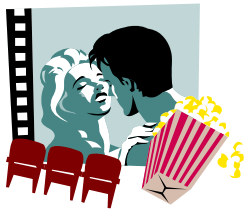
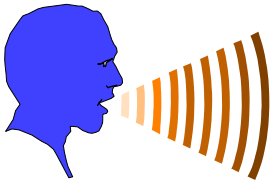



Video Processing & Communications

Two-Dimensional Motion Estimation
(Part I: Fundamentals & Basic Techniques)

Yao Wang
Polytechnic University, Brooklyn, NY11201
yao@vision.poly.edu

Acknowledgement

- We wish to thank Yao Wang from Polytechnic University, Brooklyn, for providing the Powerpoint slides corresponding to the book “Video Processing and Communications”.

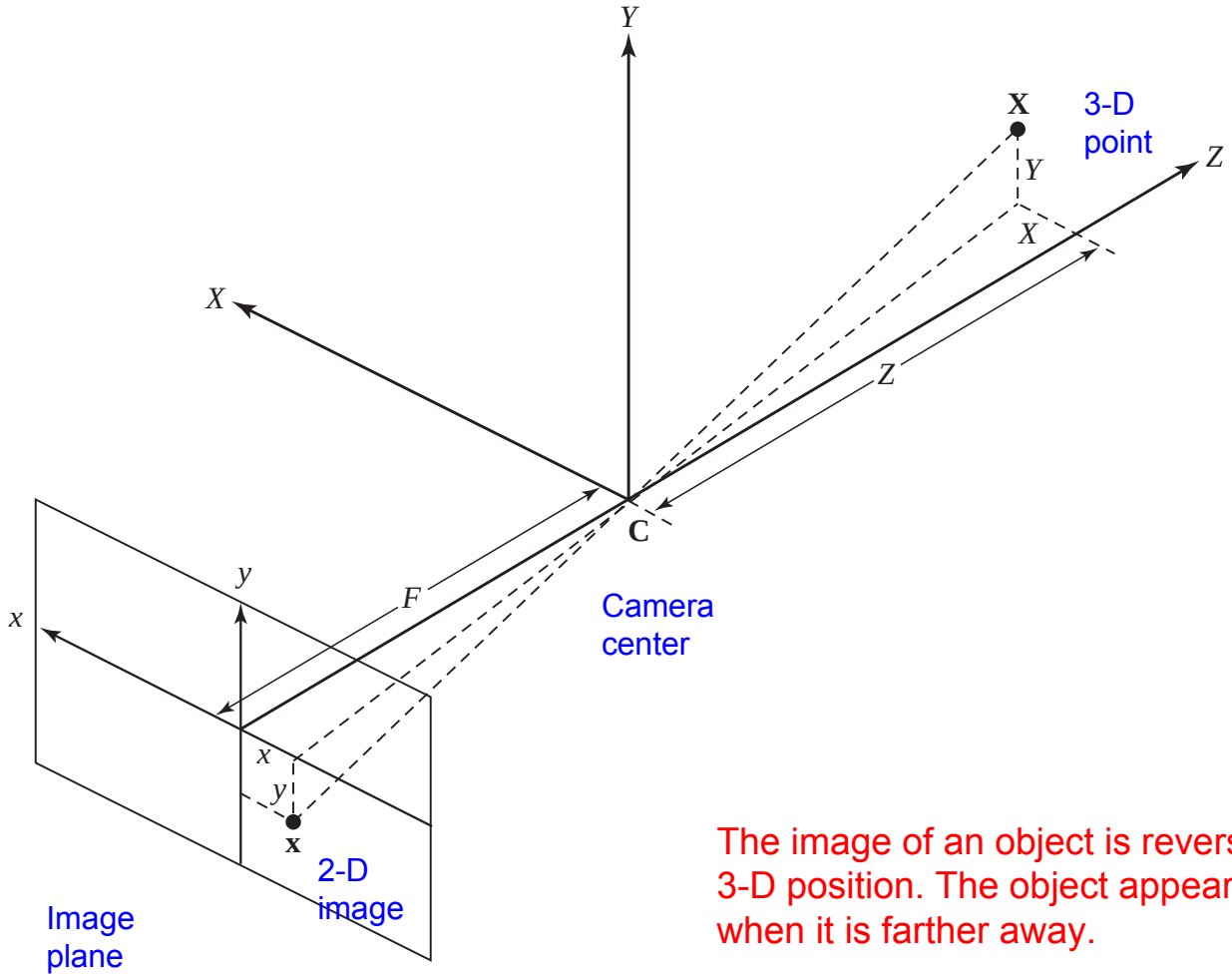
Outline

- 3-D motion model
- 2-D motion model
- 2-D motion vs. optical flow
- Optical flow equation and ambiguity in motion estimation
- Motion representation
- Motion estimation techniques
  - Block-based motion estimation
    - Exhaustive search
      - Integer-pixel accuracy
      - Fractional-pixel accuracy
    - Hierarchical search
  - Mesh-based motion estimation

2-D Motion Model

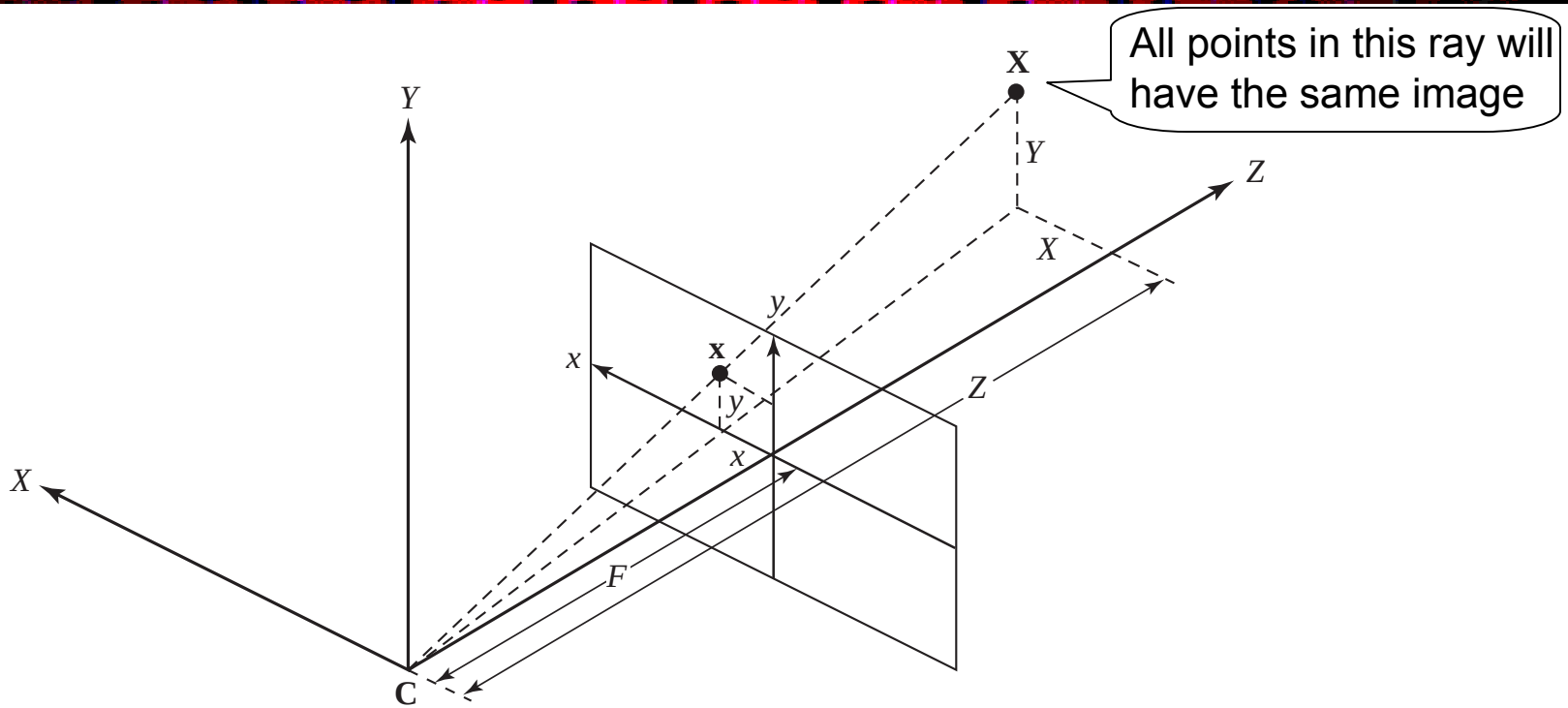
- Camera projection
- 3D motion
- Projection of 3-D motion
- 2D motion due to rigid object motion
  - Projective mapping
- Approximation of projective mapping
  - Affine model
  - Bilinear model

Pinhole Camera



The image of an object is reversed from its 3-D position. The object appears smaller when it is farther away.

