Motion Estimation for Video Coding Standards

Prof. Ja-Ling Wu

Department of Computer Science and Information Engineering
National Taiwan University
Introduction of Motion Estimation

• The goal of video compression is to reduce the total transmission bit rate for reconstructing images at the receiver.
Introduction of Motion Estimation

• Hence, we encode the motion information which occupies only a small amount of the transformation bandwidth additional to the picture content information.

• Motion estimation techniques are also used in many other applications:
  – Computer vision, target tracking, industrial monitoring

• Motion estimation techniques can be roughly classified into three groups:
  – Block matching groups
  – Differential (Gradient) methods
  – Fourier Methods
Introduction of Motion Estimation

• In the physical world, objects are moving in the 4-D spatio-temporal domain (three spatial coordinates and one temporal domain). However, image video taken by a single camera is a 3-D cube (2D projections of 3-D spatial objects along time axis).

• In reality, videos taken by an ordinary camera is first sampled along the temporal axis. Typically, the temporal sampling rate (frame rate) is 24.25 or 29.97 frames/sec. Spatially, every frame is sampled vertically into a number of horizontal lines, and a line is sampled into a number of pels.

![Diagram showing 3-D cube and temporal sampling](image)
Introduction of Motion Estimation

- Motion-compensated estimation is an effective means in reducing the inter-frame correlation for image sequence coding. It is the operation of predicting an image (or portion thereof) based on displacement of a previously transmitted frame in an image sequence.

- By motion estimation, we mean the estimation of displacement (or velocity) of image structures from one frame to another in a time-sequence of 2-D images.

\[ \hat{d} = [\hat{d}_x \hat{d}_y]^T \]
Introduction of Motion Estimation

- To reduce computation and storage complexity, motion parameters of objects in a picture are estimated based on two or three nearby frames.

- General assumptions:
  1. Objects are **rigid bodies**; hence, object deformation can be neglected for at least a few nearby frames.
  2. Objects move only in **translation movement** for at least a few frames.
  3. **Illumination** is unchanged under movement.
  4. **Occlusion** of one object by another and uncovered background are neglected.

```
motion estimation → motion segmentation (identify the moving object boundaries)

motion parameter estimation
```
Definition of Moving Object

• Moving object:
  – A group of contiguous pels that share the same set of motion parameters. It does not necessarily match the ordinary meaning of the object. For example, still background can be considered as a single object.

• The effect of object size:
  – Small objects (or evaluation windows)
    • Ambiguity problem: Similar objects (image patterns) may appear at multiple locations inside a picture and may lead to incorrect displacement vectors.
    • Noise sensitivity problem: Statistically, estimation based on a small set of data are more vulnerable to random noise than those based on a large set of data.
  – Large objects (or evaluation windows)
    • Accuracy problem: Pels inside an object or evaluation window do not share the same motion parameters and, therefore, the estimated motion parameters are not accurate for some or all pels in it.
Definition of Moving Object

• Practical solution to determine object size:
  – Partition images into regular, non-overlapped blocks; assuming that moving objects can be approximated reasonably well by regular shaped blocks. Then, a single displacement vector is estimated for the entire image block under the assumption that all the pels in the block share the same displacement vector.
  – The above assumption may not always be true because an image block may contain more than one moving object. In image sequence coding, however, prediction errors due to imperfect motion compensation are coded and transmitted.
  – The block-based motion estimation approach is adopted by the video coding standards for partially, at least, its robustness as compared to pel-based approach.
Definition of Moving Object

• Practical solution to determine object size:
  – Partition images into regular, non-overlapped blocks; assuming that moving objects can be approximated reasonably well by regular shaped blocks. Then, a single displacement vector is estimated for the entire image block under the assumption that all the pels in the block share the same displacement vector.
  – The above assumption may not always be true because an image block may contain more than one moving object. In image sequence coding, however, prediction errors due to imperfect motion compensation are coded and transmitted.
  – The block-based motion estimation approach is adopted by the video coding standards for partially, at least, its robustness as compared to pel-based approach.
Motion-compensated Coding Structure

- **Motion compensation:**
  1. Access the previous-frame image data according to the estimated displacement vector.
  2. Construct the predicted pels by passing the previous-frame image through the prediction filter.
Motion-compensated Coding Structure

- H.261/263: I, P
- MPEG-1/2: I,P,B → Encoding/transmission order
- A picture frame in MPEG-2 may contain two interleaved fields:
  - Frame-based motion compensation
  - Field-based motion compensation
Motion-compensated Coding Structure

