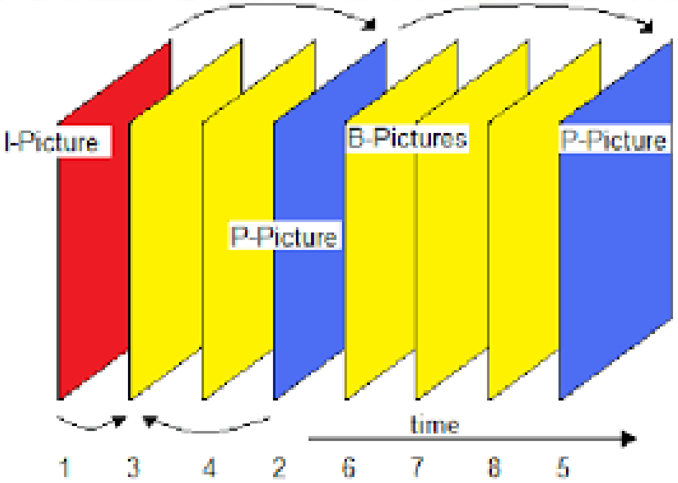
Video Processing

Introduction



- A video is a sequence of individual images, or frames, displayed in rapid succession to create the illusion of motion.
- Unlike image processing, which only considers spatial (space) data within a single frame, video processing must also account for temporal (time) relationships between frames to capture motion.
- Frames per second (fps): the number of frames displayed each second to create the illusion of motion.
- 24 fps standard for film, 30 fps for most online video, and 60 fps for gaming or slow-motion effects

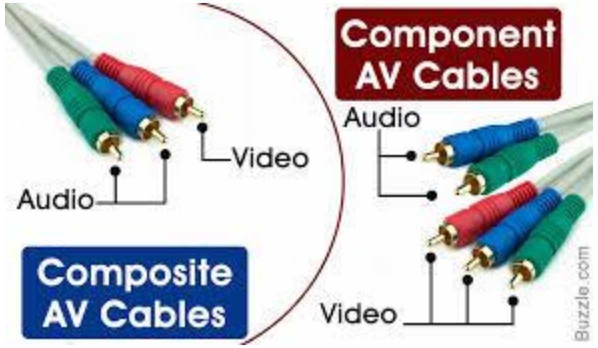
Video Display

Composite Video

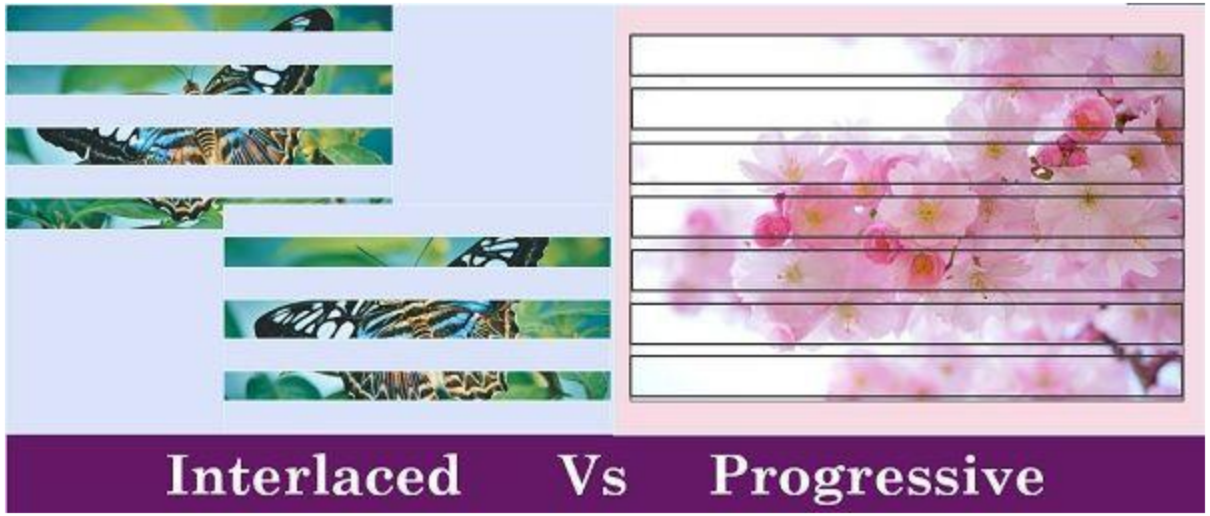
- composite video combines all visual information into a single signal.
- Merges luminance (brightness) and chrominance (color) into a single signal.

