Wireless Communication Systems
@CS.NCTU

Lecture 7: Video Basics

Instructor: Kate Ching-Ju Lin (林靖茹)

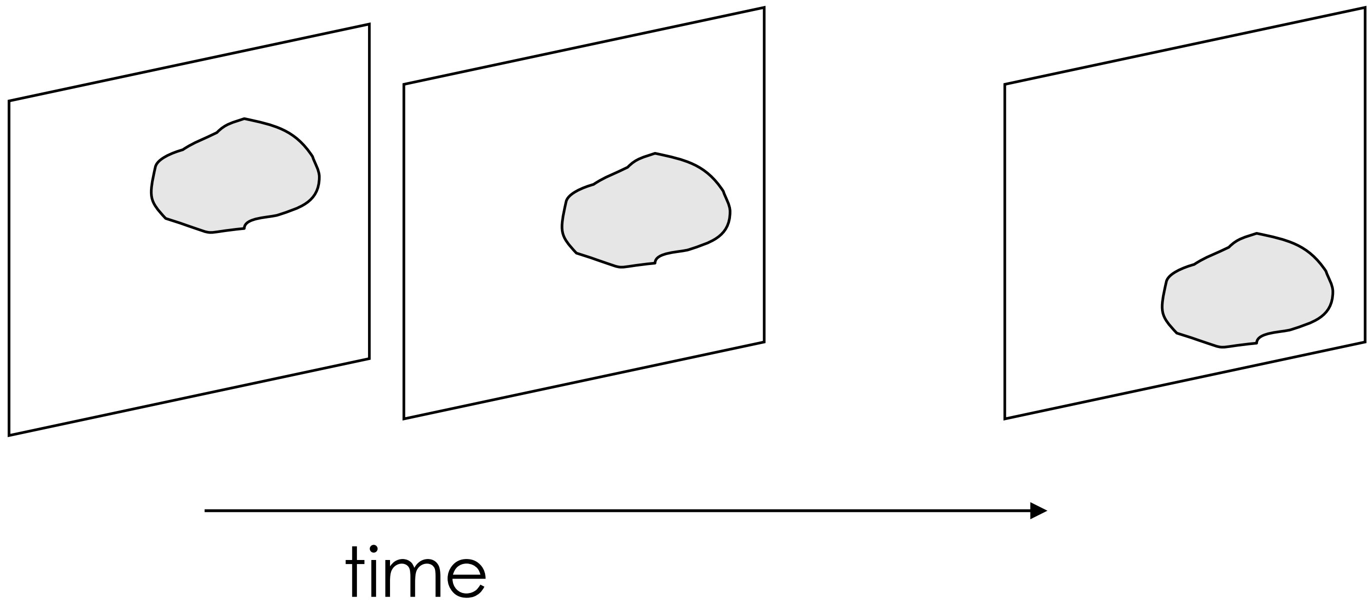
Chap. 10 of "Fundamentals of Multimedia"
Some reference from http://media.ee.ntu.edu.tw/courses/dvt/15F/

Outline

- **Introduction to video compression**
- Video codec
- Motion estimation/compensation
- Search criteria
- Search for motion vectors

What is a Video?

