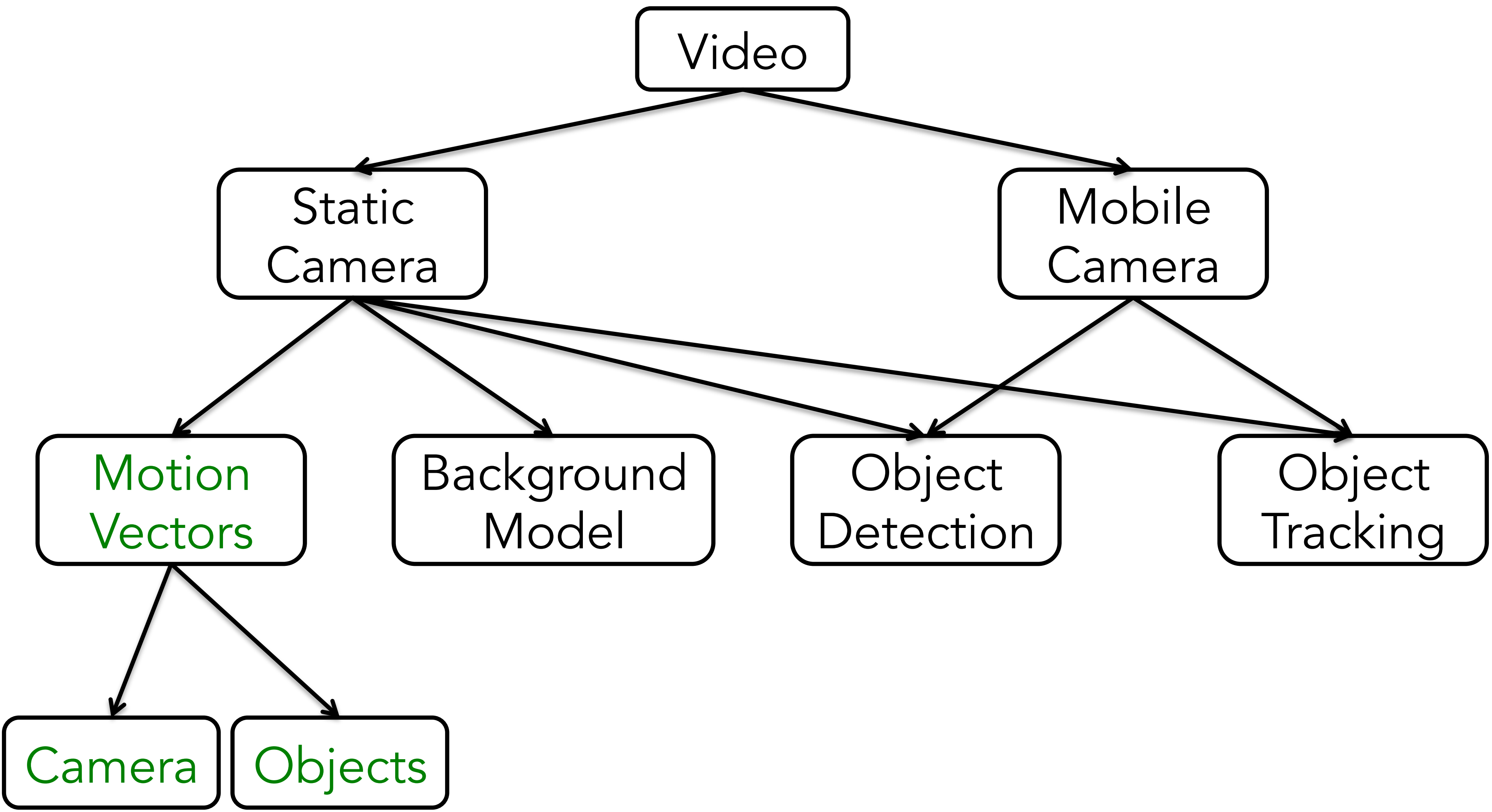
Week 9
Motion Vectors

Video Analysis



A video consists of a time-ordered sequence of frames!

Temporal redundancy!

Predictive coding based on previous frames!

Compression precedes by subtracting images: subtract in time order and code the residual error!

Can we improve the prediction?

Search for just the right parts of the image to subtract from the previous frame!