Component Video

- Splits the video information into three separate color signals (e.g., Y, Pb, and Pr).



Progressive and interlaced scan



- Progressive scanning draws each video frame sequentially from top to bottom, resulting in smoother, clearer images and higher quality but requires more bandwidth.

- Interlaced scanning divides each frame into two "fields"—one with odd-numbered lines and the other with even-numbered lines—and displays them alternately, reducing bandwidth.

Motion Estimation

- Motion estimation algorithms are based on temporal changes in image intensities.

  - Motion estimation
    1. Pixel-based motion estimation
    2. Block matching
    3. Deformable block matching
    4. Global motion estimation
    5. Region-based motion estimation
    6. Multi-resolution motion estimation
    7. Feature-based motion estimation

Optical Flow

- Optical Flow: observed or apparent 2D motion
- 2D motion vectors
- Optical Flow Equation
- Ambiguity in motion estimation

Ambiguity in motion estimation

1) At any pixel x, one cannot determine the motion vector v based on $\nabla\varphi$ and $\frac{\partial\varphi}{\partial t}$. There is only one equation for two unknowns.

$$\frac{\partial \psi}{\partial x} v_x + \frac{\partial \psi}{\partial y} v_y + \frac{\partial \psi}{\partial t} = 0$$

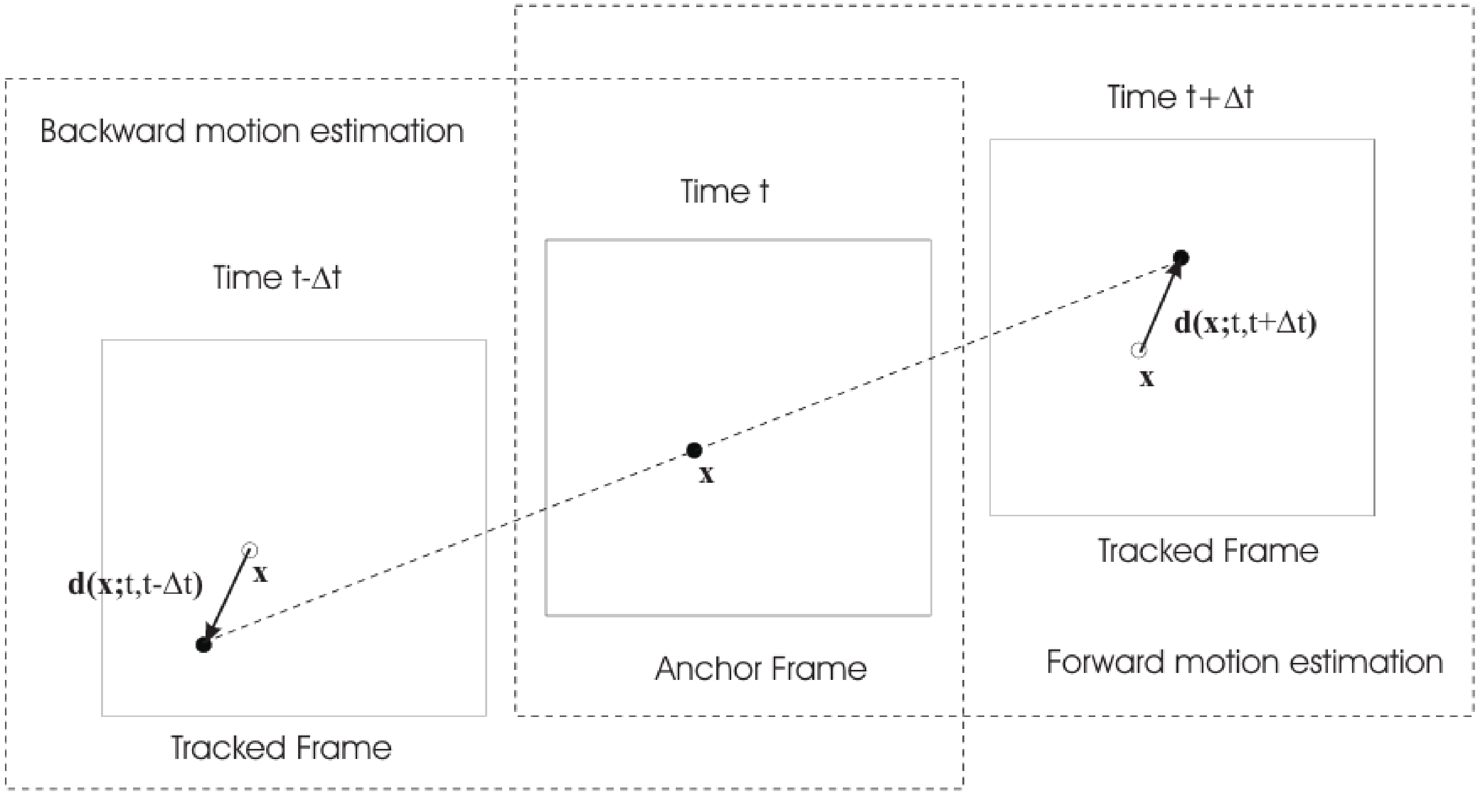
The most common constraint is that the flow vectors should vary smoothly spatially, so that one can make use of the intensity variation over a small neighborhood surrounding x to estimate the motion at x.