- A time-ordered sequence of image frames



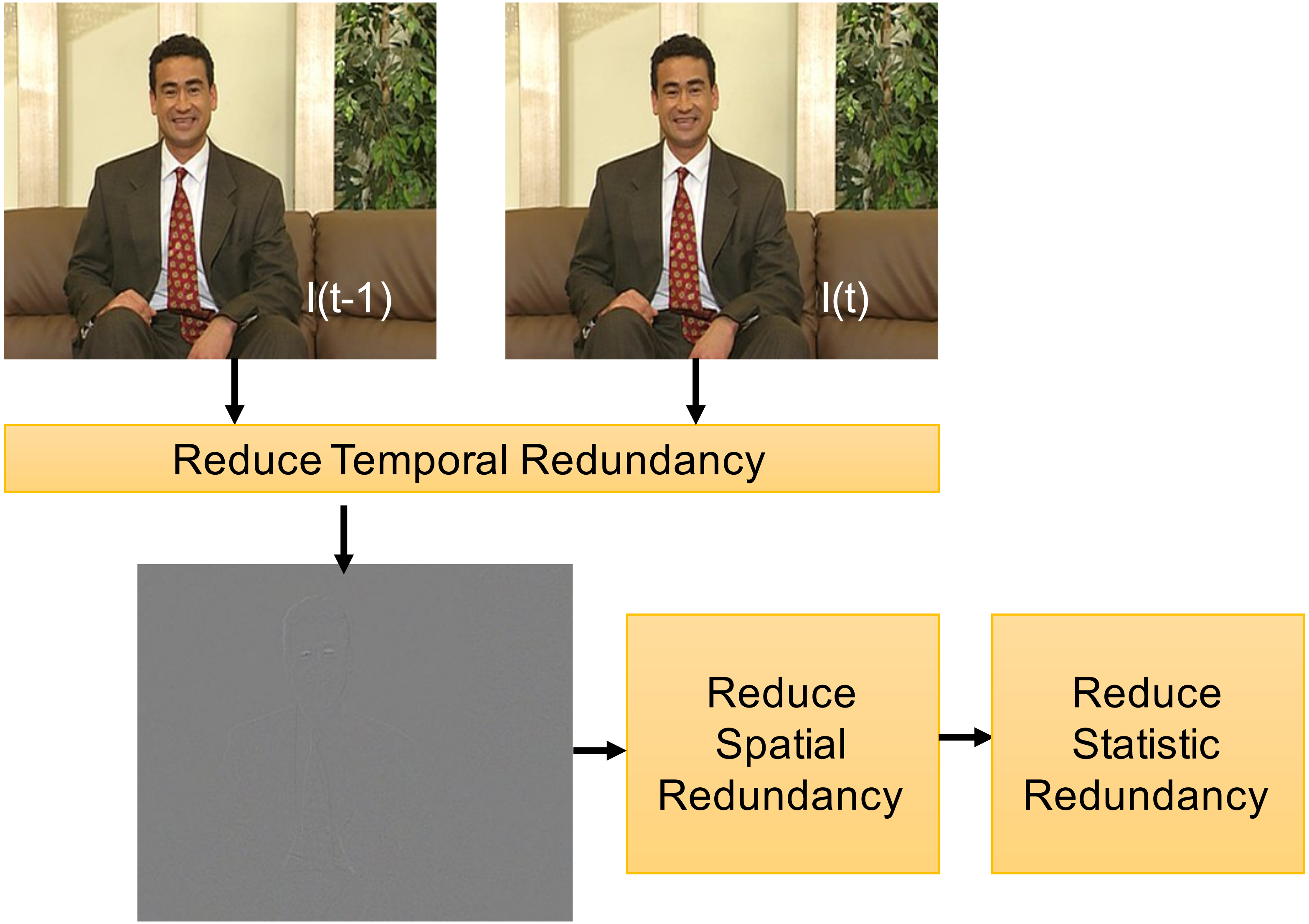
Why Need Video Compression?

- What is the video rate?
  - Number of bits per second
  - $$\frac{\text{Frame resolution (bits)}}{\text{Time interval per frame (s)}} = \text{frame resolution} \times \text{frame rate}$$
- Uncompressed video could be extremely large
  - For example, the rate of a video with resolution of 352 x 288 would be more than 35 Mbps
  - Bitrate of HDTV would easily exceed 1 Gbps
- Large video size poses challenges for storage and communications

How Video Compression Works?

