects from a point and the reflected light is captured by an imaging device.
  - $c(x, y, z, \lambda) = E(x, y, z, \lambda) \times r(x, y, z, \lambda)$: reflected light.

$\rightarrow E(x, y, z, \lambda)$

$\rightarrow c(x, y, z, \lambda) = E(x, y, z, \lambda) \cdot r(x, y, z, \lambda)$

$\text{Camera}(c(x, y, z, \lambda)) =$ 

17

17

Why Video is Hard?

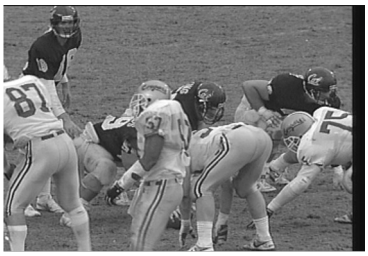
- The daunting modeling complexity
  - Scene geometry, lighting condition, object/camera motion, sensor characteristics
- We have to rely on digital computers to process video
  - Limited memory and computation resource
  - Fundamental question about computing

18

18

Example: 2D Motion Estimation

1st frame

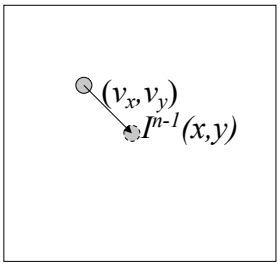


2nd frame

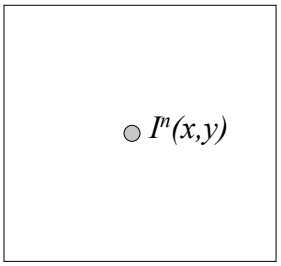
19

19

Fundamental Assumption



the *n-1*-th frame



the *n*-th frame

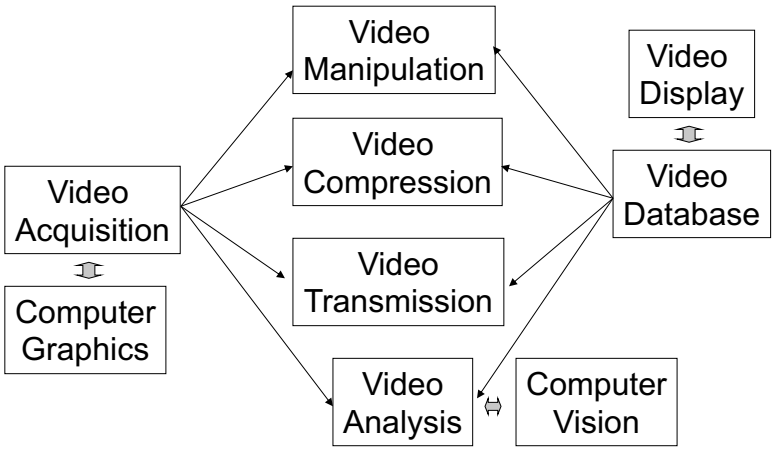
Image intensity field is *smooth* along the motion trajectory

$$I^n(x,y) = I^{n-1}(x - v_x, y - v_y)$$

20

20

Overview of Video Processing



21

21

Video Acquisition

Video camera

VHS digitization

computer-generated

22

22

Acquisition-related Problems

- Video camera
  - What if camera is not kept still?
  - Why is it difficult to improve the spatial resolution of video cameras?
- VHS digitization
  - What if VHS contains some scratches?
  - How to handle interlaced video?
- Computer-generated
  - How is this type of video different? Shouldn't we have a separate coding algorithm for this type of video?

23

23

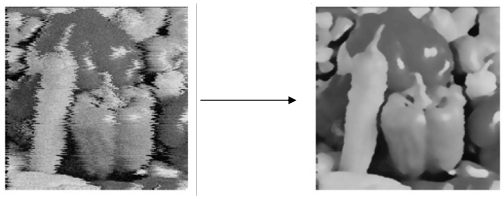
Video Manipulation

- Why?
  - Fight against a non-ideal video acquisition (e.g., analog heritage, film scratches, limited resolution) or transmission environment
  - Create new and artificial video content (e.g., spatio-temporal interpolation, background/foreground modification)

24

24

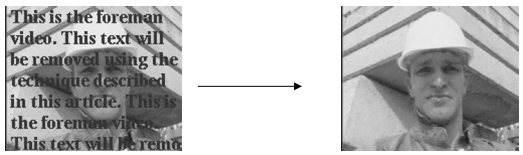
Video Dejittering



PDE-based approach by Jackie Shen

http://epubs.siam.org/sam-bin/dbq/article/41869

Video Inpainting



Cool application: remove the annoying texts added by various video conversion software

Error Concealment

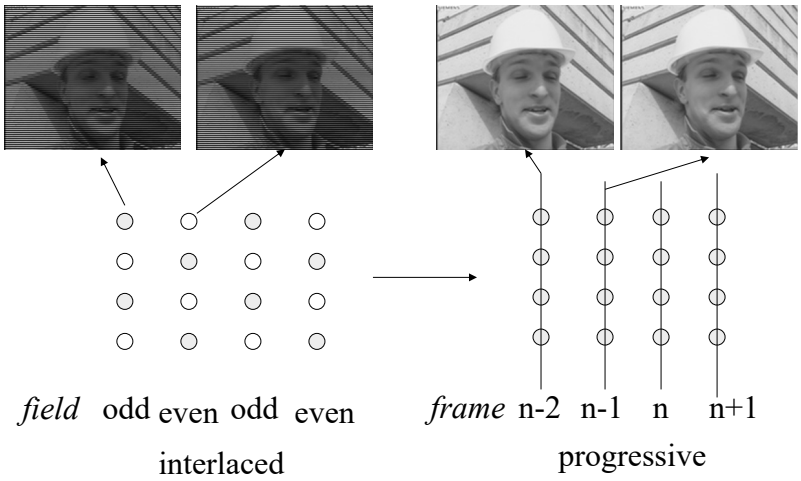
some blocks are corrupted
due to channel errors