Pinhole Camera Model: Perspective Projection

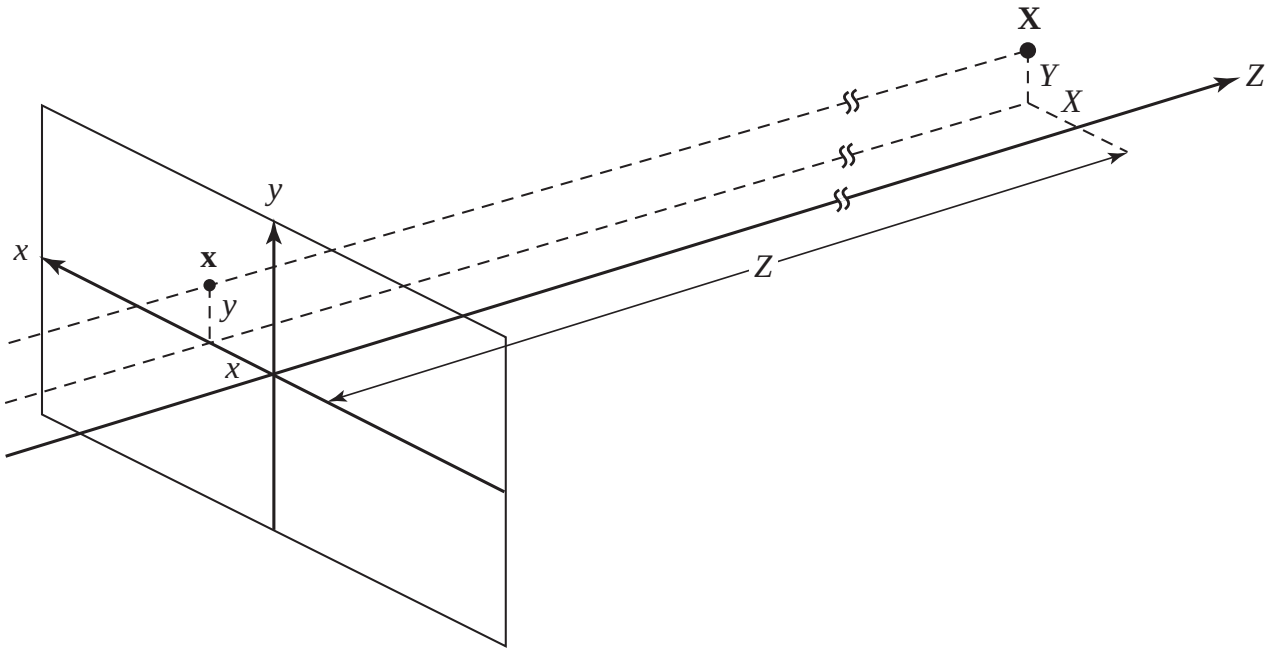


(a)

$$\frac{x}{F}=\frac{X}{Z},\frac{y}{F}=\frac{Y}{Z}\Rightarrow x=F\frac{X}{Z},y=F\frac{Y}{Z}$$

x, y are inversely related to Z

Approximate Model: Orthographic Projection



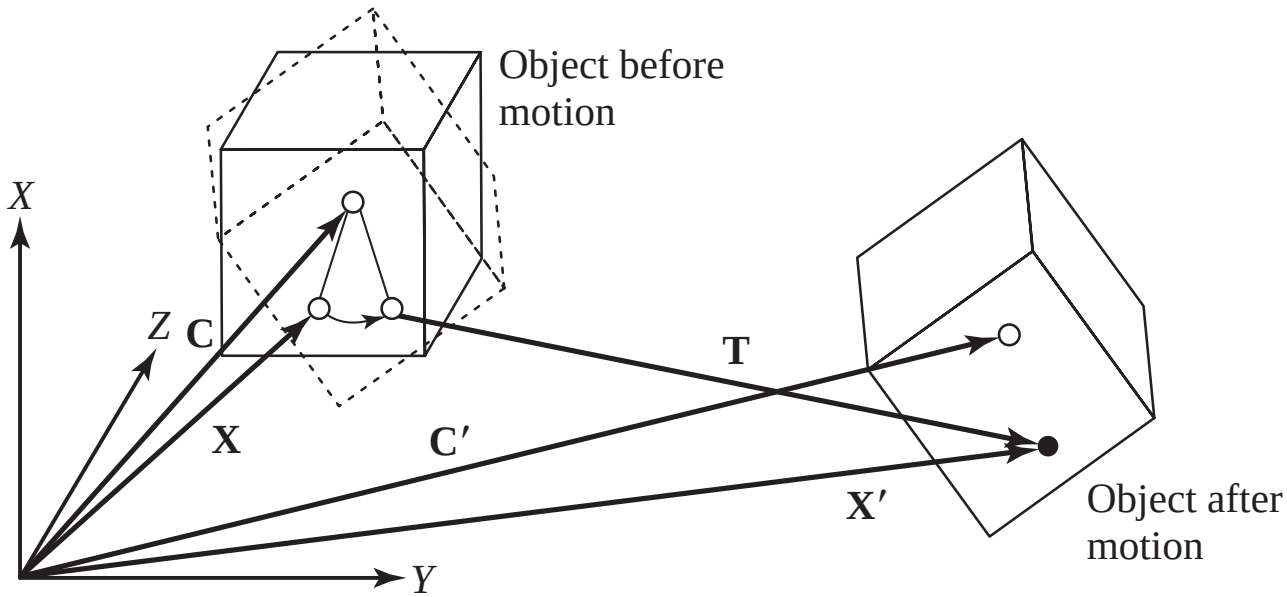
(b)

When the object is very far ($Z \to \infty$)

$x = X, y = Y$

Can be used as long as the depth variation within the object is small compared to the distance of the object.

Rigid Object Motion



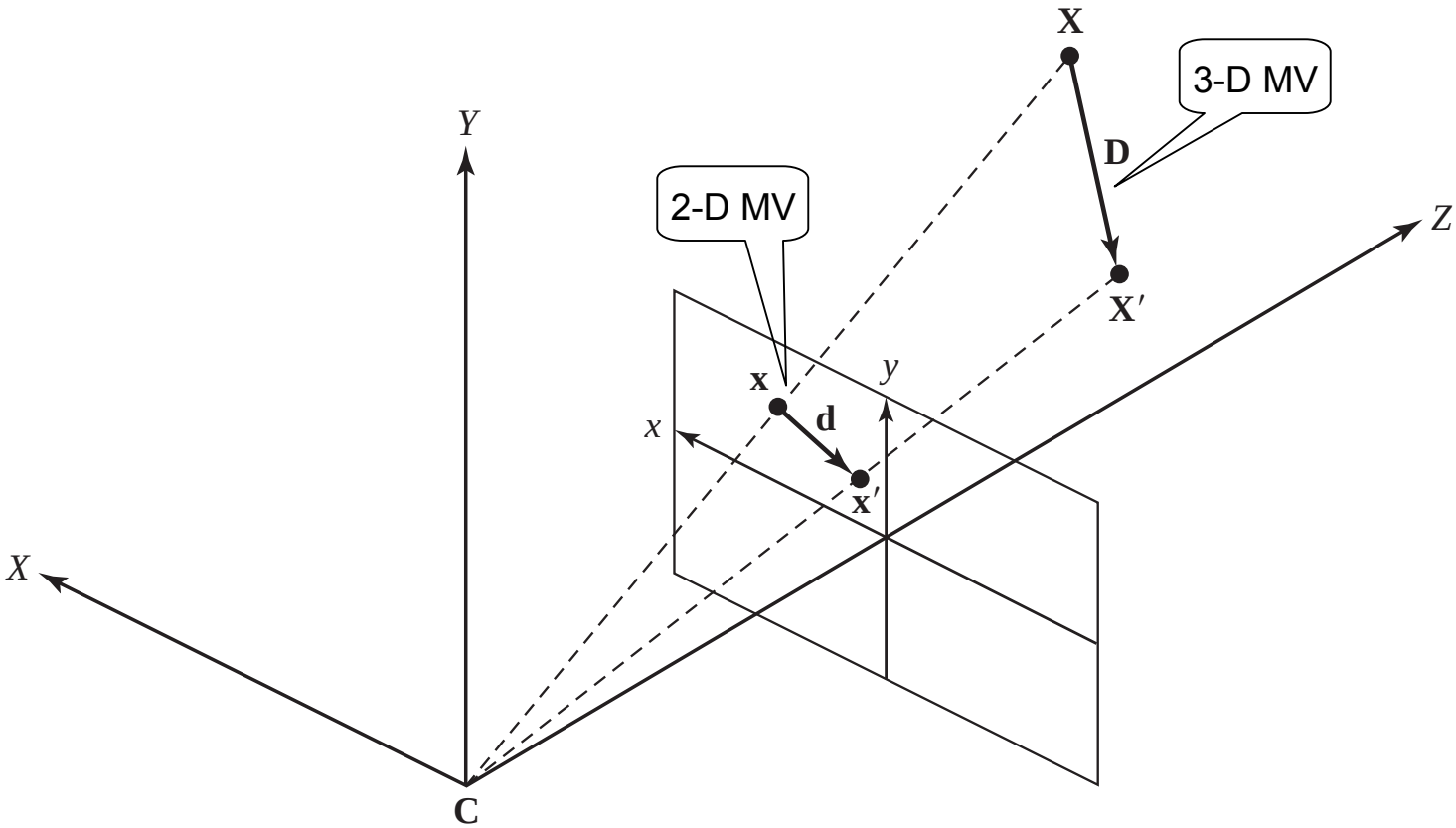
Rotation and translation wrp. the object center:

$$\mathbf{X}' = [\mathbf{R}](\mathbf{X} - \mathbf{C}) + \mathbf{T} + \mathbf{C}; \quad [\mathbf{R}]: \theta_x, \theta_y, \theta_z; \quad \mathbf{T}: T_x, T_y, T_z$$