- As long as the motion parameters we obtain can efficiently reduce the total bit rate, these parameters need not to be the true motion parameters.
- If the reconstructed images are used for estimating another motion information, a rather strong noise component can not be neglected. ➜ Drift Problem
- Notice that the current video standards specify only the decoder, i.e. the encoder which performs the motion estimation operation is not explicitly specified in the standards, a great amount of flexibility exists in choosing and designing a motion estimation scheme for a standard coder.
Block Matching Method

• Block matching is a correlation technique that searches for the best match between the current image block and candidates in a confined area of the previous frame.
  – Goal:
    • Reduce the computational load in calculating the motion vector.
    • Increase the motion vector accuracy.
Block Matching Method

Current Picture

(x,y)  N  M
Macroblock

Reference Picture

(x,y)  N  M
[-p,p] Search Region

Best Match

Motion Vector
Block Matching Method

• The size of the block affects the performance of motion estimation.
  – Small block sizes afford good approximation to the natural object boundaries; they also provide good approximation to real motion. However, small block sizes produce a large amount of raw motion information, which increases the number of transmission bits or the required data compression complexity to condense this information. Small blocks also suffer from object (block) ambiguity problem and the random noise problem.
  – Large block sizes may produce less accurate motion vectors, since a large block may likely contain pels moving at different speeds and directions.
  – 8x8, 16x16 in H.261, MPEG-1, MPEG2
Block Matching Method

• The basic operation of block matching is picking up a candidate and calculating the matching function (usually a nonnegative function of the intensity difference) between the candidate and the current block.

• This basic operation is repeated until all the candidates have gone through and then the best matches candidate is identified. The location of the best matched candidate forms the estimated displacement vector.

• Important parameters:
  – the number of candidate blocks (search points)
  – the matching function
  – the search order of candidates
Block Matching Method

- **Search range:**
  - The maximum range of motion vectors.
  - Decided by experiment or hardware constraints.
  - Assume the size of the image block is $N_1 \times N_2$, and the maximum horizontal and vertical displacements are less than $d_{\text{max-x}}$ and $d_{\text{max-y}}$, respectively.
  - Assume only integer-value motion vectors are considered:
    - the size of the search range = $(N_1 + 2d_{\text{max-x}}) \times (N_2 + 2d_{\text{max-y}})$
    - the number of candidate blocks = $(2d_{\text{max-x}} + 1) \times (2d_{\text{max-y}} + 1)$
  - Computational load of exhaustive search: $\alpha(d_{\text{max-x}} \cdot d_{\text{max-y}})$
Matching Function

- The selection of the matching function has a direct impact on computational complexity and the displacement vector accuracy.
- Let \((d_1, d_2)\) represent a motion vector candidate inside the search region and \(f(n_1, n_2, t)\) be the digitalized image intensity at the integer-valued 2-D image coordinate \((n_1, n_2)\) at the \(t\)-th frame.
  - Normalized cross-correlation function (NCF):
    \[
    NCF \ (d_1, d_2) = \frac{\sum_{n_1} \sum_{n_2} f(n_1, n_2, t) f(n_1 - d_1, n_2 - d_2, t - 1)}{[\sum_{n_1} \sum_{n_2} f^2(n_1, n_2, t)]^{1/2} [\sum_{n_1} \sum_{n_2} f^2(n_1 - d_1, n_2 - d_2, t - 1)]^{1/2}}
    \]
  - Mean square error (MSE)
    \[
    MSE(d_1, d_2) = \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} [f(n_1, n_2, t) - f(n_1 - d_1, n_2 - d_2, t - 1)]^2
    \]
  - Mean absolute difference (MAD):
    \[
    MAD(d_1, d_2) = \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} |f(n_1, n_2, t) - f(n_1 - d_1, n_2 - d_2, t - 1)|
    \]
  - Number of threshold difference (NTD)
    \[
    NTD(d_1, d_2) = \sum_{n_1} \sum_{n_2} N(f(n_1, n_2, t) - f(n_1 - d_1, n_2 - d_2, t - 1))
    \]
    where \(N(\alpha, \beta) = \begin{cases} 1, & \text{if } |\alpha - \beta| > T_0 \\ 0, & \text{if } |\alpha - \beta| \leq T_0 \end{cases}\)
Matching Function

• Remarks:
  – To estimate the motion vector, we normally maximize the value of NCF or minimize the values of the other three functions.
  – In detection theory, if the total noise (a combination of coding error and the other factors violating our motion assumptions) can be modeled as white Gaussian, then the NCF is the optimal matching criterion. However, the white Gaussian assumption is not completely valid for real images. In addition, the computation requirement of NCF is enormous. The other matching functions are regarded as more practical, and they perform almost equally well for real images.
  – NTD can be adjusted to match the subjective thresholding characteristics of the human visual system.
  – MAD is the most popular choice in designing practical image coding systems because of its good performance and relatively simple hardware structure.
Fast Search Algorithms