corrupted blocks are recovered
From surrounding neighbors
in space and time

27

27

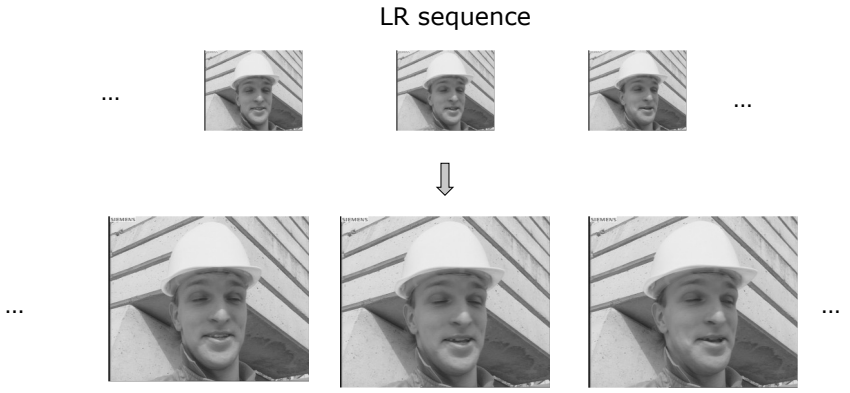
Deinterlacing



28

28

Superresolution

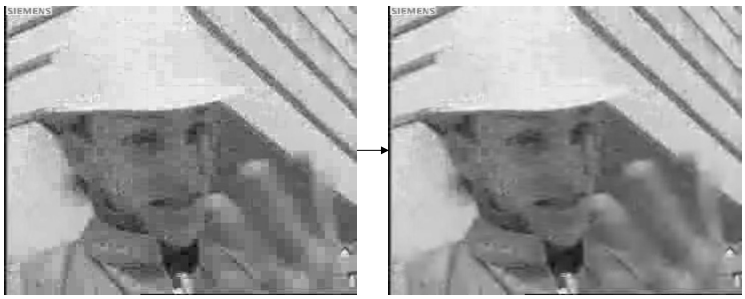


29

29

Post-processing

Deblocking: suppress block artifacts in video



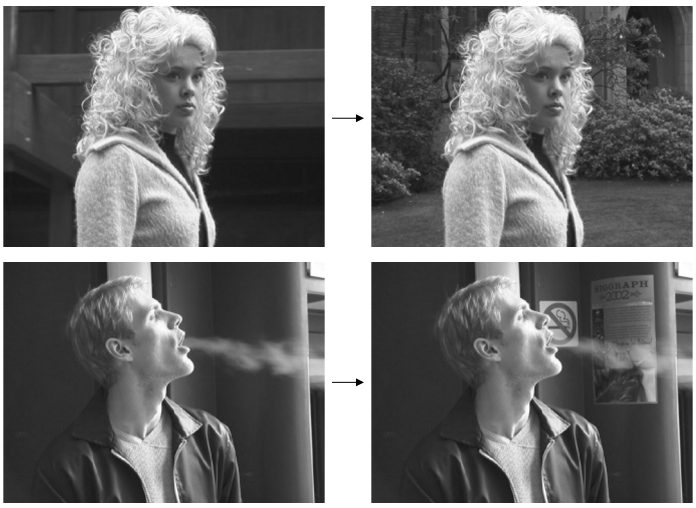
decoded video frame
at very low bit rate

processed video frame
after deblocking

30

30

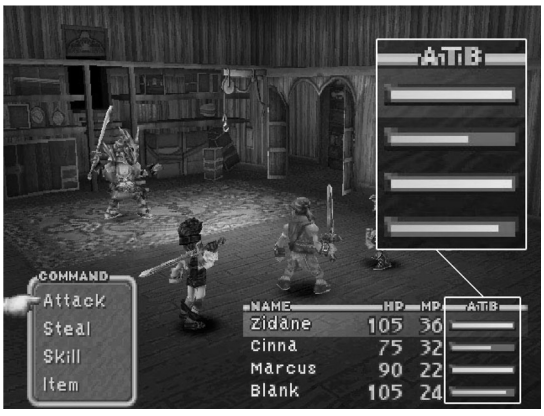
Video Matting



31

31

Video Games



32

32

Video Dynamosaics

33

33

Dynamosaics Result

Source: http://www.vision.huji.ac.il/dynmos/

34

34

Video Coding Overview

- The grand challenge
  - We still face insufficient storage space for video data even with Gigabyte hard disks
  - Video transmission through limited bandwidth channels
- Existing approaches
  - Three-dimensional waveform coding
  - Motion-compensated hybrid coding
  - Model-based coding
  - Video coding standards

35

35

Three-dimensional Waveform Coding*

- Image and video coding
  - Sub-band/wavelet coding of 2D signals
  - Wavelet works because of its good localization property in both space and frequency
  - SPHIT AND SPHIT3

http://www.cipr.rpi.edu/research/SPIHT/

36

36

Motion-compensated Predictive Coding

- Basic idea
  - DPCM coding in temporal domain
  - To reduce overhead on motion field, motion vector is assigned to each block instead of each pixel
  - After block-wise motion compensation, code motion-compensated residues like still images
- Variations: variable block size, fractional-pel accuracy, overlapped block motion compensation (OBMC)
- All existing video coding standards from H.261 to the latest H.264 fall under such category

37

37

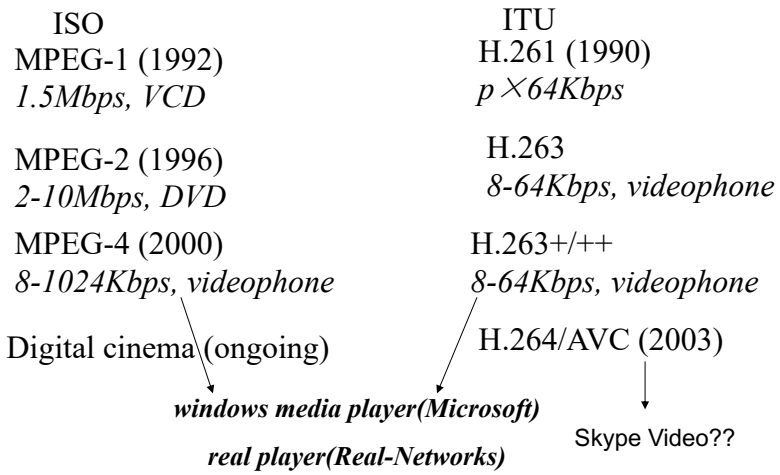
Model-based Coding

- Object-based coding
  - Attempt to replace blocks by objects
  - Its success remains uncertain due to difficulty of segmentation
- Knowledge-based coding
  - Explicitly build 3D waveframe models to represent moving objects
  - Limited success in videophone applications

38

38

Video Coding Standards



39

39

Transcoding Problem

- How to translate a piece of MPEG2 (DVD) video into WMV format?
  - Straightforward approach: decode it by MPEG2 decoder and then encoder it by MSC MPEG4 encoder
  - Transcoding approach: achieve the same goal with reduced computational cost
- When spatial or temporal resolution changes, the goal of complexity reduction becomes more difficult to achieve in transcoding

40

40

New Directions in Video Coding

- Distributed video coding for **sensor networks**
  - How to shift MC from encoder to decoder?
- Video coding for **cartoon** sequences
  - Existing techniques work terribly on them
- Video coding inspired by studies of HVS
  - You have seen the impact of motion masking
  - There also exists other properties of HVS that can be exploited

41

41

Video Transmission

- Downloading
  - Pro: you can have your own copy and can watch it offline
  - Con: you have to wait!!!
- Streaming
  - Pro: no need to store (we seldom watch a movie again and again)
  - Con: you have to have a good network connection and pray for less traffic

42

42

Video Transmission Through Networks

- Networking protocols
  - Transmission Control Protocol (TCP)
  - User Datagram Protocol (UDP)
  - Real Time Protocol (RTP) and VDP
  - Real Time Streaming Protocol (RTSP)
  - ReSerVation Protocol (RSVP)
- **Transmission Control Protocol is not suitable for video streaming** because
  - TCP imposes its own flow control and windowing schemes on the data stream, effectively destroying temporal relations between video frames
  - Reliable message delivery is unnecessary for video - losses are tolerable and TCP retransmission causes further jitter and skew.