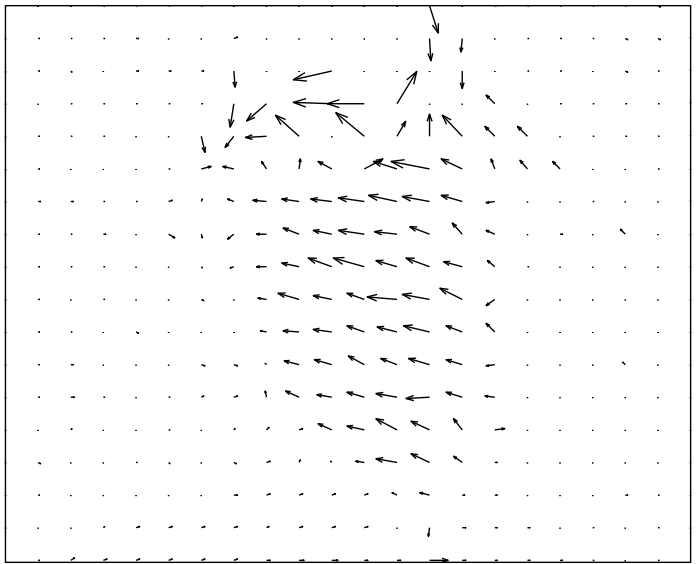
Flexible Object Motion

- Two ways to describe
  - Decompose into multiple, but connected rigid sub-objects
  - Global motion plus local motion in sub-objects
  - Ex. Human body consists of many parts each undergo a rigid motion

3-D Motion -> 2-D Motion

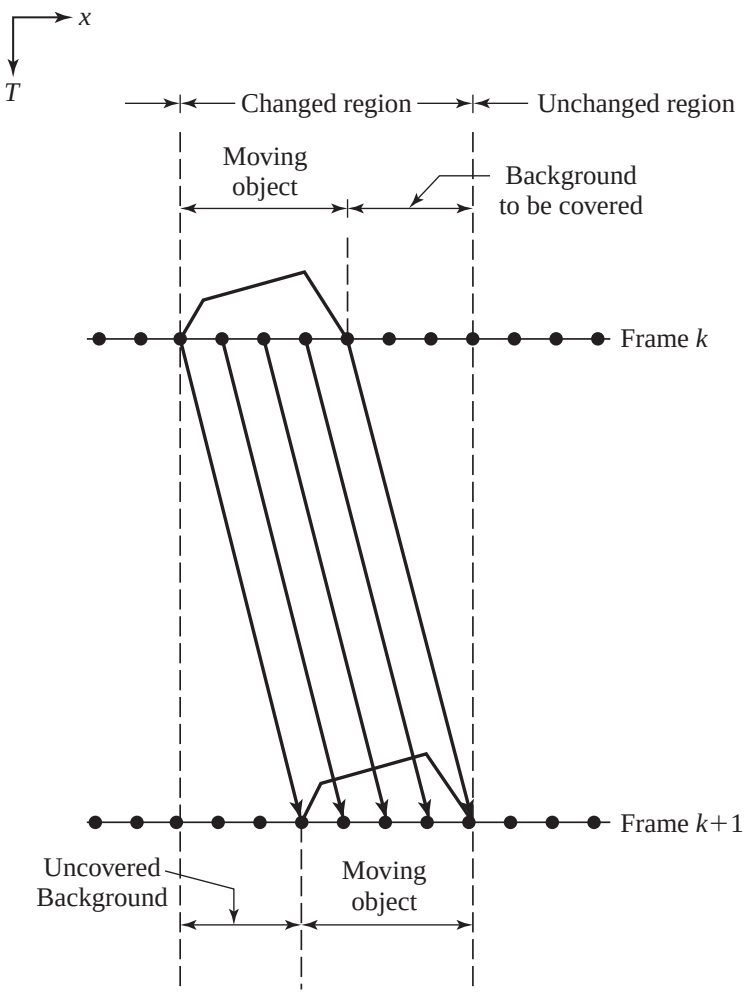


Sample Motion Field

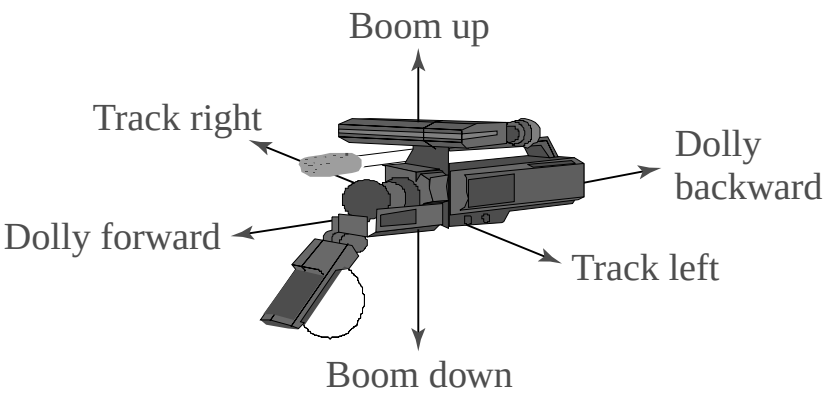


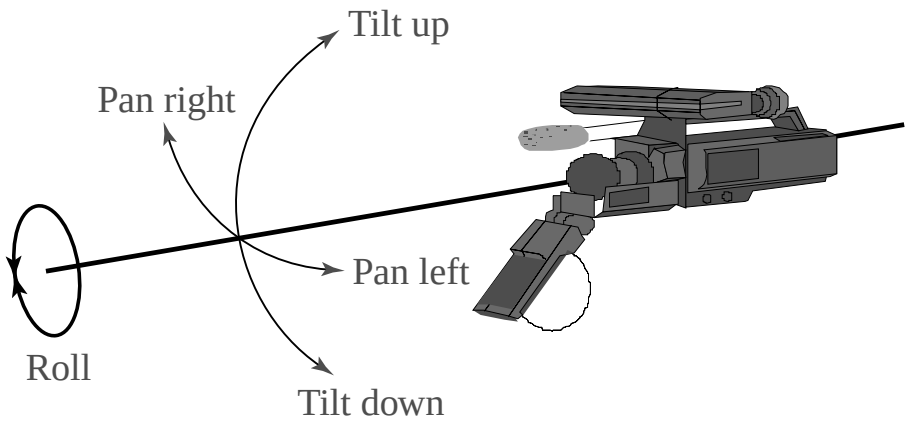
Occlusion Effect

Motion is undefined in occluded regions

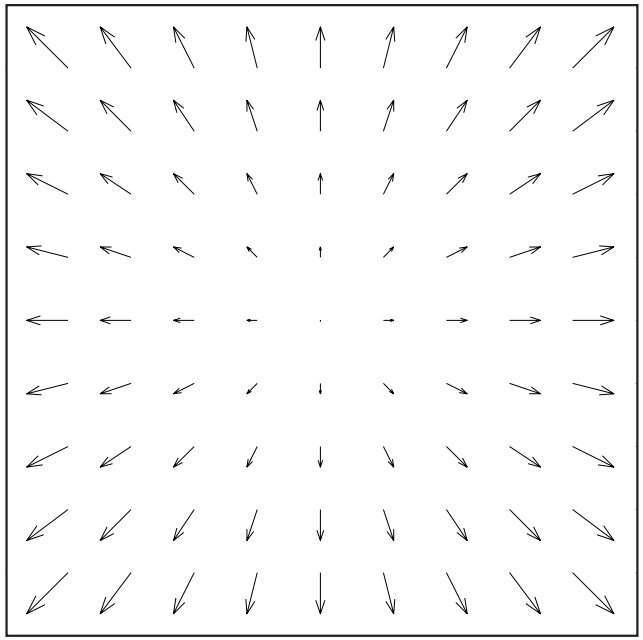


Typical Camera Motions



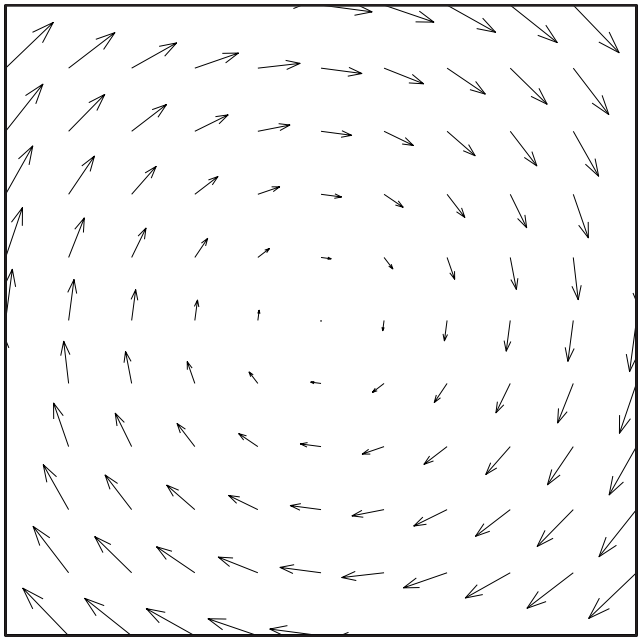


2-D Motion Corresponding to Camera Motion



(a)

Camera zoom



(b)

Camera rotation around Z-axis (roll)

