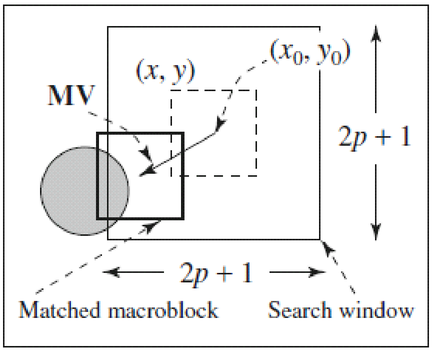
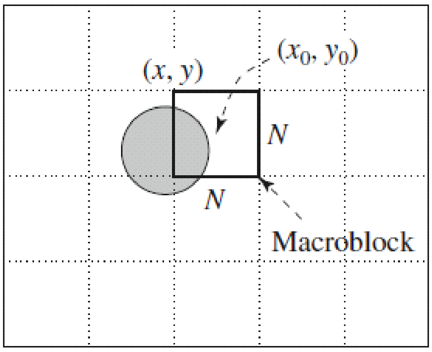
Motion Compensation

March 1, 2016 19:20

- For efficiency each image is divided into macroblocks of size N
    - Luma: N = 16
    - Chroma: N = 8
    - Performed at the pixel level
- Current frame is referred to as the 'target frame'
- A match is found between the macroblock under construction and the most similar previous/future frame
    - Usually weighted
- The displacement is called a *'motion vector'* **MV**
-





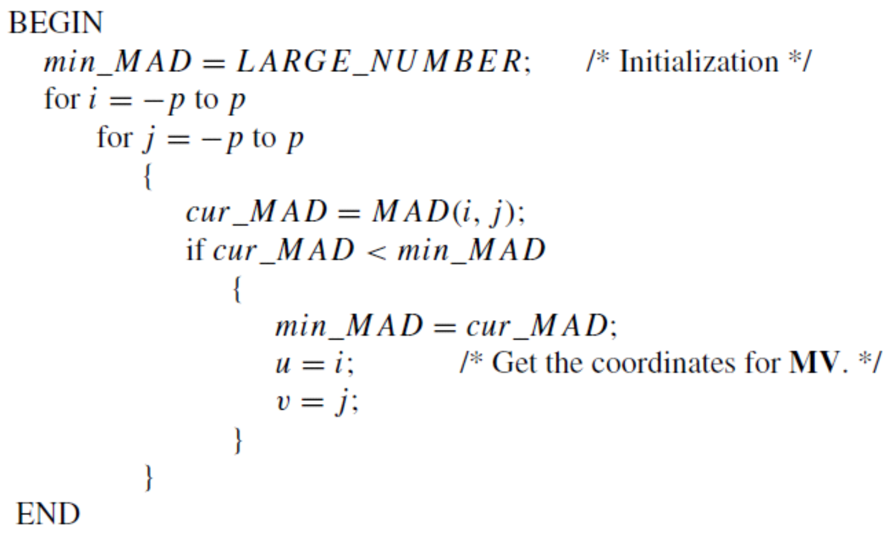
- Shows a *forward prediction*
    - Reference frame is taken to be a previous frame
        - If reference frame is taken to be a future frame, it is a *backwards prediction*

Searching For Motion Vectors

- $MV_{(u,v)}$
- To find MV we create a search window size
    - Defined as [-p, p]
        - Search window size is (2p+1) + (2p+1)
    - The center of the macro block is defined as (x_0, y_0)
        - Convenience, use the upper left corner (x, y) as the origin in the target frame
            - C(x + k, y + l) be pixels in the macroblock in the Target (current) frame
            - R(x + I + k, y + j + l) be pixels in the macroblock in the Reference frame
                - K and L are indices for pixels in the macroblock
                - I and J are horizontal and vertical displacements ((x, y) respectively)
    - Difference between the two macroblocks defined by ***Mean Absolute Difference (MAD)***

$$MAD(i, j) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} |C(x+k, y+l) - R(x+i+k, y+j+l)|, \quad (10.1)$$

- N is the size of the macroblock



```
BEGIN
  $min\_MAD = LARGE\_NUMBER$;      /* Initialization */
  for $i = -p$ to $p$
      for $j = -p$ to $p$
          {
              $cur\_MAD = MAD(i, j)$;
              if $cur\_MAD < min\_MAD$
                  {
                      $min\_MAD = cur\_MAD$;
                      $u = i$;          /* Get the coordinates for **MV**. */
                      $v = j$;
                  }
          }
 END
```