43

43

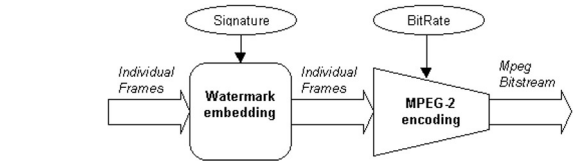
Security issues

- Video is unique
  - high data rate, power hungry, time constrained, loss-tolerant, content with varying importance
- Content access control
  - Cryptographic approaches
  - Digital video scrambling techniques
- Piracy and malicious attacks
  - Video watermarking

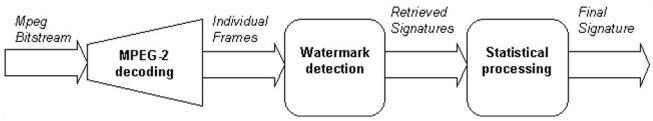
44

44

Video Content Protection by Watermarking Techniques



Signature insertion



Signature extraction

45

45

Research Ideas

- Distributed video coding for error resilience
  - Further extension of multiple descriptions
  - Motion estimation/compensation is performed at the decoder instead of encoder
- Power-constrained transmission
  - Sensor network applications and handheld devices
- Authentication in networked transmission
  - Transmission errors vs. malicious attacks
  - Transcoding distortions vs. intentional attacks

46

46

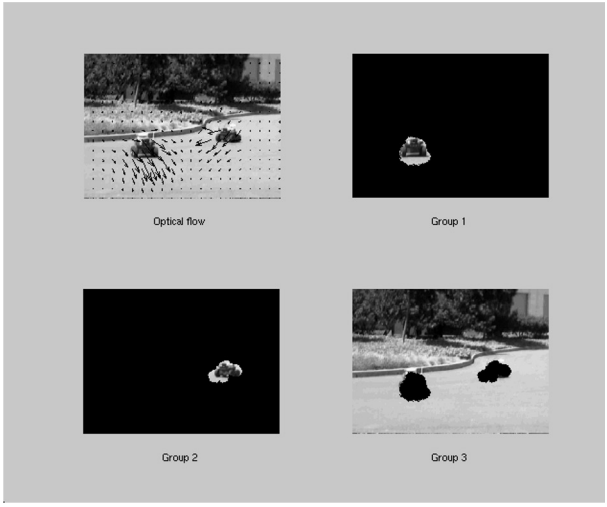
Video Analysis

- Motion segmentation
  - In contrast to image segmentation, motion offers valuable clues for separating different objects
- Motion tracking
  - Track the same object across video frames
- Motion interpretation
  - Easy for HVS, difficult for a computer (e.g., summarize a 6-hr. baseball video into 30min.)

47

47

Motion Segmentation



48

48

Motion Tracking

49

49

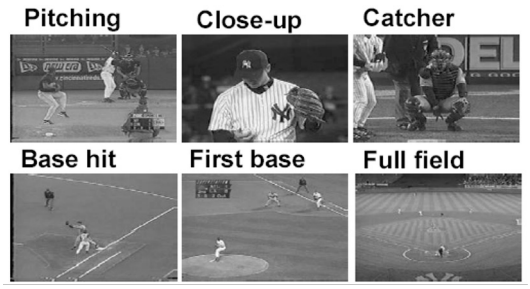
Motion Interpretation

- Scene change detection
  - Where motion tracking fails
- Cut, dissolve, wipe classification
  - Those are artificial features added by video editing staff
- Analyze each video segment
  - Camera motion: panning or zooming or still
  - Object motion: shape, direction, speed, etc.

50

50

Application (I): Video Summarization

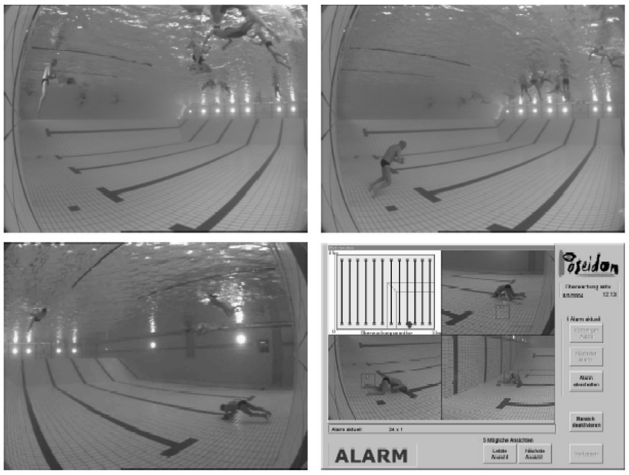


Extract “important” motion pictures such as home-runs

51

51

Application (II): Video-based Lifeguard



Application in swimming pool monitoring to prevent drowning

52

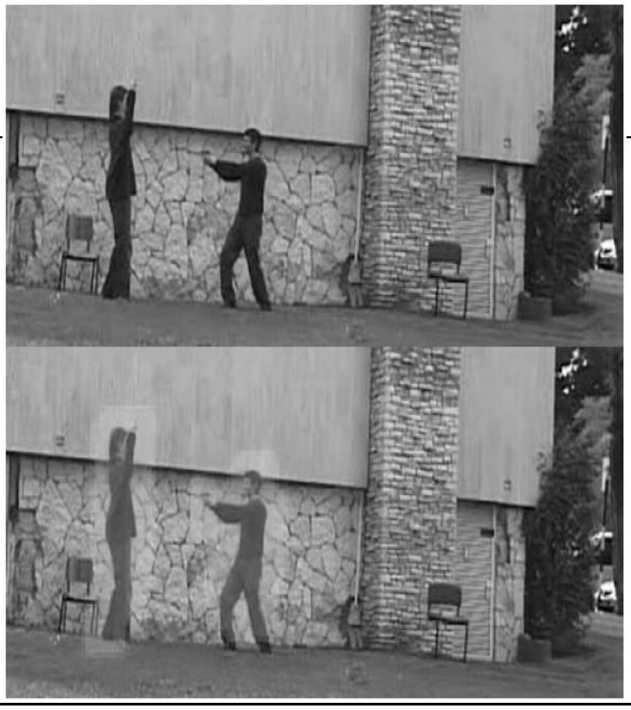
52

Application (III): Irregularity Detection

Source: http://www.wisdom.weizmann.ac.il/~vision/Irregularities.html

53

53



54

54

55

Video Database Management

- Database management
  - Indexing, parsing, browsing, querying
  - Retrieval
- What is special about video?
  - Formidable amount of data
  - Difficulty with query (content-based)
  - Inherent uncertainty and imprecision

56

56

Content-Based Video Retrieval (CBVR)

- How to provide a compact and complete video sequence representation?
  - Spatial analysis (histogram, color, texture)
  - Temporal analysis (cut, dissolve, wipe)
- How to provide easy-to-use and efficient query interface to user
  - Video browsing (slide vs. 3D)
  - Video querying (example-based, text-based)

57

57

Compressed-domain Video Analysis

- Since video data often exist in compressed format, it is preferred to do analysis with bit streams rather than pixel values
  - Examples: caption detection, shot detection etc.
- The key issue lies in how to exploit the information contained in the bit stream
  - It does not cost much computation
  - It is constrained by the adopted compression techniques and never perfect (e.g., block motion field)

58

58

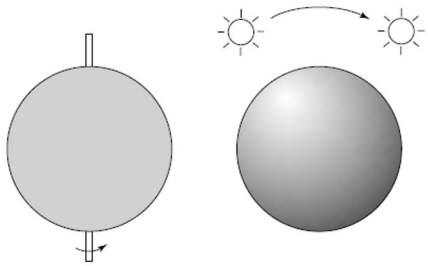
Two –Dimensional Motion Estimation

59

59

Motion vs. Optical Flow

- 2-D Motion: Projection of 3-D motion, depending on 3D object motion and projection operator
- Optical flow: "Perceived" 2-D motion based on changes in image pattern, also depends on illumination and object surface texture



On the left, a sphere is rotating under a constant ambient illumination, but the observed image does not change.