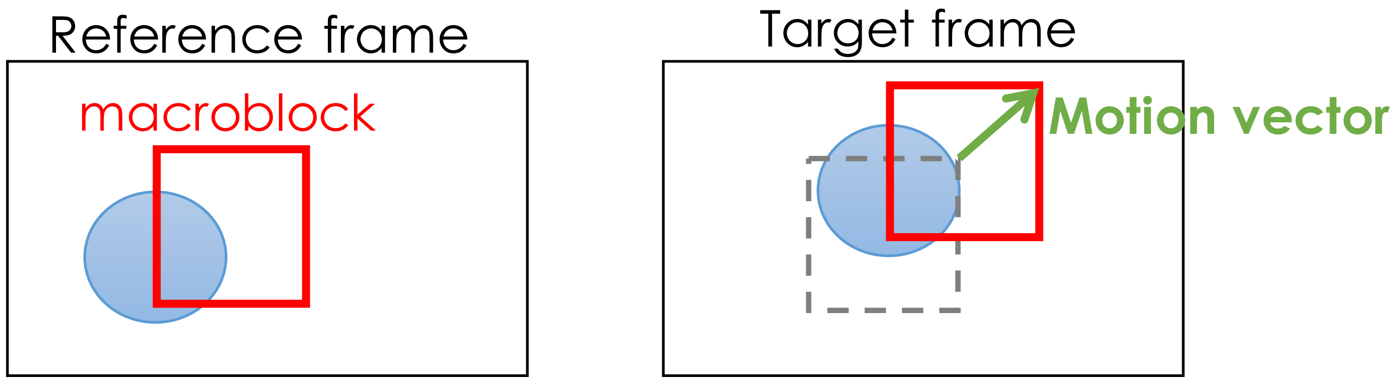
- High-level idea: predict future frames based on previous frames
  - Static video: the background of a news video is almost unchanged → Easier to compress
  - Dynamic video: the frames of a basketball game changes all the time → harder to compress
- How to predict?
  - Motion: how object is moving from one frame to the next
  - **Motion estimation/compensation:** predict a frame from neighboring reference frames
- Image vs. video compression
  - Image compression exploits spatial redundancy, while video compression exploits both spatial and temporal redundancy

Hybrid Video Coding Process



Terminologies

- **Motion vector**
  - Displacement of a reference frame to the target frame



- **Forward and backward prediction**
  - Forward: the reference frame is a previous frame
  - Backward: the reference frame is a future frame
- **Prediction error**
  - Difference between the predicted frame and the actual uncompressed frame

Outline

- Introduction to video compression
- **Video codec**
- Motion estimation/compensation
- Search criteria
- Search for motion vectors

List of Video Codec

- H.261
- H.262
- H.263
- H.264 (AVC)
- H.265 (HEVC)
- MPEG-1 (Part-2)
- MPEG-2 (Part-2)
- MPEG-4 (Part-2)
- MPEG-4 (Part-10)
- MPEG-H (Part-2)

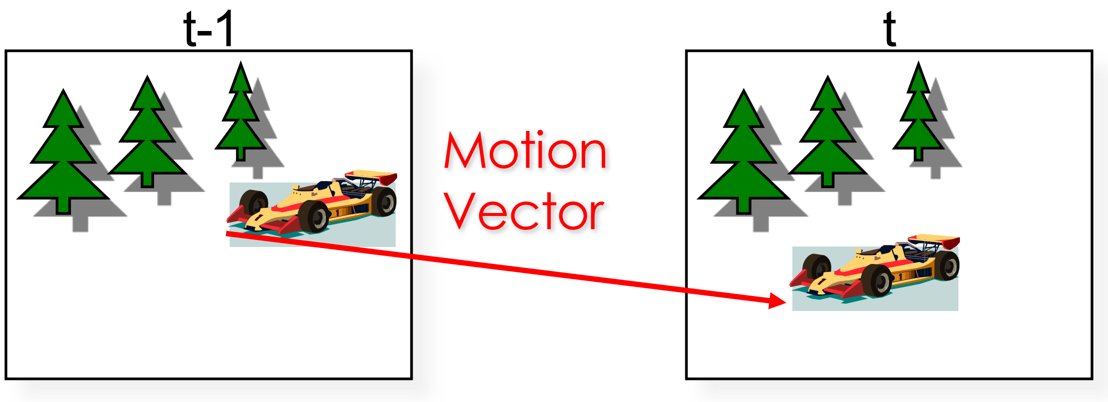
- Multi-Description Code
- MPEG-7: Object-based coding
- Scalable Video Coding (SVC, or MPEG-21 Part-13)
- Multiview Video Coding
- Dynamic Adaptive Streaming over HTTP (DASH)