• Basic Principle:
  – Breaking up the search process into a few sequential steps and choosing the next-step search direction based on the current-step result.
  – At each step, only a small number of search points are calculated. Therefore, the total number of search points is significantly reduced.
  – Since the steps are performed in sequential order, and incorrect initial search direction may lead to a less favorable result. Also, the sequential search order poses a constraint on available parallel processing structure.
Fast Search Algorithms

• Normally, a fast search algorithm starts with a rough search, computing a set of scattered search points. The distance between two nearby search points is called search step size.

• After the current step is completed, it then moves to the most promising search points and does another search with probably a smaller step size.

Simulated Annealing, Genetic Algorithm, Neural Networks, Support Vector Machine
Fast Search Algorithms

- If the matching function is “monotonic” along any direction away from the optimal point, a well designed fast algorithm can then be **guaranteed to converge** to the global optimal point. But in reality the image signal is not a simple Markov process and it contains coding and measurement noises; therefore, the monotonic matching function assumption is often not valid and consequently **fast search algorithms are often suboptimal**.

![Possible next step](image)

- **isotropic**: random selection
- **gradient-based**: normal vector

![Equipotential lines](image)
2-D-log Search Procedure

- Diamond-shaped search area (at most 5-points per step)
- 9 search points in 3x3 area surrounding the last best matching point are compared.
- The step size is reduced to half of its current value if the best match is located at the center or located on the border of the maximum search region.
3-step Search Procedure

- The search starts with a step size equal to or slightly larger than half of the max search range.
- 9-points are compared in each step.
- The step size is reduced by half, after each step, and the search ends with step size of 1 pel.
Fast Search Algorithms

• Remarks:
  – A threshold function is used to terminate the search process without reaching the final step. As long as the matching error is less than a small threshold, the resultant motion vector would be acceptable.
  – One-at-a-time search: Separate a 2-D search problem into two 1-D problems. That is, look for the best matching point in one direction first, and then looks in the other direction.

<table>
<thead>
<tr>
<th>Search algorithm</th>
<th>Number of search points</th>
<th>Number of search steps</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>min</td>
<td>max</td>
</tr>
<tr>
<td>Exhaustive search</td>
<td>225</td>
<td>225</td>
</tr>
<tr>
<td>2D-log search</td>
<td>13</td>
<td>26</td>
</tr>
<tr>
<td>3-step search</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>Modified-log search</td>
<td>13</td>
<td>19</td>
</tr>
<tr>
<td>One-at-a-time</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>Orthogonal search</td>
<td>13</td>
<td>13</td>
</tr>
</tbody>
</table>
Fast Search Algorithms

- Remarks:

  - In hardware systems, the exhaustive search and the 3-step search are often favored for their good PSNR performance, their fixed and fewer number of search steps and their identical operation in every step.
Variants of Block Matching Algorithms

- The computation load could also be reduced by calculating fewer blocks of an image.
  - To increase search efficiency, we could place the initial search point at a location predicted from motion vectors of the spatially or the temporally adjacent blocks. A best matching can often be obtained by searching a smaller region surrounding this initial point.
  - We could first separate the moving image blocks from the stationary ones and then conduct block matching only on the moving objects. This is because a moving or change detector can be implemented with much fewer calculations than a motion estimator.
  - We could use only a portion of the pels inside an image block (subsampled images) to calculate the matching function. However, the accuracy of motion might be reduced.
  - Perform the motion estimation only on alternate blocks in an image, and the motion vectors of the missing blocks are “interpolated” from the calculated motion vectors.
Variants of Block Matching Algorithms

- Basic principle:
  - A large block size is chosen at the beginning to obtain a rough estimation of the motion vector. Because a large-size image pattern is used in matching. The ambiguity problem—blocks of similar content can often be eliminated. However, motion vectors estimated from large blocks are not accurate. We then refine the estimated motion vector by decreasing the block size and the search region.
  - A new search with a smaller block size starts from an initial motion vector that is the best matched motion vector in the previous stage. Because pels in a small block are more likely to share the same motion vector, the reduction of block size typically increases the motion vector accuracy.
Variants of Block Matching Algorithms

• Variable-block-size motion estimation:
  – Image frames are partitioned into non-overlapped large image blocks. If the motion-compensated estimation error is higher than a threshold, this large block is not well compensated; therefore, it is further partitioned into, say, four smaller blocks.