On the right, a point light source is rotating around a stationary sphere, causing the highlight point on the sphere to rotate.

60

60

General Consideration

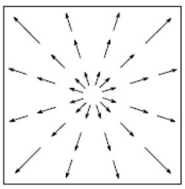
- Two categories of approaches:
  - Feature based (more often used in object tracking, 3D reconstruction from 2D)
  - Intensity based (based on constant intensity assumption) (more often used for motion compensated prediction, required in video coding, frame interpolation) -> Our focus
- Three important questions
  - How to represent the motion field?
  - What criteria to use to estimate motion parameters?
  - How to search motion parameters?

61

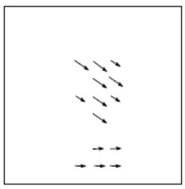
61

Motion Representation

Global:
Entire motion field is represented by a few global parameters



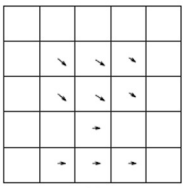
(a)



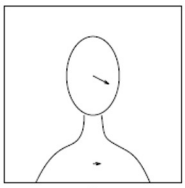
(b)

Pixel-based:
One MV at each pixel, with some smoothness constraint between adjacent MVs.

Block-based:
Entire frame is divided into blocks, and motion in each block is characterized by a few parameters.



(c)



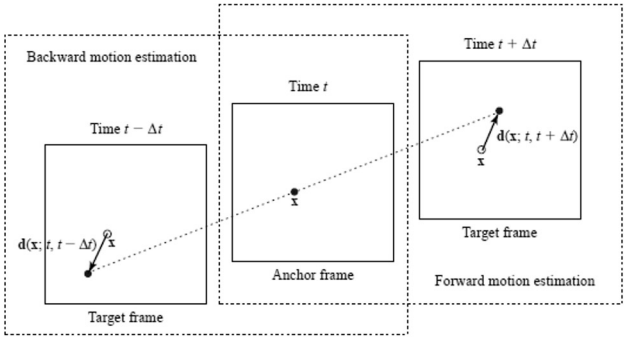
(d)

Region-based:
Entire frame is divided into regions, each region corresponding to an object or sub-object with consistent motion, represented by a few parameters.

62

62

Notations



Anchor frame: $\psi_1(\mathbf{x})$

Target frame: $\psi_2(\mathbf{x})$

Motion parameters: $\mathbf{a}$

Motion vector at a pixel in the anchor frame: $\mathbf{d}(\mathbf{x})$

Motion field: $\mathbf{d}(\mathbf{x};\mathbf{a}), \mathbf{x} \in \Lambda$

Mapping function:

$\mathbf{w}(\mathbf{x};\mathbf{a}) = \mathbf{x} + \mathbf{d}(\mathbf{x};\mathbf{a}), \mathbf{x} \in \Lambda$

63

63

Motion Estimation Criterion

- To minimize the displaced frame difference (DFD)

$$E_{\text{DFD}}(\mathbf{a}) = \sum_{x \in \Lambda} \left| \psi_2(\mathbf{x} + \mathbf{d}(\mathbf{x};\mathbf{a})) - \psi_1(\mathbf{x}) \right|^p \rightarrow \min$$

$$p = 1 : \text{MAD}; \quad P = 2 : \text{MSE}$$

- To satisfy the optical flow equation

$$E_{\text{OF}}(\mathbf{a}) = \sum_{x \in \Lambda} \left| (\nabla \psi_1(\mathbf{x}))^T \mathbf{d}(\mathbf{x};\mathbf{a}) + \psi_2(\mathbf{x}) - \psi_1(\mathbf{x}) \right|^p \rightarrow \min$$

- To impose additional smoothness constraint using regularization technique (Important in pixel- and block-based representation)

$$E_s(\mathbf{a}) = \sum_{\mathbf{x} \in \Lambda} \sum_{\mathbf{y} \in N_x} \left\| \mathbf{d}(\mathbf{x};\mathbf{a}) - \mathbf{d}(\mathbf{y};\mathbf{a}) \right\|^2$$

$$w_{DFD} E_{\text{DFD}}(\mathbf{a}) + w_s E_s(\mathbf{a}) \rightarrow \min$$

- Bayesian (MAP) criterion: to maximize the a posteriori probability

$$P(D = \mathbf{d} | \psi_2, \psi_1) \rightarrow \max$$

64

64

Optimization Methods

- Exhaustive search
  - Typically used for the DFD criterion with p=1 (MAD)
  - Guarantees reaching the global optimal
  - Computation required may be unacceptable when number of parameters to search simultaneously is large!
  - Fast search algorithms reach sub-optimal solution in shorter time
- Gradient-based search
  - Typically used for the DFD or OF criterion with p=2 (MSE)
    - the gradient can often be calculated analytically
    - When used with the OF criterion, closed-form solution may be obtained
  - Reaches the local optimal point closest to the initial solution
- Multi-resolution search
  - Search from coarse to fine resolution, faster than exhaustive search
  - Avoid being trapped into a local minimum

65

65

Block-Based Motion Estimation

- Assume all pixels in a block undergo a coherent motion, and search for the motion parameters for each block independently
- Block matching algorithm (BMA): assume translational motion, 1 MV per block (2 parameter)
  - Exhaustive BMA (EBMA)
  - Fast algorithms

66

66

Block-Matching Algorithm

- Overview:
  - Assume all pixels in a block undergo a translation, denoted by a single MV
  - Estimate the MV for each block independently, by minimizing the DFD error over this block
- Minimizing function:

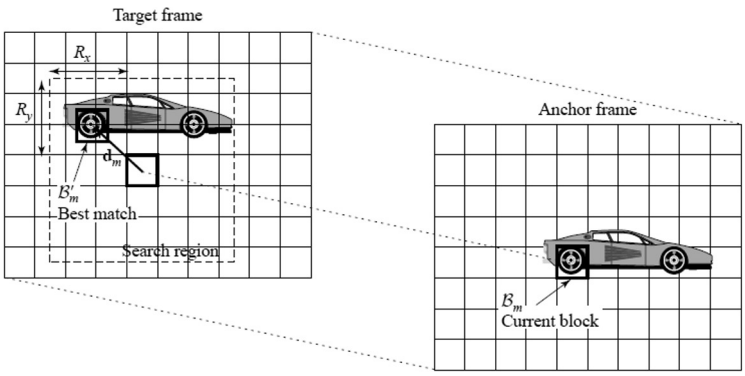
$$E_{\mathrm{DFD}}(\mathbf{d}_{\mathrm{m}}) = \sum_{\mathbf{x}\in B_m} \left|\psi_2(\mathbf{x}+\mathbf{d}_m) - \psi_1(\mathbf{x})\right|^p \rightarrow \min$$

- Optimization method:
  - Exhaustive search (feasible as one only needs to search one MV at a time), using MAD criterion (p=1)
  - Fast search algorithms
  - Integer vs. fractional pel accuracy search

67

67

Exhaustive Block Matching Algorithm (EBMA)