Outline

- Introduction to video compression
- Video codec
- **Motion estimation/compensation**
- Search criteria
- Search for motion vectors

Motion Estimation/Compensation

- **Motion estimation**
  - To derive the motion information between frames
  - Or, find the corresponding points between frames



- **Motion compensation**
  - To compensate the displacement between frames based on motion vectors
  - Or, align or wrap one frame to the other

Motion Estimation/Compensation

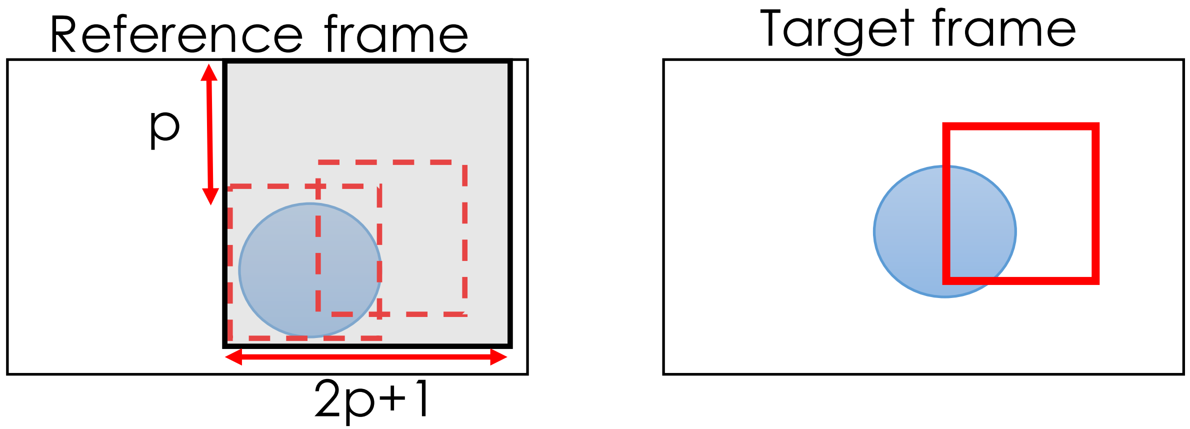
Original

Reconstructed

Similar to DPCM

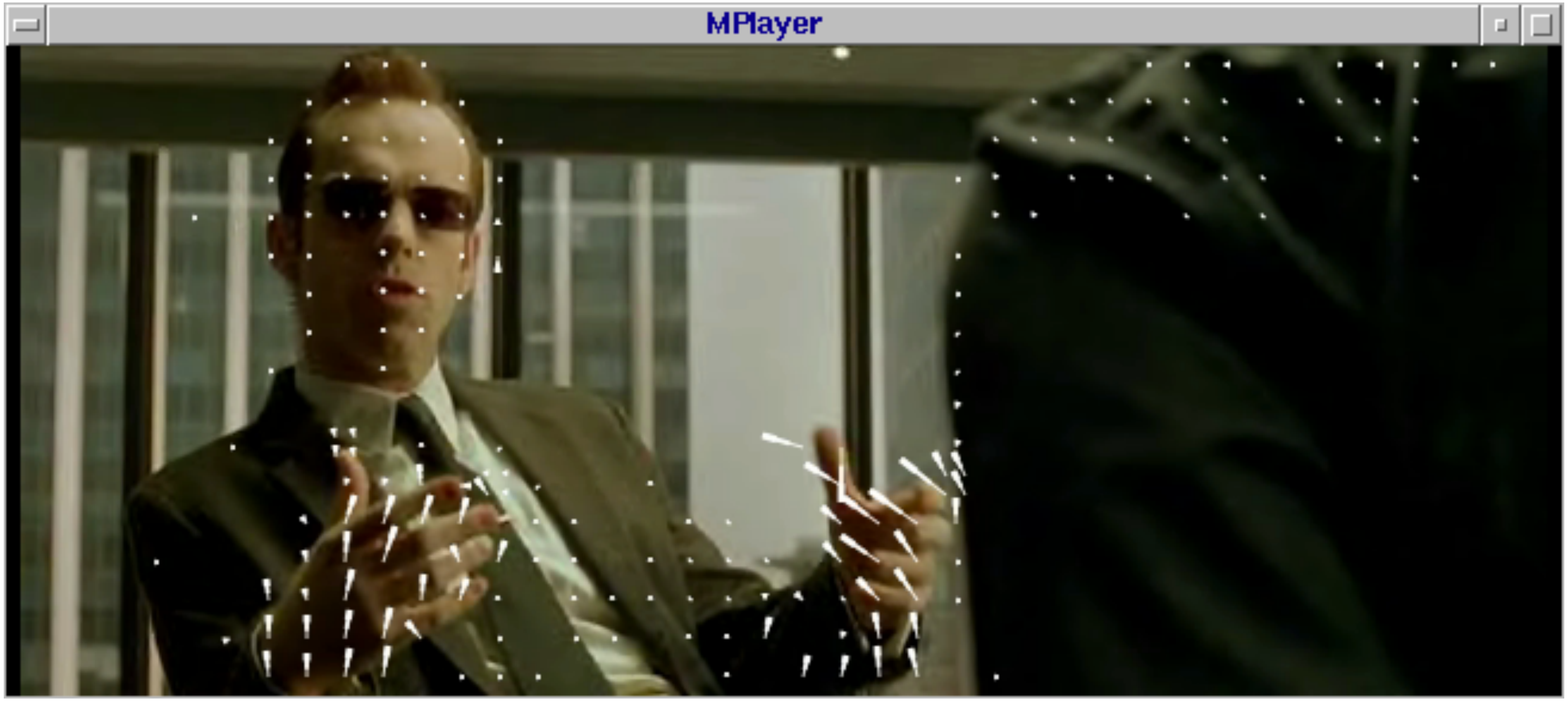
Difference

Search for Motion Vector



- **High-level idea:** search a macroblock in the reference frame **most similar to** the target macroblock
- Find a motion vector that minimizes the difference between the two macroblocks
  - ~60% complexity of encoding
  - Might search only a sub-area of the frame (range [-p,p])

Example of Motion Vectors



Source: https://people.xiph.org/~xiphmont/demo/theora/demo2.html

Outline

- Introduction to video compression
- Video codec
- Motion estimation/compensation
- **Search criteria**