2-D Motion Corresponding to Rigid Object Motion

- General case:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$$

$$\xrightarrow[\text{Perspective Projection}]{}$$

$$x' = F\frac{(r_1 x + r_2 y + r_3 F)Z + T_x F}{(r_7 x + r_8 y + r_9 F)Z + T_z F}$$

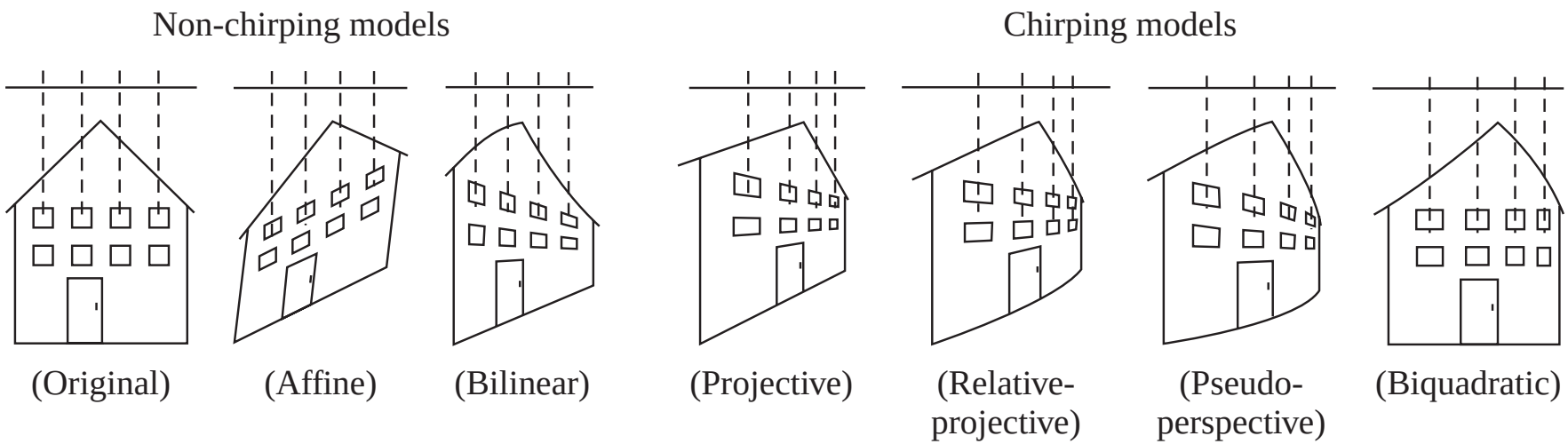
$$y' = F\frac{(r_4 x + r_5 y + r_6 F)Z + T_y F}{(r_7 x + r_8 y + r_9 F)Z + T_z F}$$

- Projective mapping:

When the object surface is planar $(Z = aX + bY + c)$:

$$x' = \frac{a_0 + a_1 x + a_2 y}{1 + c_1 x + c_2 y}, \quad y' = \frac{b_0 + b_1 x + b_2 y}{1 + c_1 x + c_2 y}$$

Projective Mapping



Two features of projective mapping:

- Chirping: increasing perceived spatial frequency for far away objects
- Converging (Keystone): parallel lines converge in distance

Affine and Bilinear Model

- Affine (6 parameters):

$$\begin{bmatrix} d_x(x,y) \\ d_y(x,y) \end{bmatrix} = \begin{bmatrix} a_0 + a_1 x + a_2 y \\ b_0 + b_1 x + b_2 y \end{bmatrix}$$

  – Good for mapping triangles to triangles

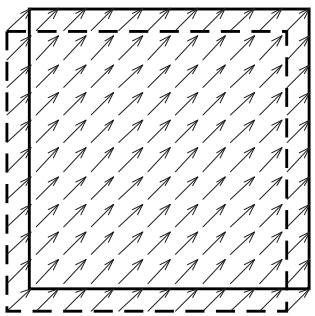
- Bilinear (8 parameters):

$$\begin{bmatrix} d_x(x,y) \\ d_y(x,y) \end{bmatrix} = \begin{bmatrix} a_0 + a_1 x + a_2 y + a_3 xy \\ b_0 + b_1 x + b_2 y + b_3 xy \end{bmatrix}$$

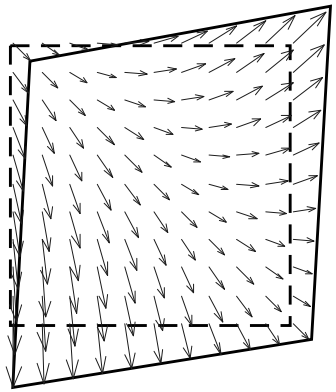
  – Good for mapping blocks to quadrangles

Motion Field Corresponding to Different 2-D Motion Models

Translation



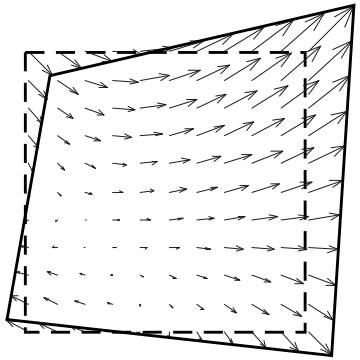
(a)



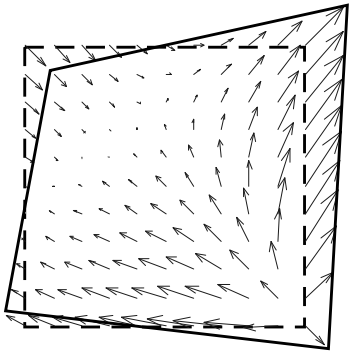
(b)

Affine

Bilinear



(c)

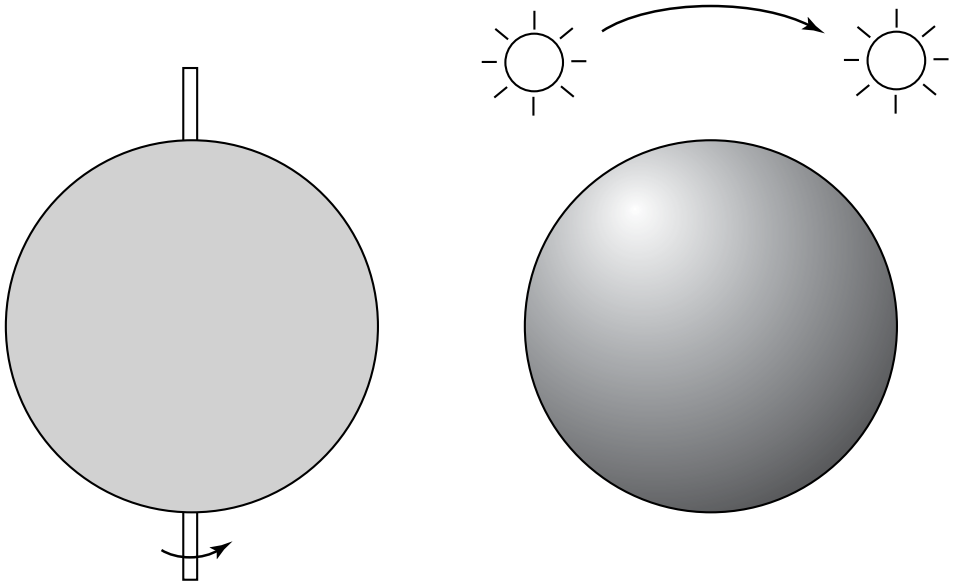


(d)

Projective

2-D Motion vs. Optical Flow

- 2-D Motion: Projection of 3-D motion, depending on 3D object motion and projection operator
- Optical flow: "Perceived" 2-D motion based on changes in image pattern, also depends on illumination and object surface texture



On the left, a sphere is rotating under a constant ambient illumination, but the observed image does not change.

On the right, a point light source is rotating around a stationary sphere, causing the highlight point on the sphere to rotate.

Optical Flow Equation

- When illumination condition is unknown, the best one can do it to estimate optical flow.
- Constant intensity assumption -> Optical flow equation

Under "constant intensity assumption":

$$\psi(x+d_x, y+d_y, t+d_t) = \psi(x,y,t)$$

But, using Taylor's expansion :

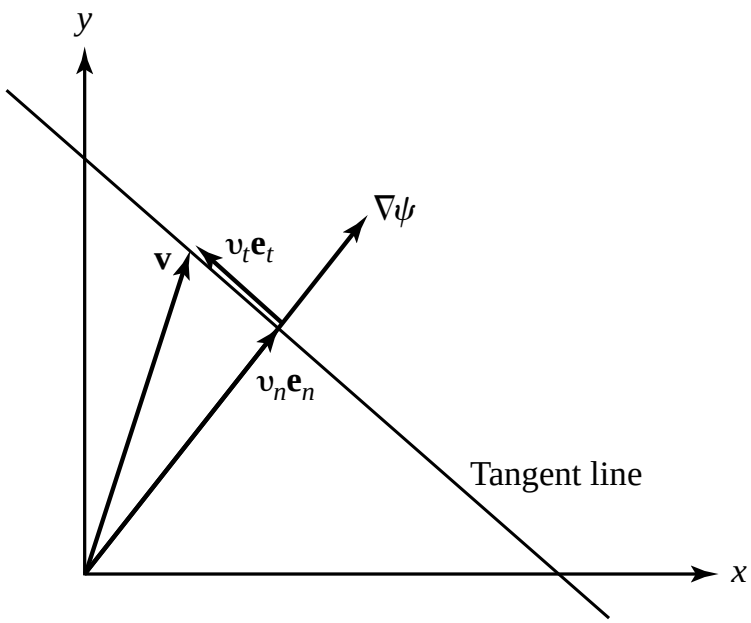
$$\psi(x+d_x, y+d_y, t+d_t) = \psi(x,y,t) + \frac{\partial \psi}{\partial x} d_x + \frac{\partial \psi}{\partial y} d_y + \frac{\partial \psi}{\partial y} d_t$$