68

68

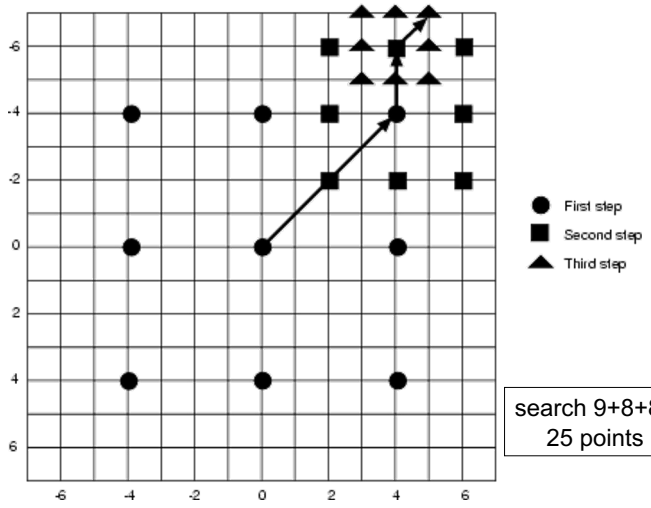
Complexity of Integer-Pel EBMA

- Assumption
  - Image size: MxM
  - Block size: NxN
  - Search range: (-R,R) in each dimension
  - Search stepsize: 1 pixel (assuming integer MV)
- Operation counts (1 operation=1 "-", 1 "+", 1 "*"):
  - Each candidate position: *N^2*
  - Each block going through all candidates: *(2R+1)^2 N^2*
  - Entire frame: *(M/N)^2 (2R+1)^2 N^2=M^2 (2R+1)^2*
    - Independent of block size!
- Example: M=512, N=16, R=16, 30 fps
  - Total operation count = 2.85x10^8/frame =8.55x10^9/second
- Regular structure suitable for VLSI implementation
- Challenging for software-only implementation

69

69

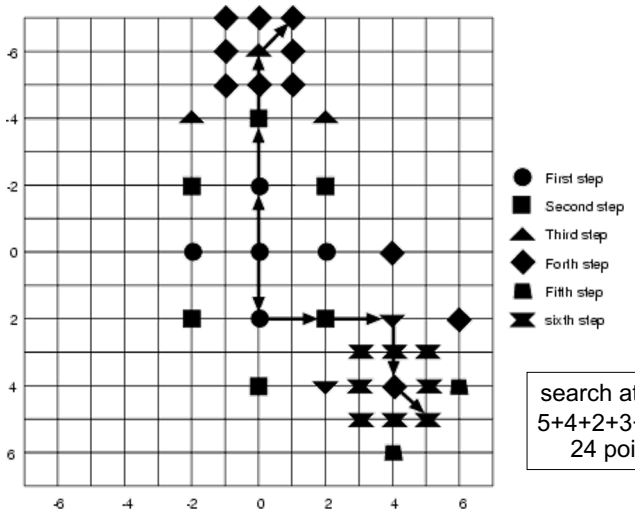
Fast BMA (1): 3-Step-Search



70

70

Fast BMA (2): 2D-Log Search

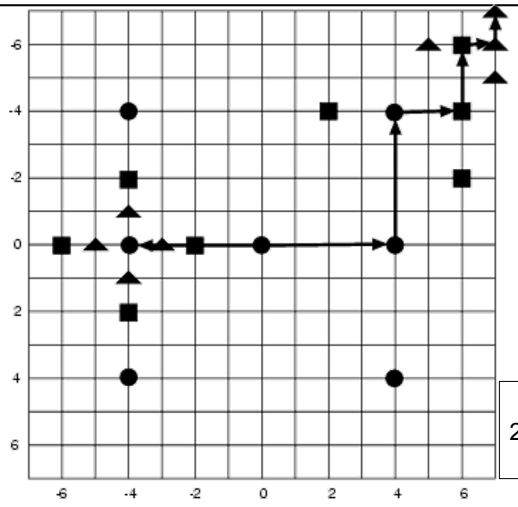


search at most
5+4+2+3+2+8=
24 points

71

71

Fast BMA (3): Orthogonal Search

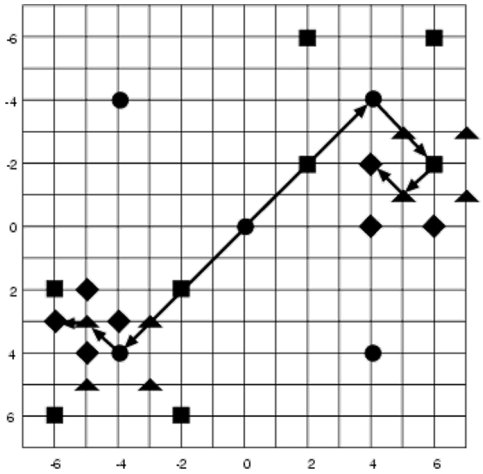


search at most
2(3+2+2+2+2+2)=
26 points

72

72

Fast BMA (4): Cross Search



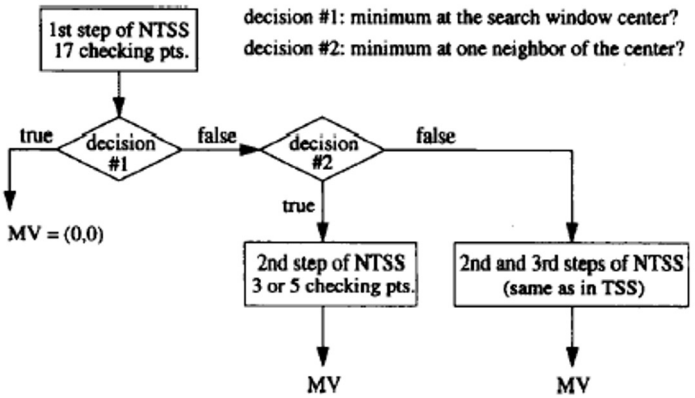
As the step size decreased to one, a (+) cross search pattern (as shown in lower-left side of figure) is used if the minimum BDM point of the previous step is either the center, upper-left or lower-right checking point. Otherwise, (X) cross search pattern (as shown in upper-right side of figure) is used.