- Search for motion vectors

Matching Criteria

- How do we **quantify "similarity"**?
- Specifically, how $C(x,y)$ differs from $R(x+i, y+i)$?
  - $C(x,y)$: luminance of the location (x,y) in the target macroblock
  - $R(x+i, y+i)$: luminance of the location $(x+i, y+j)$ in the reference macroblock, where i and j are the horizontal and vertical displacements
- Metrics
  - Mean Squared Error (MSE)
  - Mean Absolute Error (MAE)
  - Cross-Correlation Function (CCF)
  - Pel Difference Classification (PDC)
  - Minimized Maximum Error (MiniMax)

Matching Criteria

- Mean Squared Error (MSE)

$$\text{MSE}(i,j) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} [C(x+k, y+l) - R(x+i+k, y+j+l)]^2$$

- Mean Absolute Error (MAE)

$$\text{MAE}(i,j) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} |C(x+k, y+l) - R(x+i+k, y+j+l)|$$

- Cross-Correlation Function (CCF)

$$\text{CCF}(i,j) = \frac{\frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} C(x+k, y+l) \cdot R(x+i+k, y+j+l)}{\left[\sum_{k=0}^{N-1} \sum_{l=0}^{N-1} C(x+k, y+l)^2\right]^{1/2} \left[\sum_{k=0}^{N-1} \sum_{l=0}^{N-1} R(x+i+k, y+j+l)^2\right]^{1/2}}$$