Ambiguity in motion estimation

2) Aperture Problem: Ambiguity in estimating the motion vector. The word “aperture" here refers to the small window over which to apply the constant intensity assumption. The motion can be estimated uniquely only if the aperture contains at least two different gradient directions.

3) In regions with constant brightness so that $\|\nabla\varphi\| = 0$; the flow vector is indeterminate. The estimation of motion is reliable only in regions with brightness variation, i.e., regions with edges or non- at textures.

Motion Estimation



Motion Estimation (Terminologies)

- In general, we can represent the motion field as $d(x;a)$, where $a = [a_1, a_2, \ldots, a_l]^T$ is a vector containing all the motion parameters.
- Similarly, the mapping function can be denoted by $w(x;a) = x + d(x;a)$.
- The motion estimation problem is to estimate the motion parameter vector a.

Motion Estimation Algorithms

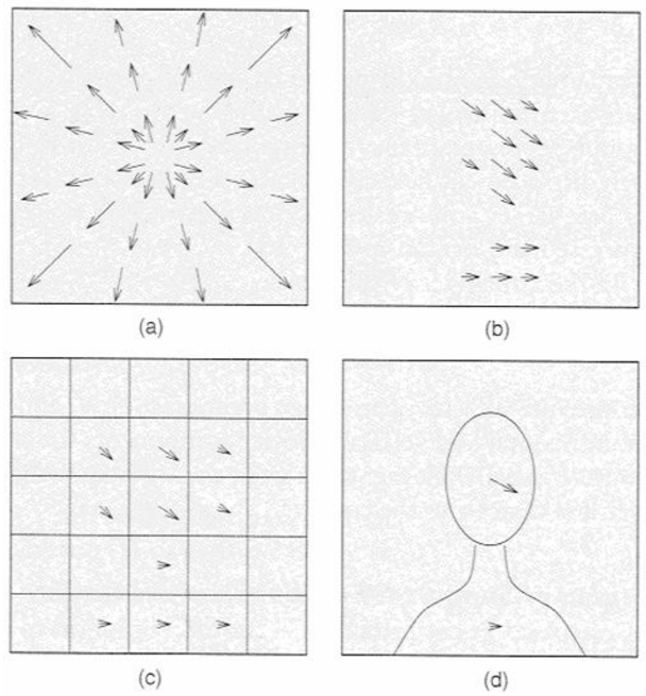
Feature-based Approach	Intensity-based Approach
✓ Establishes correspondences between selected feature points in two consecutive frames.	✓ Applies the constant intensity assumption or optical flow equation at every pixel.
✓ Motion model parameters are estimated using least squares fitting.	✓ Motion is estimated to minimize brightness change between frames.
✓ Applicable primarily to parametric motion models (e.g., affine, rigid).	✓ Suitable when motion cannot be represented by a simple model.
✓ Effective for estimating global motion, such as camera panning or rotation.	✓ Provides pixel-wise or block-wise motion fields for complex, non-rigid motion.

Intensity-based estimation

- Intensity-based motion estimation problem can be converted into an optimization problem
  - ✓Motion Representation : how to parameterize the motion field?????
  - ✓Motion Estimation Criteria : how to estimate the model parameters?????
  - ✓Minimization Methods : how to search for the optimal parameters?????