Compare the above two, we have the optical flow equation :

$$\frac{\partial \psi}{\partial x} d_x + \frac{\partial \psi}{\partial y} d_y + \frac{\partial \psi}{\partial t} d_t = 0 \quad \text{or} \quad \frac{\partial \psi}{\partial x} v_x + \frac{\partial \psi}{\partial y} v_y + \frac{\partial \psi}{\partial t} = 0 \ \text{ or } \ \nabla \psi^T \mathbf{v} + \frac{\partial \psi}{\partial t} = 0$$

Ambiguities in Motion Estimation

- Optical flow equation only constrains the flow vector in the gradient direction v_n
- The flow vector in the tangent direction (v_t) is under-determined
- In regions with constant brightness ($\nabla \psi = 0$), the flow is indeterminate -> Motion estimation is unreliable in regions with flat texture, more reliable near edges

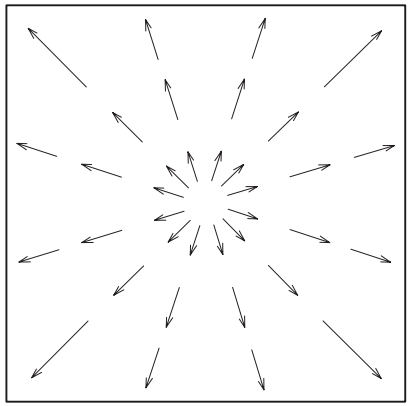


$$\mathbf{v} = v_n \mathbf{e}_n + v_t \mathbf{e}_t$$

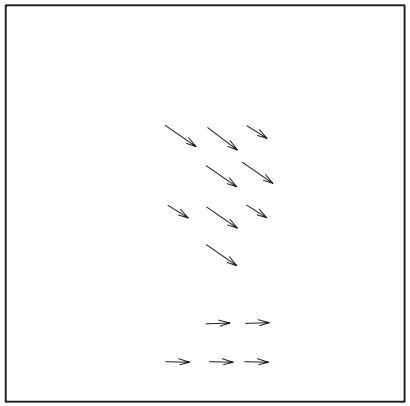
$$v_n \|\nabla \psi\| + \frac{\partial \psi}{\partial t} = 0$$

Motion Representation

Global:
Entire motion field is represented by a few global parameters



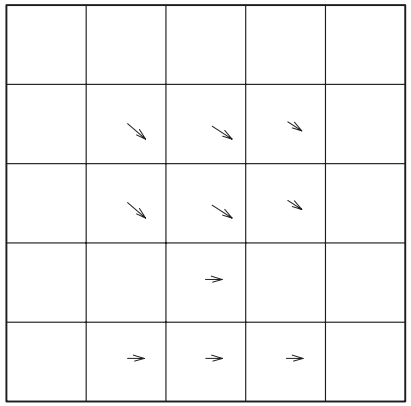
(a)



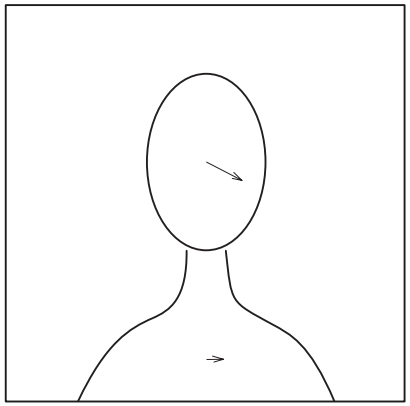
(b)

Pixel-based:
One MV at each pixel, with some smoothness constraint between adjacent MVs.

Block-based:
Entire frame is divided into blocks, and motion in each block is characterized by a few parameters.



(c)



(d)

Region-based:
Entire frame is divided into regions, each region corresponding to an object or sub-object with consistent motion, represented by a few parameters.

Other representation: mesh-based (control grid) (to be discussed later)

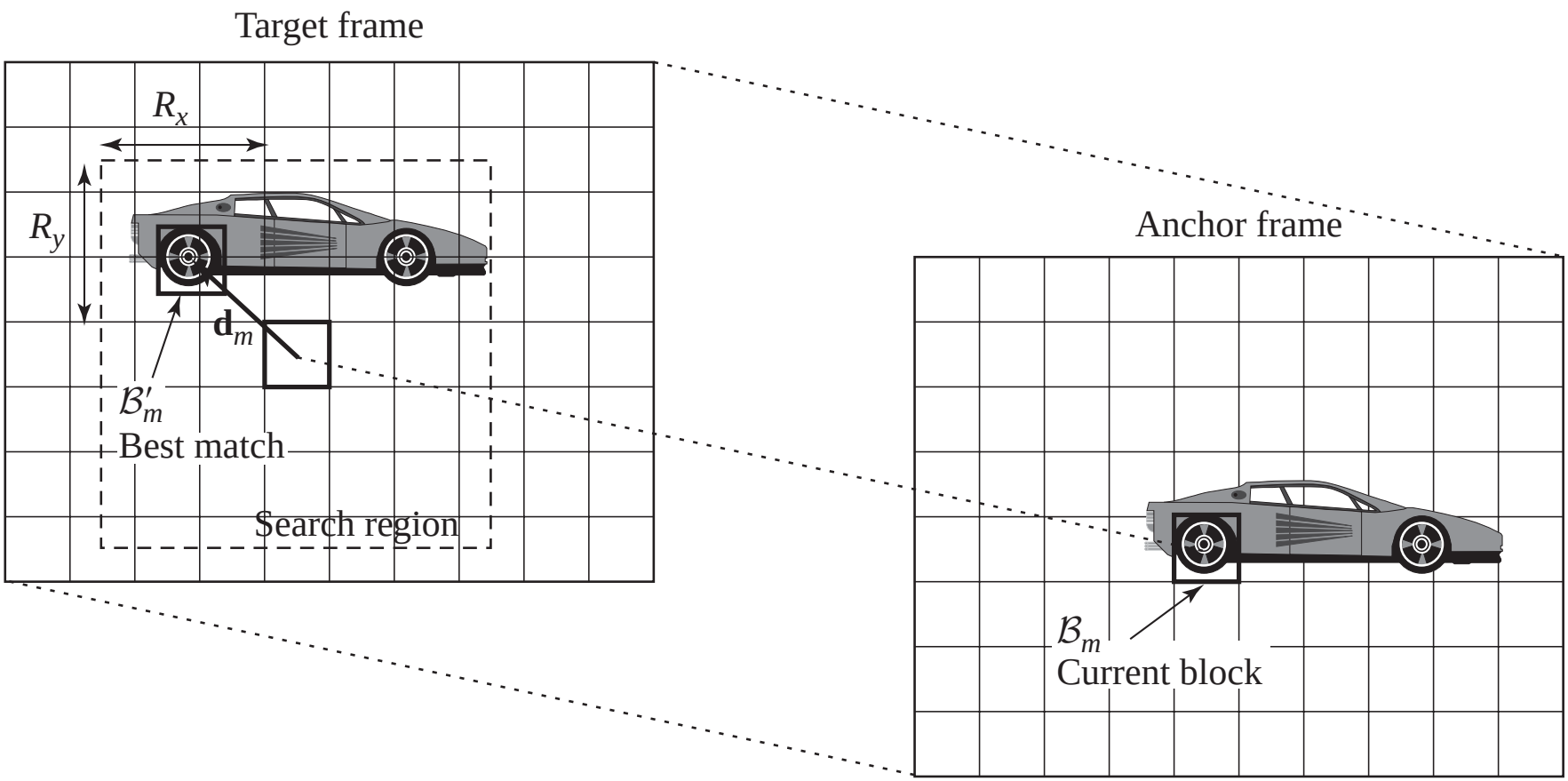
Block Matching Algorithm

- Overview:
  - Assume all pixels in a block undergo a translation, denoted by a single MV
  - Estimate the MV for each block independently, by minimizing the DFD error over this block
- Minimizing function:

$$E_{\mathrm{DFD}}(\mathbf{d}_{\mathrm{m}}) = \sum_{\mathbf{x}\in B_m} \left|\psi_2(\mathbf{x}+\mathbf{d}_m) - \psi_1(\mathbf{x})\right|^p \rightarrow \min$$

- Optimization method:
  - Exhaustive search (feasible as one only needs to search one MV at a time), using MAD criterion (p=1)
  - Fast search algorithms (not discussed here)
  - Integer vs. fractional pel accuracy search

Exhaustive Block Matching Algorithm (EBMA)