Displacement of the best reference macroblock

$(i^*, j^*) = \min\arg\ \text{MSE}(i,j)$ or $\min\arg\ \text{MAE}(i,j)$ or $\max\arg\ \text{CCF}(i,j)$

Matching Criteria

- Pel Difference Classification (PDC)
  - compare each pixel of the target block with that of the reference block
  - classify each pixel pair as either matching or not ($\Delta<\varepsilon$?)
  - count the number of matches
  - greater is better
- Minimized Maximum Error (MiniMax)
  - find the pel difference of each pixel
  - find the maximal pel difference of a reference block (Δ_{max})
  - identify the one that minimizes the maximal pel difference

Outline

- Introduction to video compression
- Video codec
- Motion estimation/compensation
- Search criteria
- **Search for motion vectors**

Search Algorithms

- Full search
- 2D logarithmic search (TDL)
- Binary search
- Hierarchical search
- Pel-recursive
- Optical flow
- Orthogonal search
- One at a time
- Cross search

Full Search

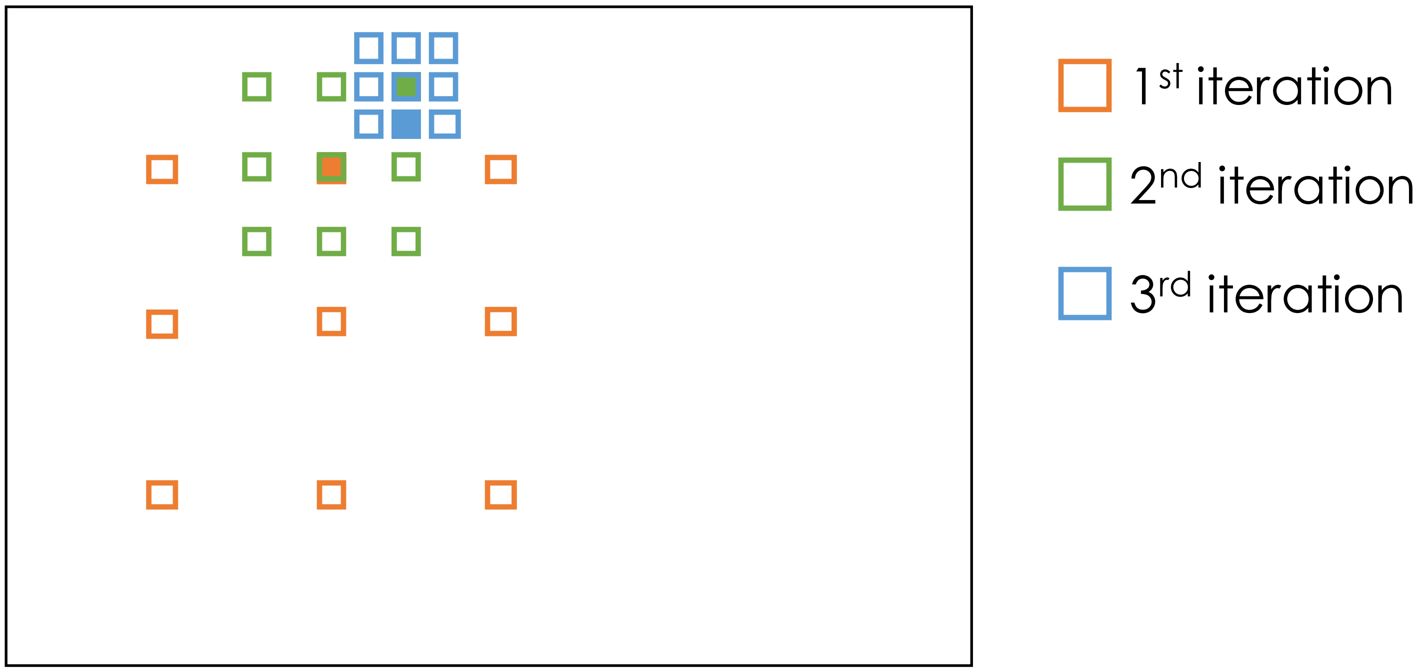
- Sequentially search the whole $(2p+1)\times(2p+1)$ macroblocks in the reference frame
- A.k.a sequential search

```
minMAD = INT_MAX;
for i = -p to p
    for j = -p to p
    {
        nowMAD = MAD(i,j);
        if nowMAD < minMAD
        {
            minMAD = nowMAD;
            u = i; v = j;
        }
    }
```