search at most 5+4+4+4= 17 points

73

73

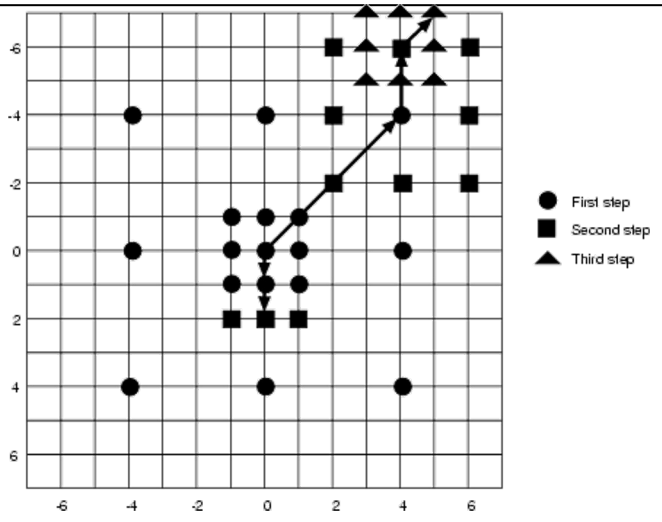
Fast BMA (5): New 3-Step Search



74

74

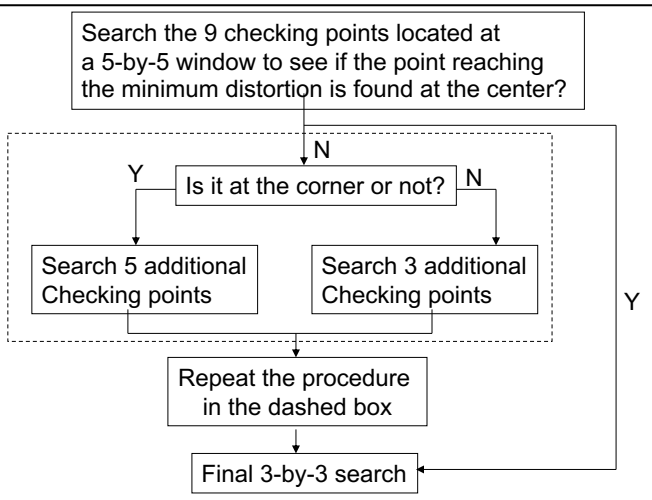
New 3-Step Search: Examples



75

75

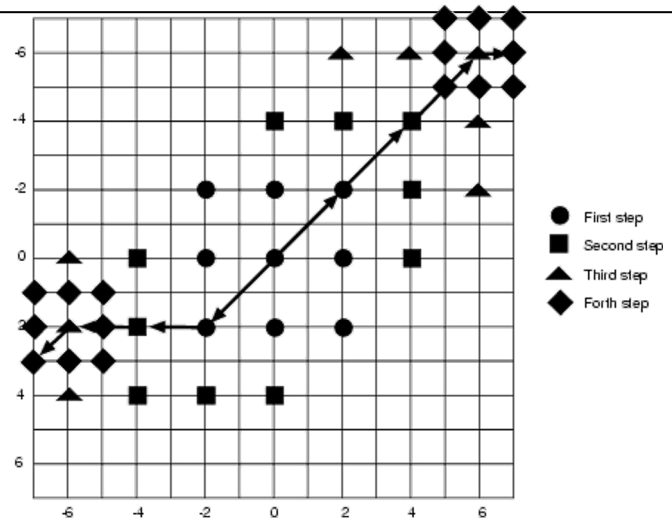
Fast BMA (6): 4-Step Search



76

76

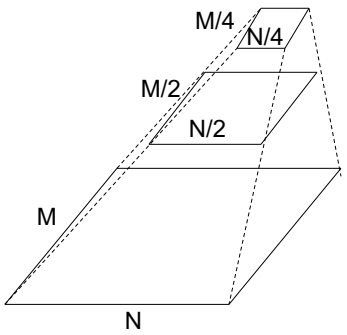
4-Step Search: Examples



77

77

Multi-resolution Representation of Images

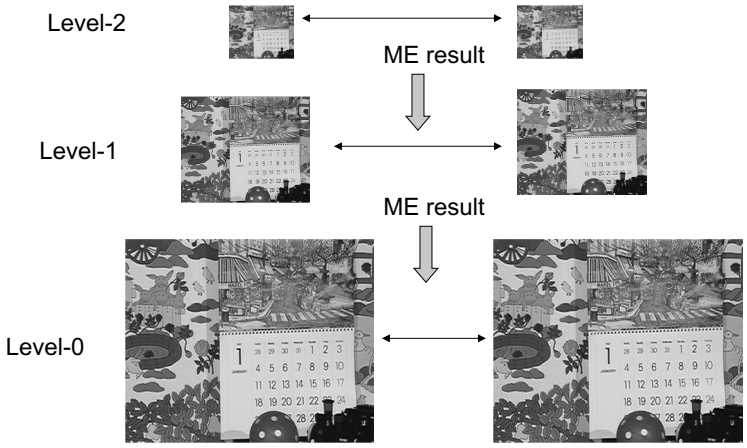


Multi-resolution representation by pyramid

78

78

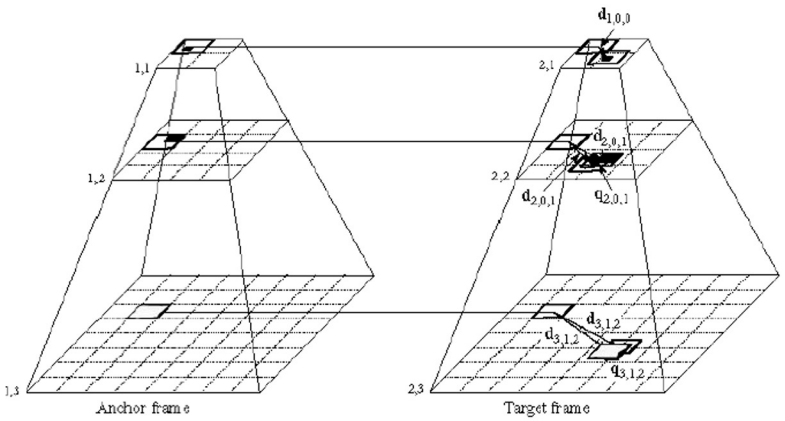
Why does Hierarchical Strategy Help?



79

79

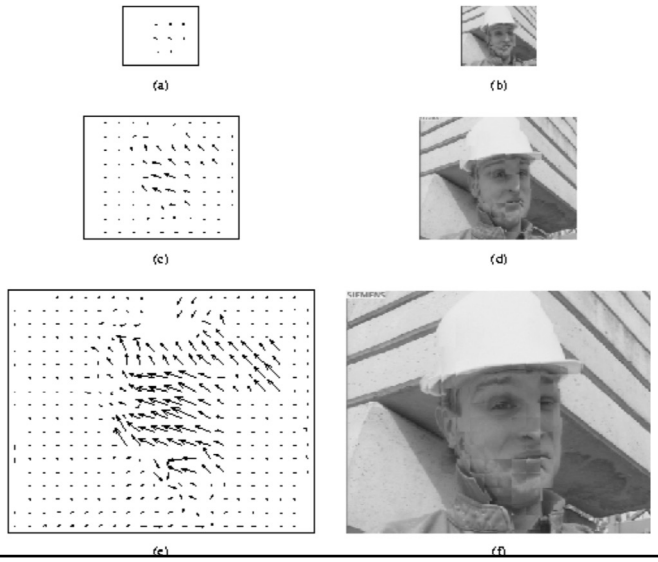
Hierarchical Block Matching Algorithm (HBMA)



80

80

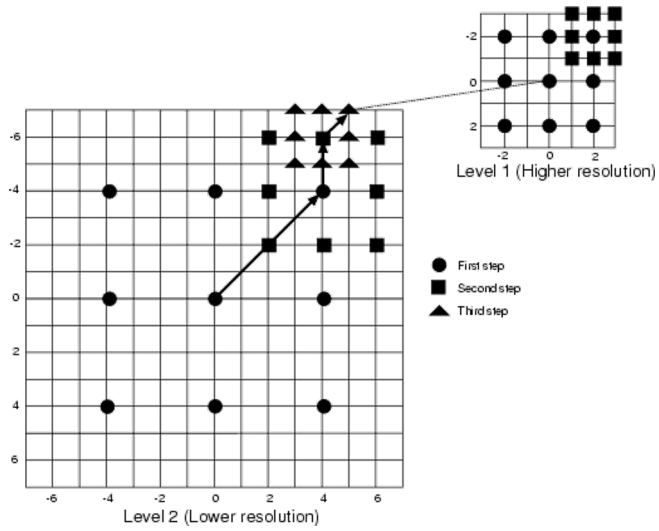
Example: Three-level HBMA



81

81

Fast BMA (7): *Hierarchical Search*



82

82

Summary

- Why do we care fast BMA?
  - Driven by the application demands of video coding
- Can we go beyond BMA?
  - The block-based constraint is simple but not appropriate for accounting for arbitrary shape of moving objects
  - The integer-pel accuracy is not sufficient to account for continuous nature of motion