Video Compression with Motion Compensation

- Not every frame of the video needs to be coded independently as a new image
- The difference between the current frame and other frame(s) in the sequence will be coded
- Small values and low entropy, good for compression

Motion Compensation

1. Motion Estimation (motion vector search)
2. MC-based Prediction
3. Calculate prediction error, i.e., the difference

Motion Compensation Steps

- Each image is divided into *macroblocks* of size N x N.
- A match is sought between the macroblock in the Target (current) Frame and the most similar macroblock in previous and/or future frame(s) (referred to as *Reference frame(s)*).
- The displacement of the reference macroblock to the target macroblock is called a *motion vector* **MV**.

How do we get the motion vector?

Mean Absolute Difference (MAD)

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

N — size of the macroblock,

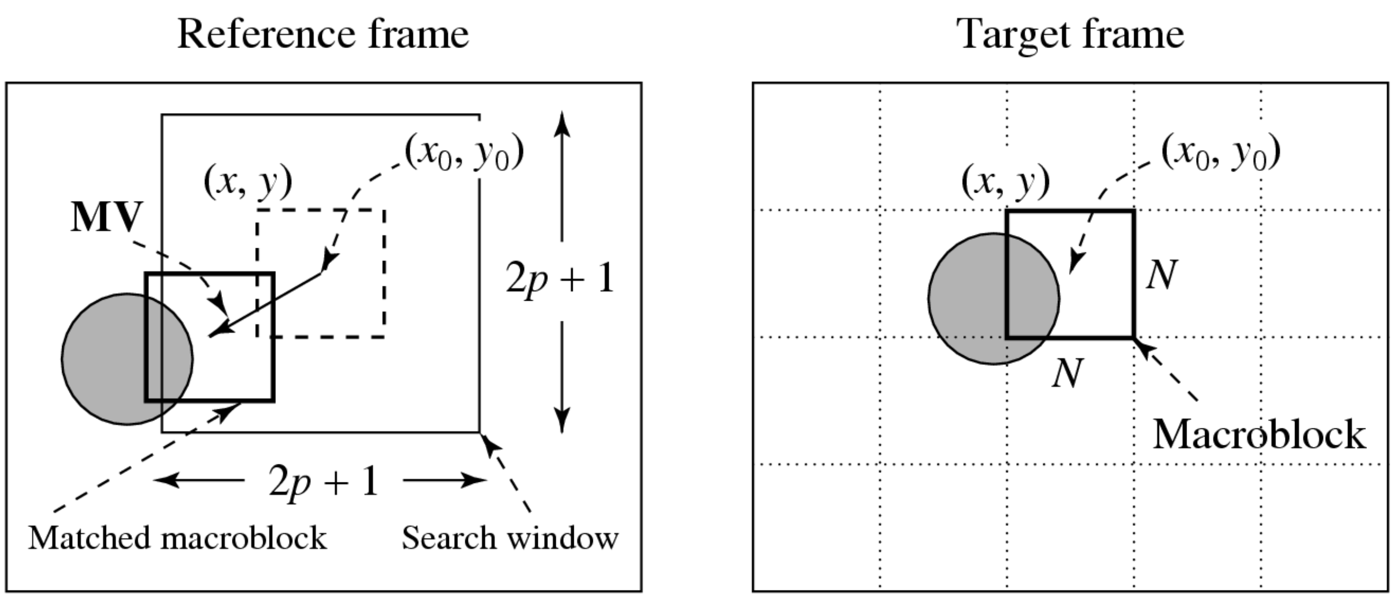
k and *l* — indices for pixels in the macroblock,

i and *j* — horizontal and vertical displacements,

C (*x* + *k*, *y* + *l*) — pixels in macroblock in Target frame,

R (*x* + *i* + *k*, *y* + *j* + *l*) — pixels in macroblock in Reference frame.

Macroblocks and Motion Vector



MV search is usually limited to a search window of size (2p + 1) x (2p + 1).