Motion Representation

1) Pixel-based motion representation
   - ✓ Motion vector at every pixel.
   - ➢ Requires the estimation of a large number of unknowns.
2) Global motion representation
   - ✓ Characterize the entire motion field.
   - ➢ Effective for estimating global motion, such as camera panning or rotation.
3) Region-based motion representation
   - ✓ Each region characterized by motion vector
   - ➢ Segmentation and estimation have to be accomplished iteratively
4) Block-based motion representation
   - ✓ Each block characterized by motion vector
   - ➢ The resulting motion is often discontinuous across block boundaries, since it does not impose any constraints



a) global, b) pixel-based, c) block-based d) region-based

Pixel-based Motion Estimation

- In pixel-based motion estimation, one tries to estimate a motion vector for every pixel.
- This problem is ill-defined
  - If one uses the constant intensity assumption, for every pixel in the anchor frame, there are many pixels in the tracked frame that have exactly the same image intensity.
  - If one uses the optical flow equation instead, the problem is again indeterminate, because there is only one equation for two unknowns.
- Regularization Technique
- Using a multi-point neighbourhood

Regularization Using Motion Smoothness Constraint

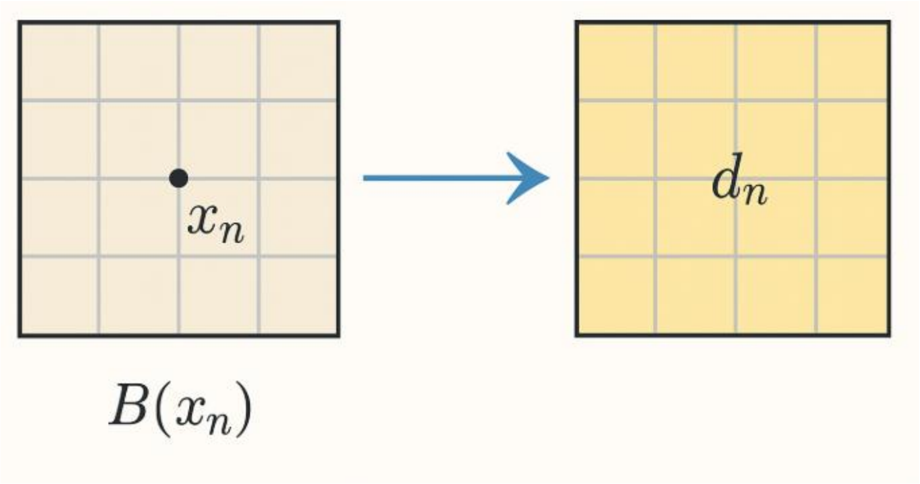
- Estimate the motion vectors by minimizing the following objective function, which is a combination of the flow-based criterion and a motion smoothness criterion:

$$E(\mathbf{v}(\mathbf{x})) = \sum_{\mathbf{x}\in\Lambda} \left(\frac{\partial \psi}{\partial x} v_x + \frac{\partial \psi}{\partial y} v_y + \frac{\partial \psi}{\partial t} \right)^2 + w_s \left(\|\boldsymbol{\nabla} v_x\|^2 + \|\boldsymbol{\nabla} v_y\|^2 \right)$$

Using a Multipoint Neighborhood

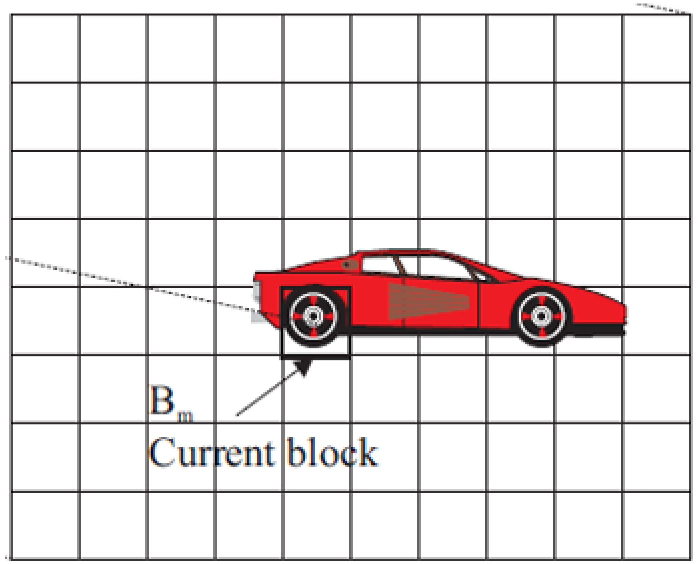
- When estimating the motion vector at a pixel x_n,we assume that the motion vectors of all the pixels in a neighborhood $B(x_n)$ surrounding it are the same, being d_n.
- To estimate the motion vector d_n for x_n, we minimize the Displaced Frame Difference (DFD) error over $B(x_n)$