- Extremely high complexity and memory requirement
  - Each macroblock needs $3*(2p+1)(2p+1)N^2 = O(p^2N^2)$

2D Logarithmic Search

- Similar to the concept of binary search
- Subsample a few blocks, and then keep searching only in the better area



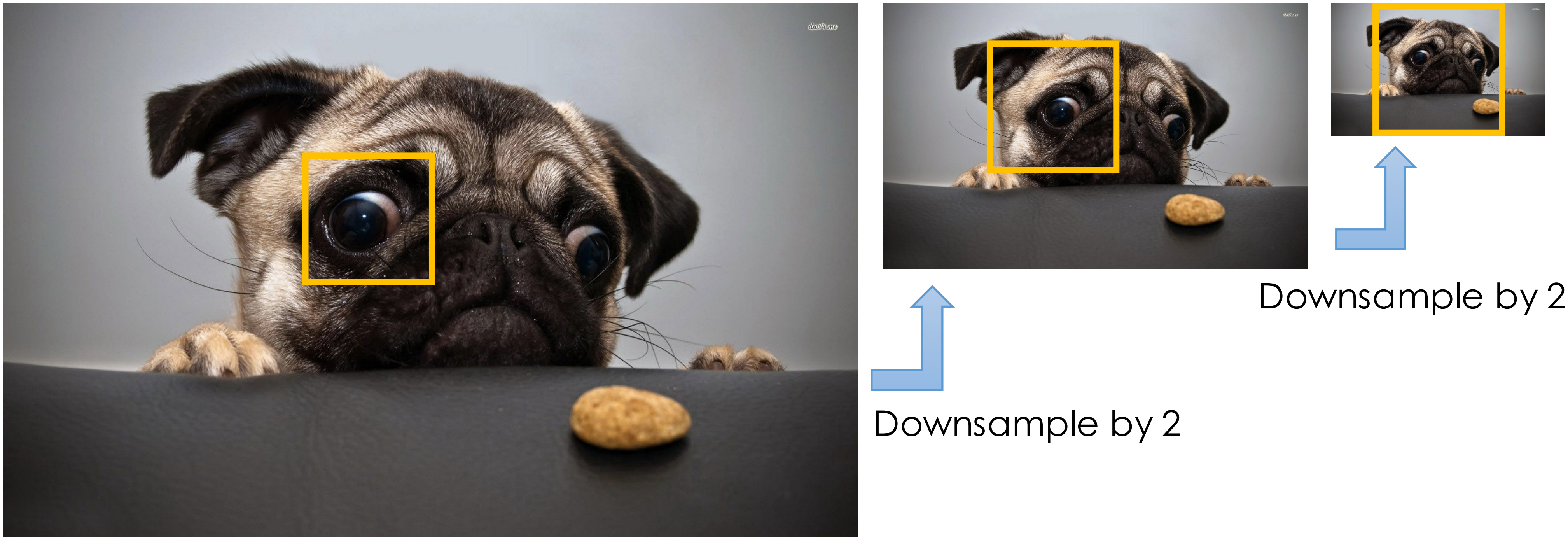
2D Logarithmic Search

- Key ideas: repeat the following tasks in iterations
  1. search k × k MBs centered at (u,v) in each iteration
  2. find the one output min MAD
  3. reduce the step-size to half, and let (u,v) be ($u+i^*$, $v+j^*$)
  4. until step-size = 1 // cannot be halved anymore
- Pseudo code for the 3 × 3 example

```
offset = round(p/2)
while (true)
{
    specify 3 x 3 macroblocks:
        (x + i * offset, y + j * offset), i,j = [-1,1];
    find the macroblock with min MAD (or MSE, CCF);
    (x,y) = the center of the selected macroblock // recentering
    break if offset = 1;
    offset = round(offset/2); // shrinking the search area
}
```