Find a vector $(\boldsymbol{i}, \boldsymbol{j})$ as the motion vector $\mathbf{MV} = (\mathbf{u}, \mathbf{v})$, such that $\boldsymbol{MAD}(\boldsymbol{i}, \boldsymbol{j})$ is minimum:

$$(u,v) = \Big[\ (i,j) \mid MAD(i,j)\ \ is\ minimum,\ \ i \in [-p,p],\ j \in [-p,p]\ \Big]$$

Sequential Search

Sequentially search the whole $(2p + 1)$ x $(2p + 1)$ window in the Reference!

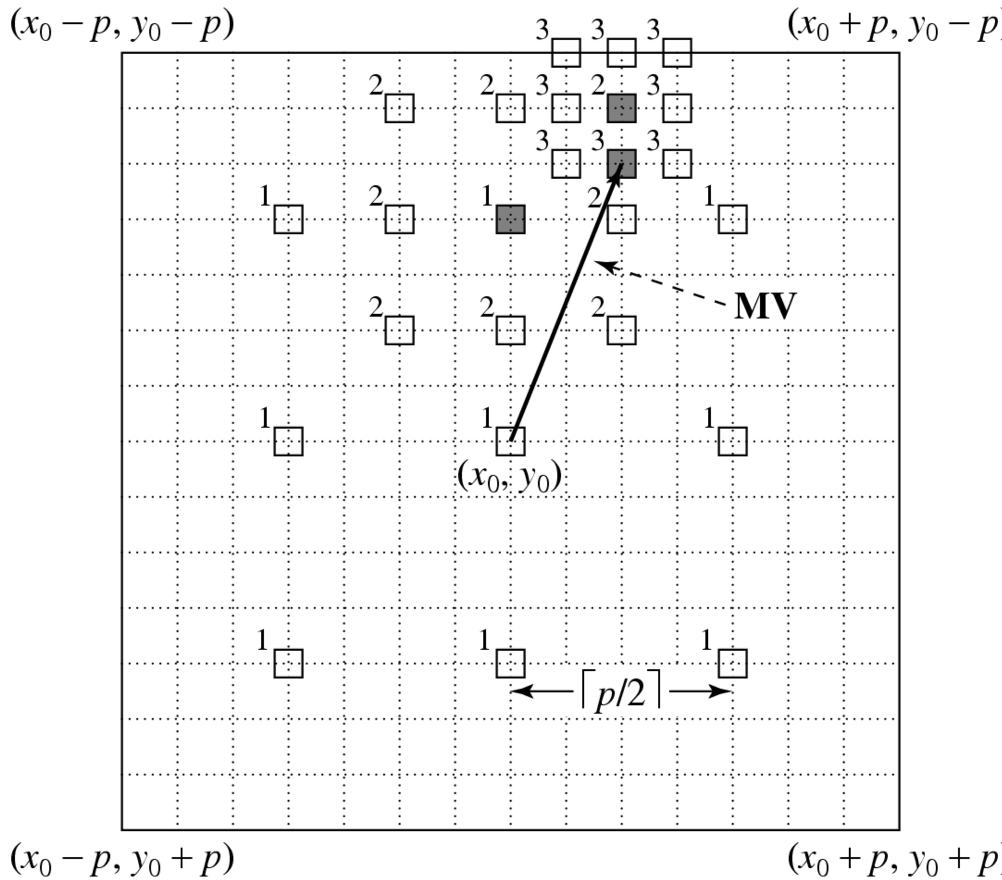
Sequential Search Complexity

$$(2p + 1) \quad (2p + 1) * \quad N^2 * \quad 3 \Rightarrow \quad O(p^2 N^2).$$

The cost for obtaining a motion vector for a single macroblock (N*N) — assuming each pixel comparison requires three operations (subtraction, absolute value, addition).

Can we get less MAD?

Three Step Search



Three Step Search

- Initially only nine locations in the search window are used as seeds for a MAD-based search; they are marked as ‘1’.
- After the one that yields the minimum ***MAD*** is located, the center of the new search region is moved to it and the step-size (“offset”) is reduced to half.
- In the next iteration, the nine new locations are marked as ‘2’ and so on.

Number of operations per macroblock?

$$\left(8\cdot(\log_2 p + 1) + 1\right)\cdot N^2 \cdot 3$$

Note: It takes three steps for p = 7, for larger p, it may need more than 3 steps, $(\log_2 p + 1)$ to be precise!