Sample Matlab Script for Integer-pel EBMA

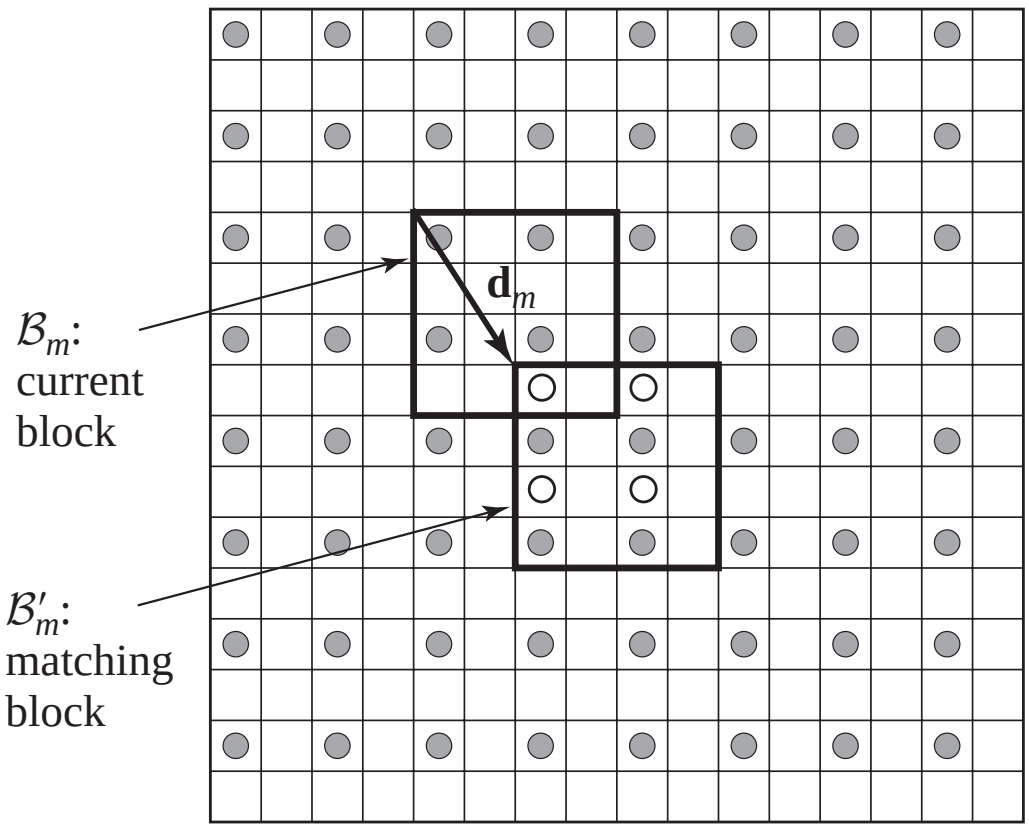
```
%f1: anchor frame; f2: target frame, fp: predicted image;
%mvx,mvy: store the MV image
%widthxheight: image size; N: block size, R: search range

for i=1:N:height-N,
  for j=1:N:width-N  %for every block in the anchor frame
     MAD_min=256*N*N;mvx=0;mvy=0;
     for k=-R:1:R,
       for l=-R:1:R  %for every search candidate
         MAD=sum(sum(abs(f1(i:i+N-1,j:j+N-1)-f2(i+k:i+k+N-1,j+l:j+l+N-1))));
               % calculate MAD for this candidate
         if MAD<MAX_min
               MAD_min=MAD,dy=k,dx=l;
         end;
     end;end;
     fp(i:i+N-1,j:j+N-1)= f2(i+dy:i+dy+N-1,j+dx:j+dx+N-1);
               %put the best matching block in the predicted image
     iblk=(floor)(i-1)/N+1; jblk=(floor)(j-1)/N+1; %block index
     mvx(iblk,jblk)=dx; mvy(iblk,jblk)=dy; %record the estimated MV
end;end;
```

Note: A real working program needs to check whether a pixel in the candidate matching block falls outside the image boundary and such pixel should not count in MAD. This program is meant to illustrate the main operations involved. Not the actual working matlab script.

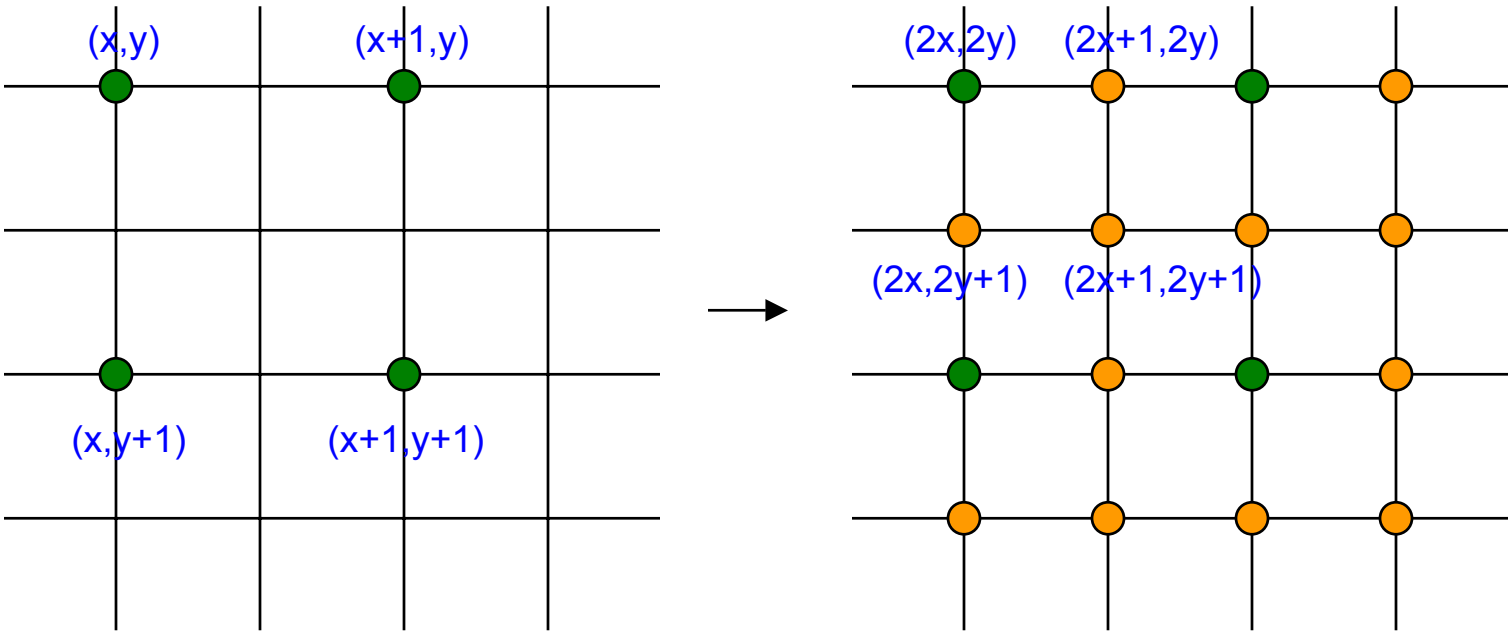
Fractional Accuracy EBMA

- Real MV may not always be multiples of pixels. To allow sub-pixel MV, the search stepsize must be less than 1 pixel.
- Half-pel EBMA: stepsize=1/2 pixel in both dimensions
- Difficulty:
  - Target frame only has integer pels
- Solution:
  - Interpolate the target frame by factor of two before searching
  - Bilinear interpolation is typically used
- Complexity:
  - 4 times of integer-pel, plus additional operations for interpolation.
- Fast algorithms:
  - Search in integer precisions first, then refine in a small search region in half-pel accuracy.