83

83

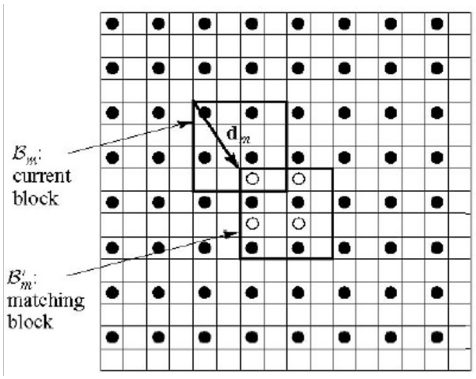
Fractional Accuracy EBMA

- Real MV may not always be multiples of pixels. To allow sub-pixel MV, the search stepsize must be less than 1 pixel
- Half-pel EBMA: stepsize=1/2 pixel in both dimension
- Difficulty:
  - Target frame only have integer pels
- Solution:
  - Interpolate the target frame by factor of two before searching
  - Bilinear interpolation is typically used
- Complexity:
  - 4 times of integer-pel, plus additional operations for interpolation.
- Fast algorithms:
  - Search in integer precisions first, then refine in a small search region in half-pel accuracy.

84

84

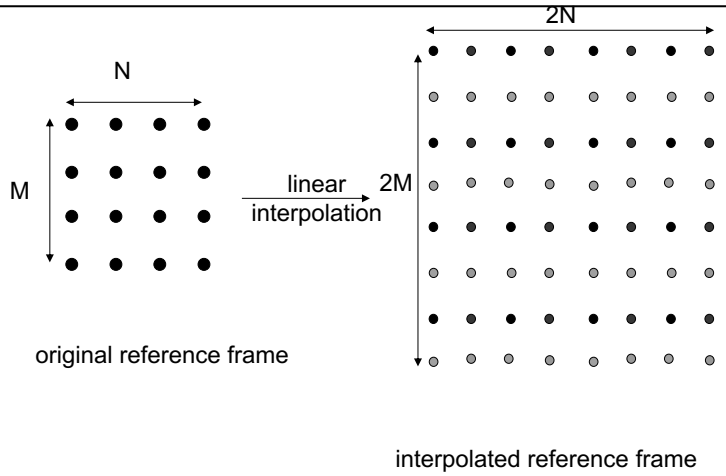
Why Do We Need Fraction-pel?



85

85

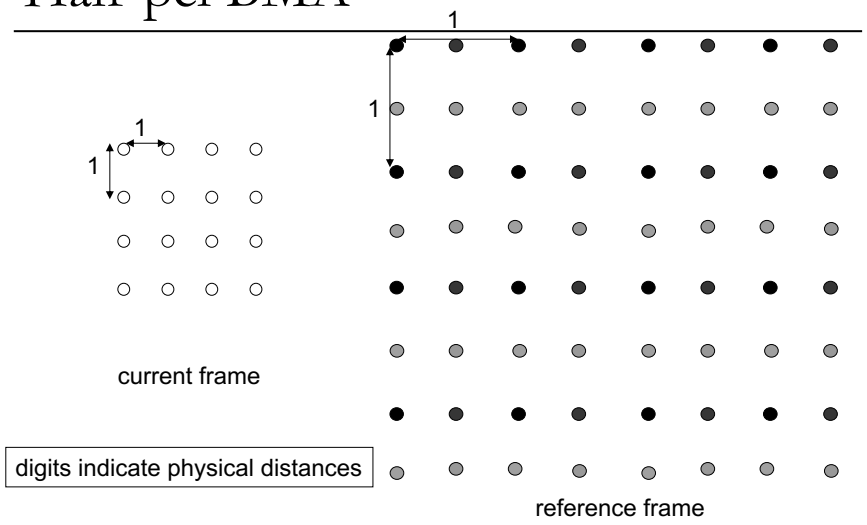
Fractional-pel BMA



86

86

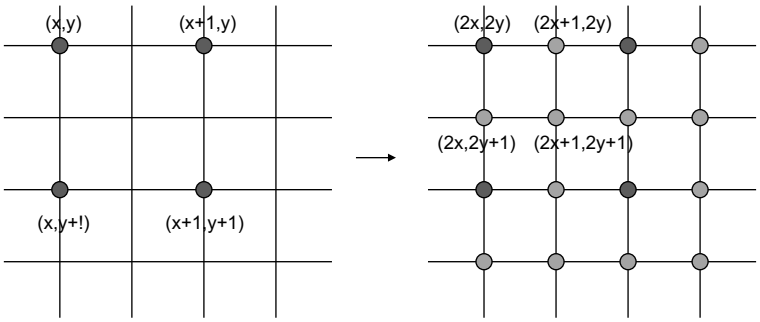
Half-pel BMA



87

87

Bilinear Interpolation

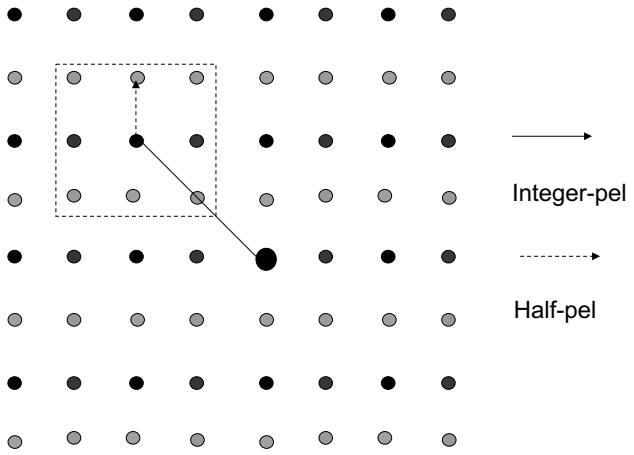


O[2x,2y]=I[x,y]
O[2x+1,2y]=(I[x,y]+I[x+1,y])/2
O[2x,2y+1]=(I[x,y]+I[x,y+1])/2
O[2x+1,2y+1]=(I[x,y]+I[x+1,y]+I[x,y+1]+I[x+1,y+1])/4

88

88

Hierarchical Strategy for Half-pel BMA



89

89

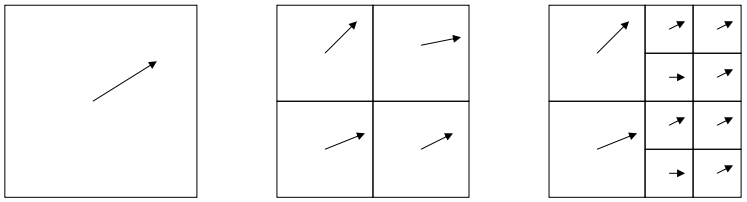
Generalizations of BMA

- Variable block-size matching algorithms
  - Widely used by various video coding standards
  - H.264 includes three variable block sizes: 4-by-4, 8-by-8 and 16-by-16
- Fractional-pel accuracy BMA
  - Half-pel : MPEG-1/2/4, H.263/H.263+
  - Quarter-pel: H.264 (even 1/8-pel)
- Tradeoff between overhead on motion and MCP efficiency