$$E_n(\mathbf{d}_n) = \frac{1}{2} \sum_{\mathbf{x} \in \mathcal{B}(\mathbf{x}_n)} w(\mathbf{x}) \left(\psi_2(\mathbf{x} + \mathbf{d}_n) - \psi_1(\mathbf{x})\right)^2$$

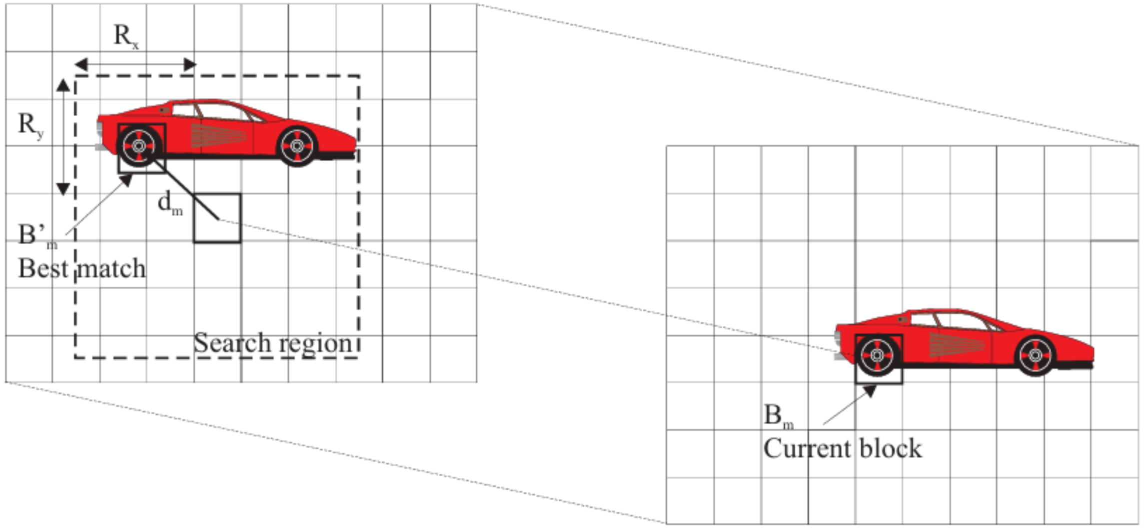


Block Matching Algorithm

- Smoothness constraints are imposed to regularize the problem in pixel based motion estimation.
- One way of imposing the smoothness constraint on the estimated motion field is to divide the image domain into non-overlapping small regions, called blocks.
- If the block is sufficiently small, then this model can be quite accurate.
- In the simplest case, the motion in each block is assumed to be constant, i.e., the entire block undergoes a translation. This is called the block-wise translational model.