Half-Pel Accuracy EBMA



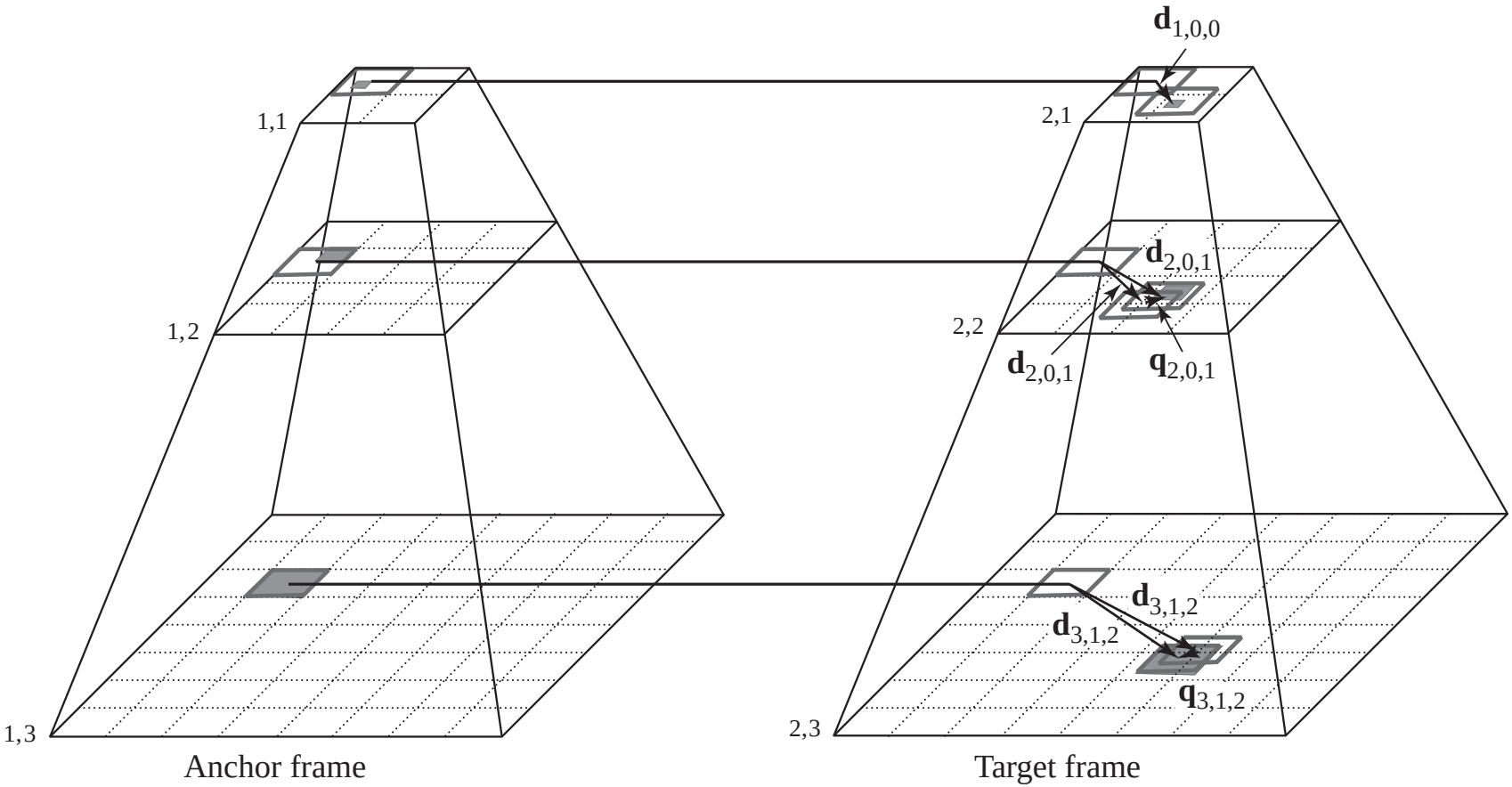
Bilinear Interpolation

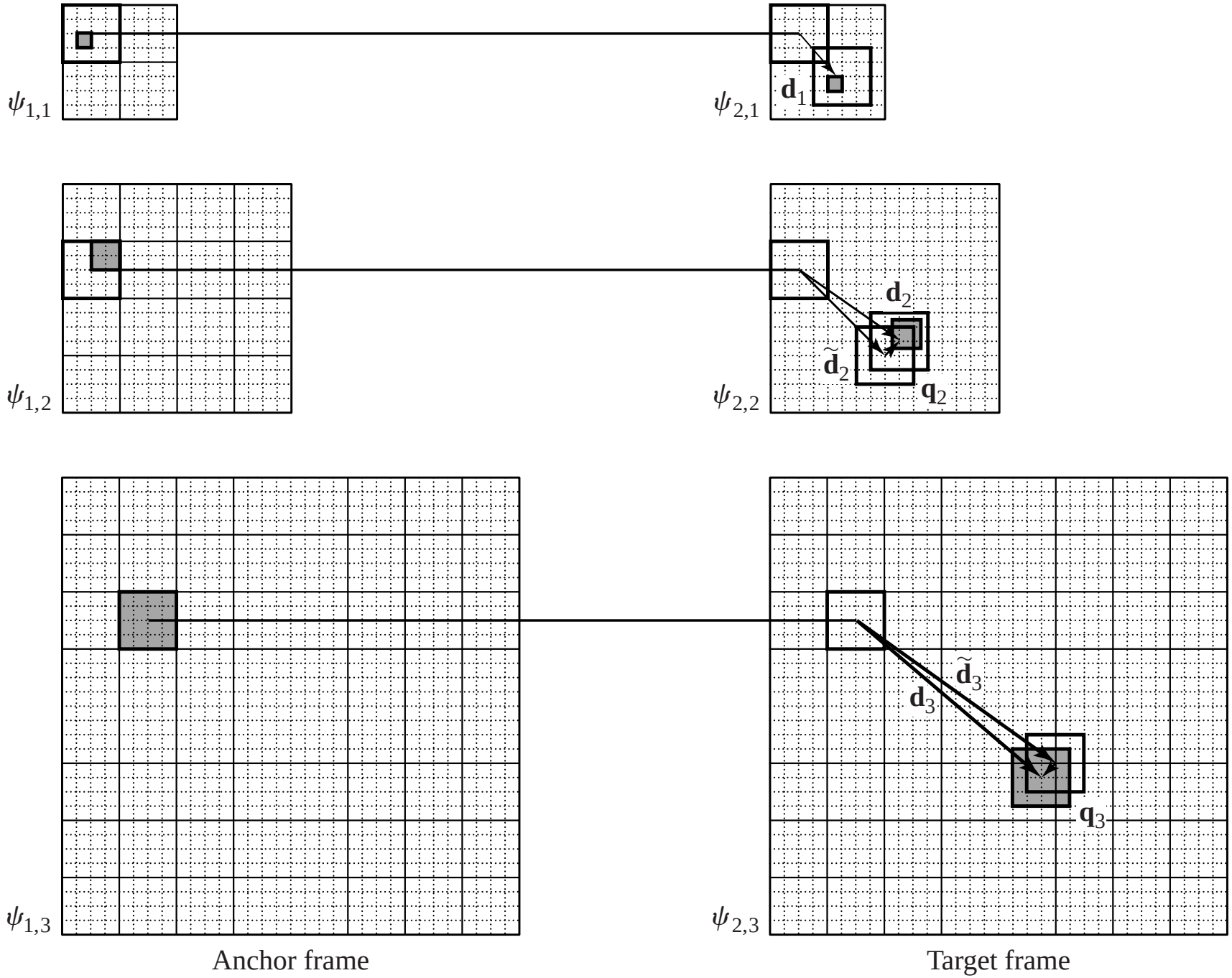


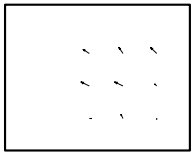
O[2x,2y]=I[x,y]
O[2x+1,2y]=(I[x,y]+I[x+1,y])/2
O[2x,2y+1]=(I[x,y]+I[x,y+1])/2
O[2x+1,2y+1]=(I[x,y]+I[x+1,y]+I[x,y+1]+I[x+1,y+1])/4

Example: Half-pel EBMA

Hierarchical Block Matching Algorithm (HBMA)

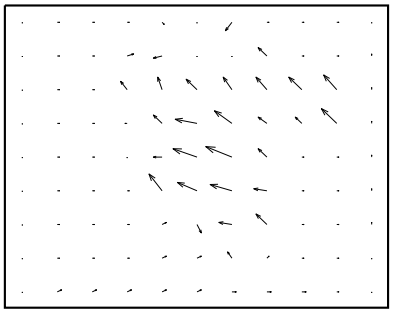



$\psi_{1,1}$
$\psi_{2,1}$
$\mathbf{d}_1$
$\psi_{1,2}$
$\psi_{2,2}$
$\mathbf{d}_2$
$\tilde{\mathbf{d}}_2$
$\mathbf{q}_2$
$\psi_{1,3}$
$\psi_{2,3}$
$\tilde{\mathbf{d}}_3$
$\mathbf{d}_3$
$\mathbf{q}_3$
Anchor frame
Target frame



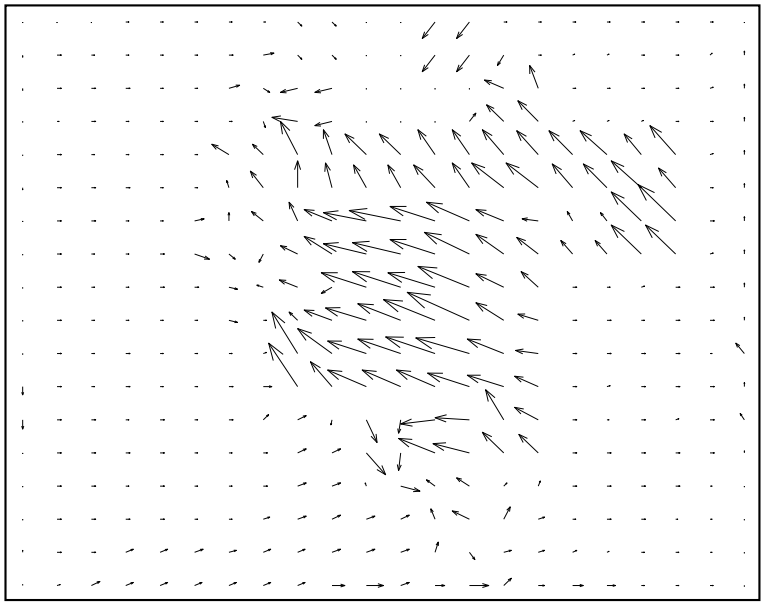
(a)

(b)



(c)

(d)