Hierarchical Search

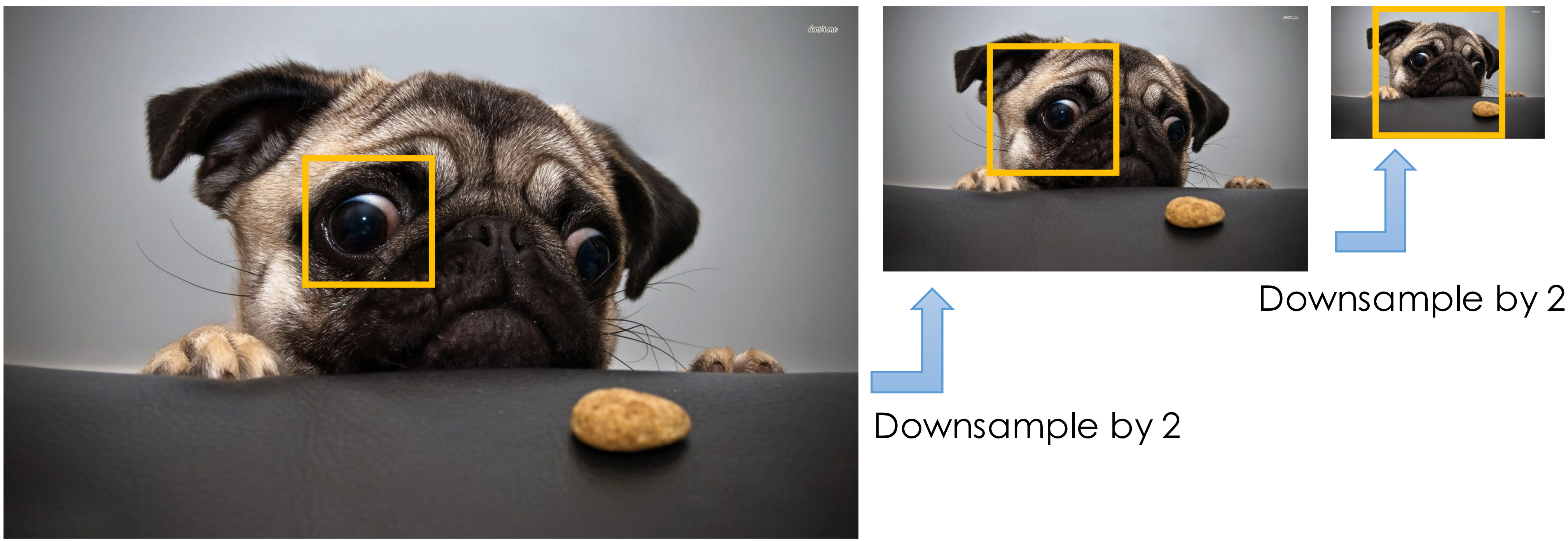
- instead of searching a frame with a high resolution, reduce the resolution by downsampling
- search with coarse resolution
- refine the resolution and search only in the area of interest



Hierarchical Search

Complexity?

subsampling + ☐ x number of layers



Hierarchical Search

```
get p x p macroblocks centered at (x,y)
    in the lowest resolution level k;
while (true)
{
    find the macroblock with min MAD (or MSE, CCF);
    (x,y) = the center of the selected macroblock;
    get p x p macroblocks centered at (x,y) in level k-1;
    break if k = 1;
    k = k - 1; // refine the resolution
}
```

Summary

- Introduction to video compression
  - Leverage spatial and temporal redundancy
- Motion estimation/compensation
  - Find movement of objects across frames
- Search criteria
  - Quantify similarity between two macroblocks
  - MSE, MAE, CCF, etc
- Search for motion vectors
  - Full sequential search
  - 2D logarithmic search
  - Hierarchical search