Block Matching Algorithm

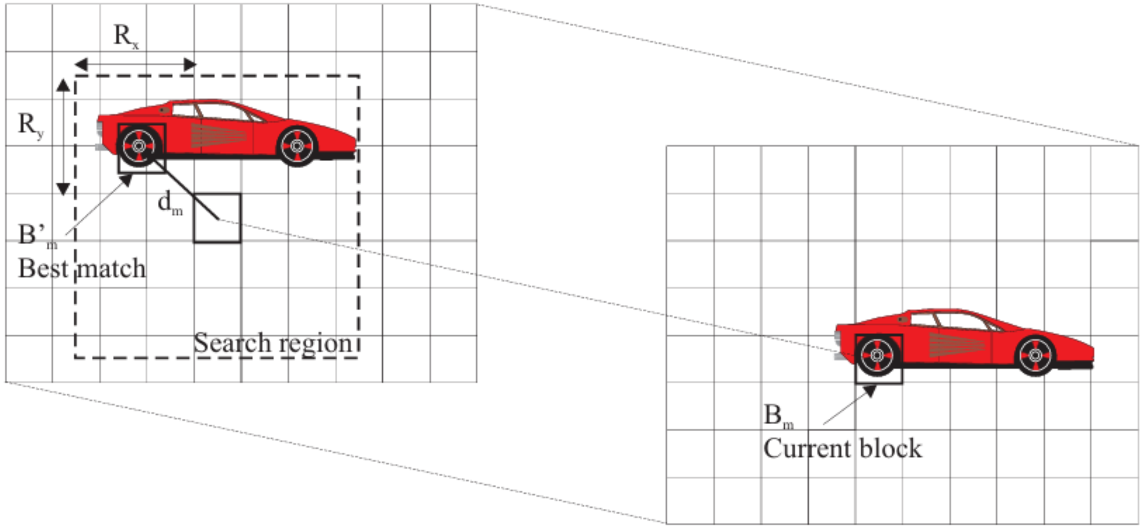


- Given an image block in the anchor frame B_m, the motion estimation problem is to determine a matching block B'_m in the tracked frame so that the error between these two blocks is minimized.

- The displacement vector d_m between the spatial positions of these two blocks (the center or a selected corner) is the MV of this block.

$$E_m(\mathbf{d}_m) = \sum_{\mathbf{x} \in \mathcal{B}_m} |\psi_2(\mathbf{x} + \mathbf{d}_m) - \psi_1(\mathbf{x})|^p$$

The Exhaustive Search Block Matching Algorithm (EBMA)

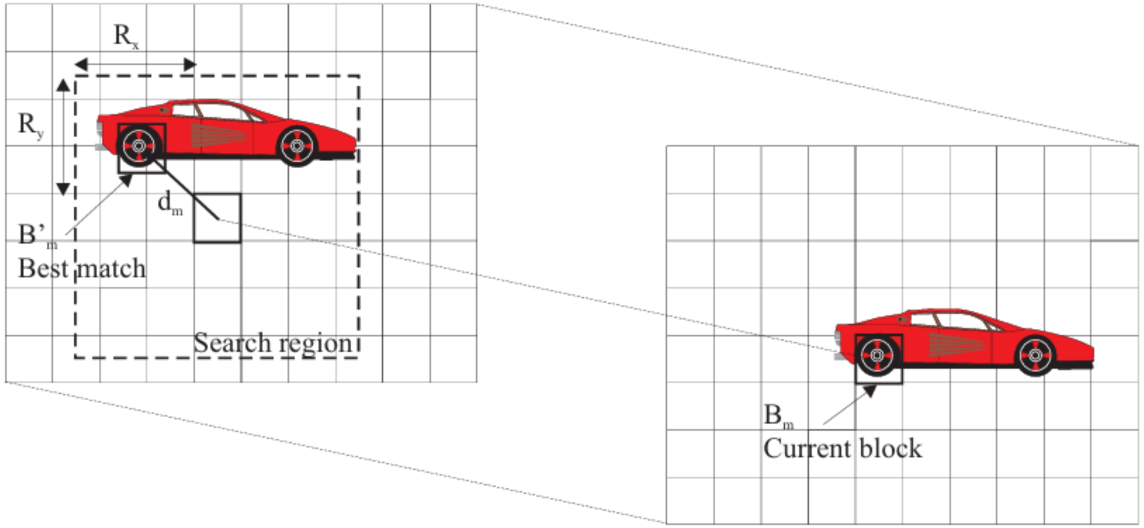


- One way to determine the d_m that minimizes the error is by using an exhaustive search method : Exhaustive Block Matching Algorithm (EBMA).

- The displacement vector d_m between the spatial positions of these two blocks (the center or a selected corner) is the motion vector of this block.

$$E_m(\mathbf{d}_m) = \sum_{\mathbf{x} \in \mathcal{B}_m} |\psi_2(\mathbf{x} + \mathbf{d}_m) - \psi_1(\mathbf{x})|^p$$

The Exhaustive Search Block Matching Algorithm (EBMA)



- EBMA determines the optimal d_m for a given block d_m in the anchor frame by comparing it with all candidate blocks B'_m in the tracked frame within predefined search region and finding the one with the minimum error.
- The search region is usually symmetric with respect to the current block, up to R_x pixels to the left and right, and up to R_y pixels above and below.