90

90

Variable Block-size BMA

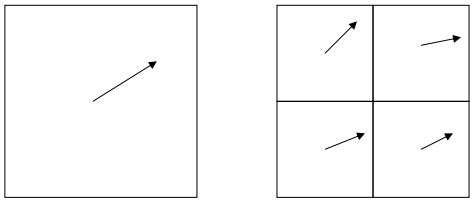


16-by-16 8-by-8 4-by-4

91

91

BMA Strategy Adopted by H.263



16-by-16

Macroblock level

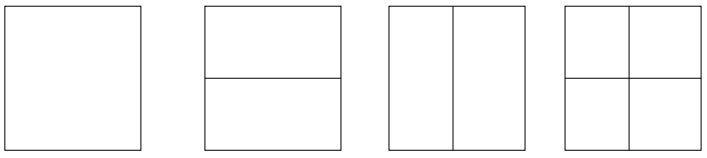
8-by-8

Block level

92

92

BMA Strategy Adopted by H.264



16-by-16 8-by-16 16-by-8 8-by-8

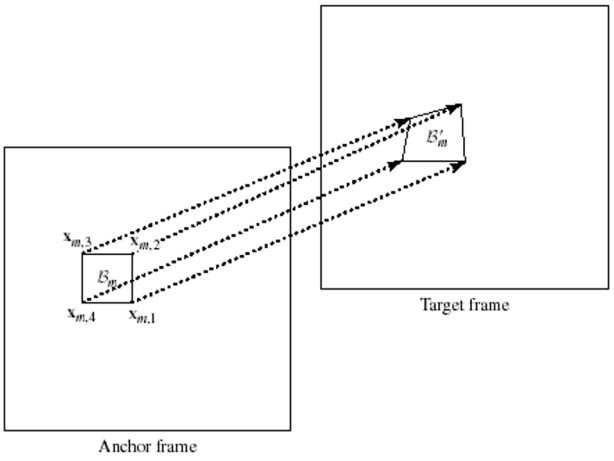
8-by-8 4-by-8 8-by-4 4-by-4

Note: require overhead to signal which partition is adopted by the encoder

93

93

Deformable Block Matching Algorithm



94

94

Overview of DBMA

- Three steps:
  - Partition the anchor frame into regular blocks
  - Model the motion in each block by a more complex motion
    - The 2-D motion caused by a flat surface patch undergoing rigid 3-D motion can be approximated well by projective mapping
    - Projective Mapping can be approximated by affine mapping and bilinear mapping
  - Estimate the motion parameters block by block independently
    - Discontinuity problem cross block boundaries still remain

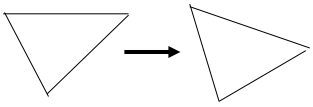
95

95

Affine and Bilinear Model

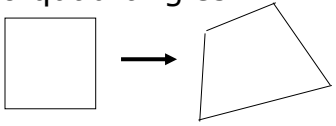
- Affine (6 parameters):
  - Good for mapping triangles to triangles

$$\begin{bmatrix} d_x(x,y) \\ d_y(x,y) \end{bmatrix} = \begin{bmatrix} a_0 + a_1 x + a_2 y \\ b_0 + b_1 x + b_2 y \end{bmatrix}$$



- Bilinear (8 parameters):
  - Good for mapping blocks to quadrangles

$$\begin{bmatrix} d_x(x,y) \\ d_y(x,y) \end{bmatrix} = \begin{bmatrix} a_0 + a_1 x + a_2 y + a_3 xy \\ b_0 + b_1 x + b_2 y + b_3 xy \end{bmatrix}$$



96

96

Mesh Based Estimation

The computation of a motion vector is affected by the neighboring vectors.

Step 1: The current frame is divided into picture elements (which may be any polygon) such that a mesh or control grid is formed .

Step 2: Then the nodes of each mesh is searched for in the previous reference frame.

Step 3: After knowing the displacement vectors of the nodes of the picture element the displacement vectors of the rest of the pixels are obtained by interpolating the known motion vectors.

97

97

Node Search Technique

1. Hierarchical mesh based matching algorithm (HMMA).
2. Hierarchical block based matching algorithm (HBMA).

In HMMA the corners of blocks are taken as nodes while in HBMA the centers of blocks are taken as nodes.

While in terms of PSNR values : The coding gain of HMMA is not significant.

But in case of prediction accuracy mesh based models tend to give more pleasing prediction, especially in the presence of non-translational motions, like rotation and turning.

So, by using HBMA we can certainly exploit lower complexity advantage of BMAs in mesh based models as well.

98

98

Mesh Based Estimation vs. BMAs

ADVANTAGES:

Mesh based models give in general a more continuous effect than BMAs .

So, in terms of prediction accuracy, mesh based models can give visually more pleasing prediction, specially in the presence non-translational motions, such as head rotation and turning.

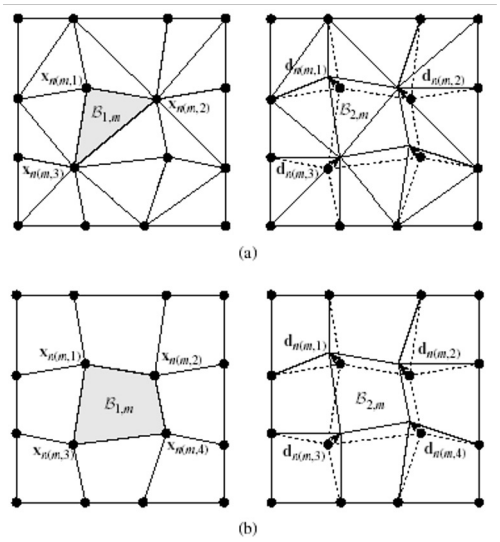
DISADVANTAGES:

While in terms of computational complexity the BMAs certainly have an edge over Mesh based ME

99

99

Mesh-Based Motion Estimation



A control grid is used to partition a frame into non-overlapping polygon elements. The nodal motion is constrained so that a feasible mesh is still formed with the motion.

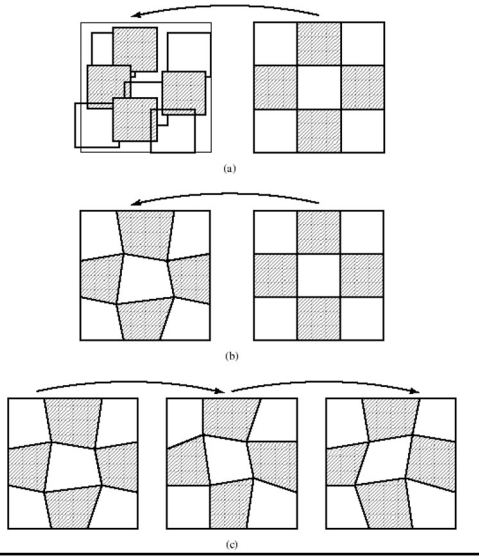
(a) Using a triangular mesh

(b) Using a quadrilateral mesh

100

100

Mesh-based vs Block-based



(a) block-based ME

(b) mesh-based ME

(c) mesh-based motion tracking

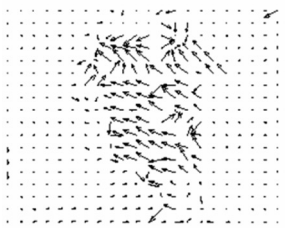
101

101

Example: BMA vs. Mesh-based

Target

Anchor



EBMA (half-pel) (29.86dB)

Predicted

Mesh-based method (29.72dB)

102

102

Experiment

Frame #1

Frame #2

103

103