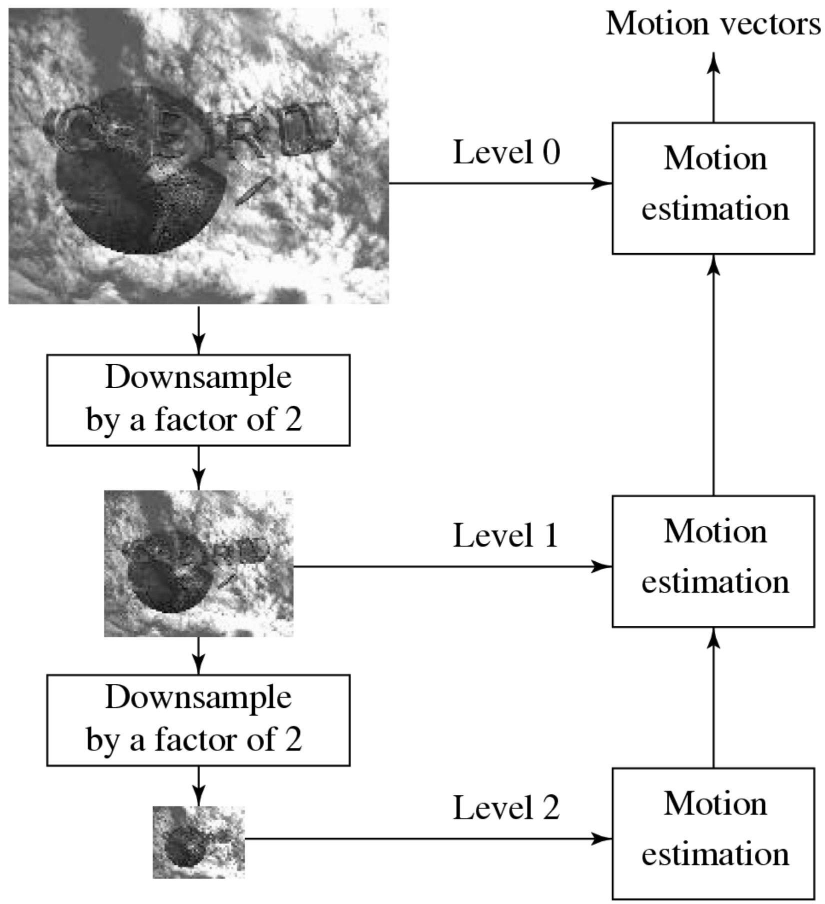
2D Logarithmic Search

- Select an initial step size (s)
- Calculate the error for the block at the center of search area and four point at x and y axis at distance of s from center
- If the position of best match is at center keep the center unchanged and reduce the step size by half, otherwise the best match becomes the center
- Repeat until step size becomes 1; when the step size becomes 1 all the 8 neighbor blocks around the center will be checked for finding the best match

Hierarchical Search

Estimation motion vectors in low resolution image and refine in high resolution image!

A Three-level Hierarchical Search for Motion Vectors



Hierarchical Search

- The search can benefit from a hierarchical (multiresolution) approach in which initial estimation of the motion vector can be obtained from images with a significantly reduced resolution.

- A three-level hierarchical search in which the original image is at Level 0, images at Levels 1 and 2 are obtained by down-sampling from the previous levels by a factor of 2, and the initial search is conducted at Level 2.

- Since the size of the macroblock is smaller and p can also be proportionally reduced, the number of operations required is greatly reduced.

Hierarchical Search

- Given the estimated motion vector (u^k, v^k) at Level k, a 3 x 3 neighborhood centered at $(2\cdot u^k, 2\cdot v^k)$ at Level $k-1$ is searched for the refined motion vector.
- The refinement is such that at Level $k-1$ the motion vector (u^{k-1}, v^{k-1}) satisfies:
  - $(2u^k - 1 \le u^{k-1} \le 2u^k + 1,\ 2v^k - 1 \le v^{k-1} \le 2v^k + 1)$

Comparison of Computational Cost

Search Method	OPS_per_second for 720×480 at 30 fps	
	$p = 15$	$p = 7$
Sequential search	29.89×10^9	7.00×10^9
2D Logarithmic search	1.25×10^9	0.78×10^9
3-level Hierarchical search	0.51×10^9	0.40×10^9