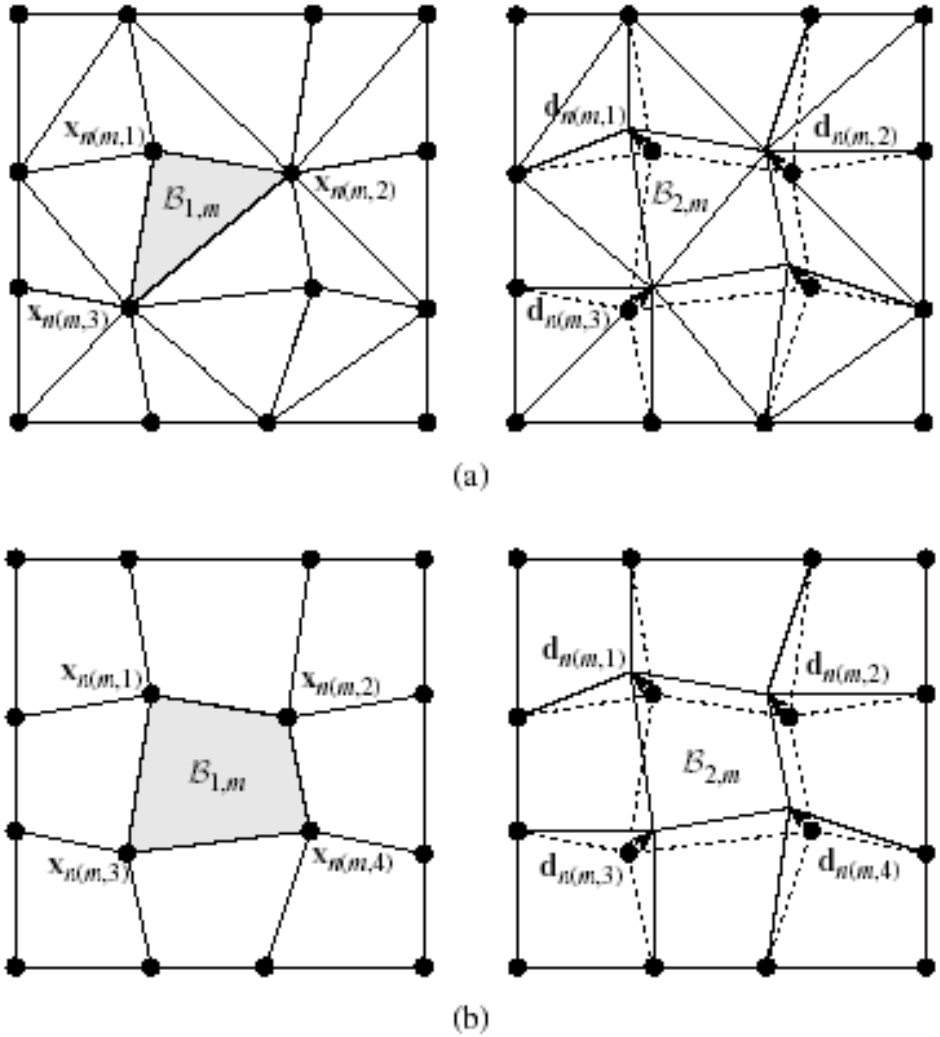
(e)

(f)

Predicted anchor frame (29.32dB)

Example: Three-level HBMA

Mesh-Based Motion Estimation: Overview

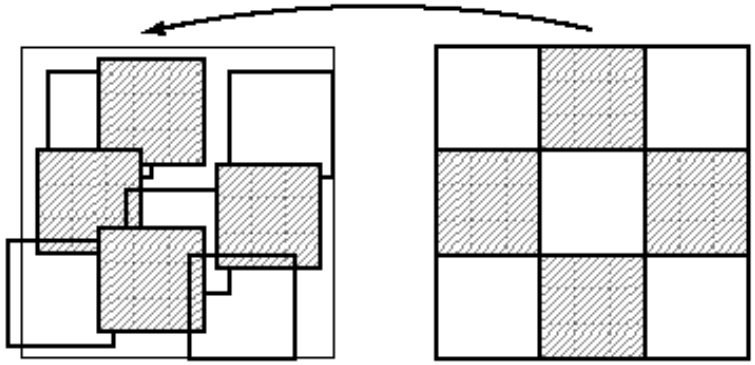


$$\mathbf{d}_m(\mathbf{x}) = \sum_{k \in \mathcal{K}} \phi_{m,k}(\mathbf{x}) \mathbf{d}_{n(m,k)}, \quad \mathbf{x} \in \mathcal{B}_{1,m},$$

(a) Using a triangular mesh

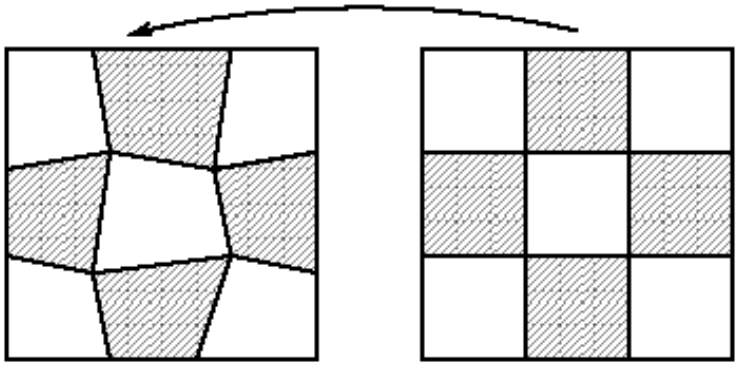
(b) Using a quadrilateral mesh

Mesh-based vs. block-based motion estimation



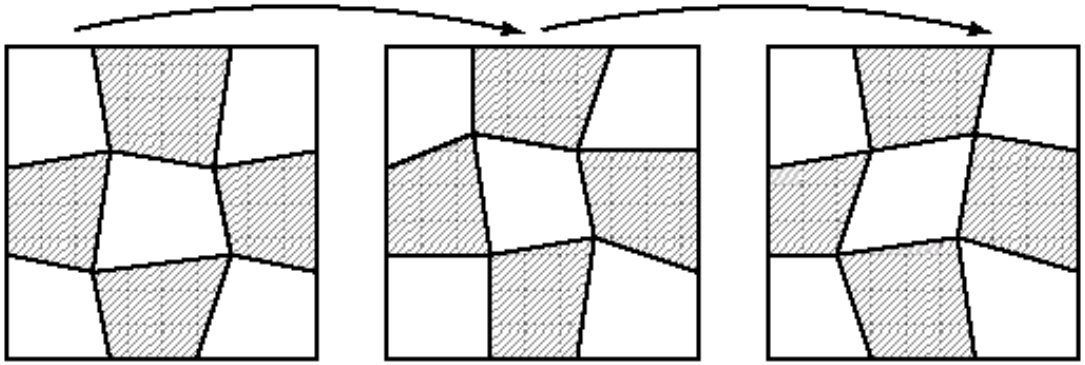
(a)

(a) block-based backward ME



(b)

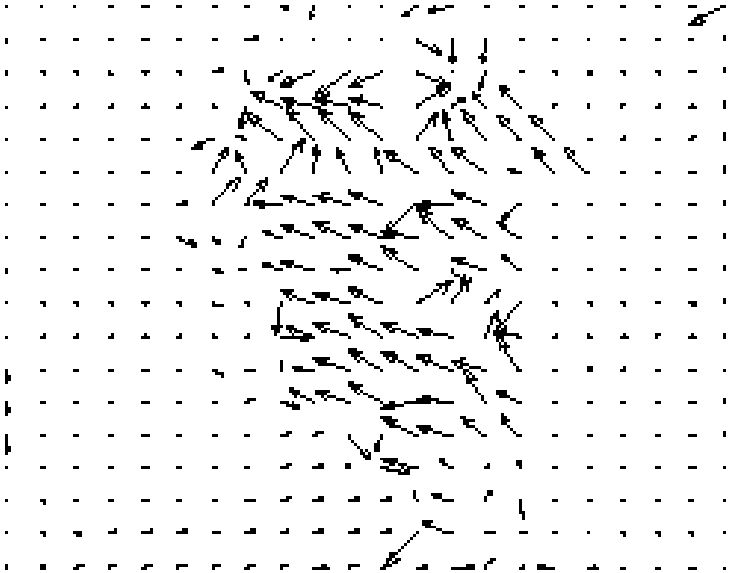
(b) mesh-based backward ME



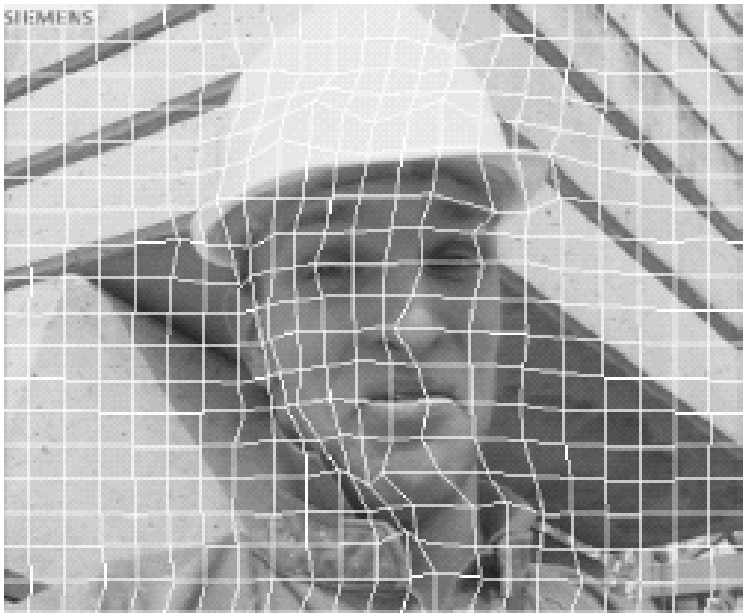
(c)

(c) mesh-based forward ME

Example: Half-pel EBMA



EBMA (29.86dB)



mesh-based method (29.72dB)

EBMA vs. Mesh-based Motion Estimation