ECE 634: Digital Video Systems
Motion estimation: 1/24/17

Professor Amy Reibman
MSEE 356
reibman@purdue.edu

http://engineering.purdue.edu/~reibman/ece634/index.html

Outline 1/24/17

- Motion estimation
- A computer assignment

Reading resources

- J. Konrad, "Motion Detection and Estimation", Chapter 3 in A. Bovik (ed.), *The Essential Guide to Video Processing*, Elsevier, 2009.
- A. M. Tekalp, Digital video processing, Prentice Hall, 1995
  - Chapter 5: 5.1,5.2
  - Chapter 6: 6.1, 6.3, 6.4
- Y. Wang, J. Ostermann, and Y.-Q. Zhang, *Video Processing and Communications*, Prentice Hall, 2002.
  - Chapter 5.1, 5.3.2, 5.5: Video Modeling
  - Chapter 6.1-6.4, 6.7, 6.9, skip Sec. 6.4.5, 6.4.6: Two-dimensional motion estimation
  - Appendix A and B: Gradients and steepest descent
- R. Szeliski, *Computer Vision: Algorithms and Applications*, Springr 2010, Chapter 8

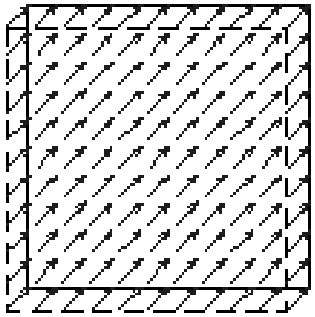
Summary (last class)

- 3D Motion
  - Rigid vs. non-rigid motion
- Camera model: 3D → 2D projection
  - Perspective projection vs. orthographic projection
- What causes 2D motion?
  - Object motion projected to 2D
  - Camera motion
- Models corresponding to typical camera motion and object motion
  - Piece-wise projective mapping is a good model for projected rigid object motion
  - Can be approximated by affine or bilinear functions
  - Affine functions can also characterize some global camera motions
- Ways to represent motion:
  - Pixel-based, block-based, region-based, global, etc.

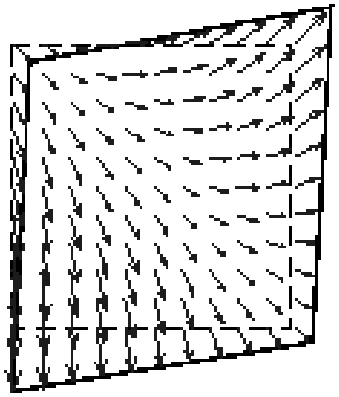
©Yao Wang, 2003

Motion Field Corresponding to Different 2-D Motion Models

Translation



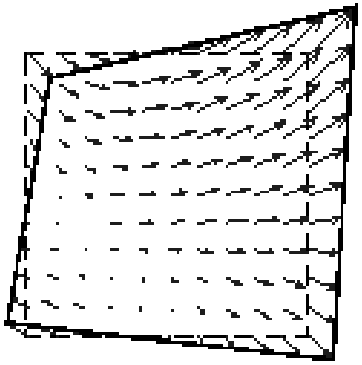
(a)



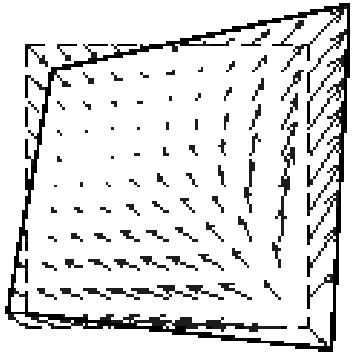
(b)

Affine

Bilinear



(c)



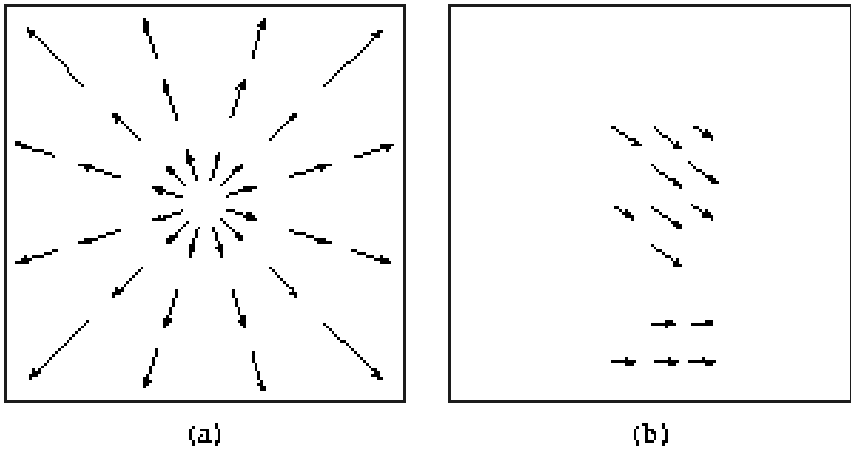
(d)

Projective

©Yao Wang, 2003

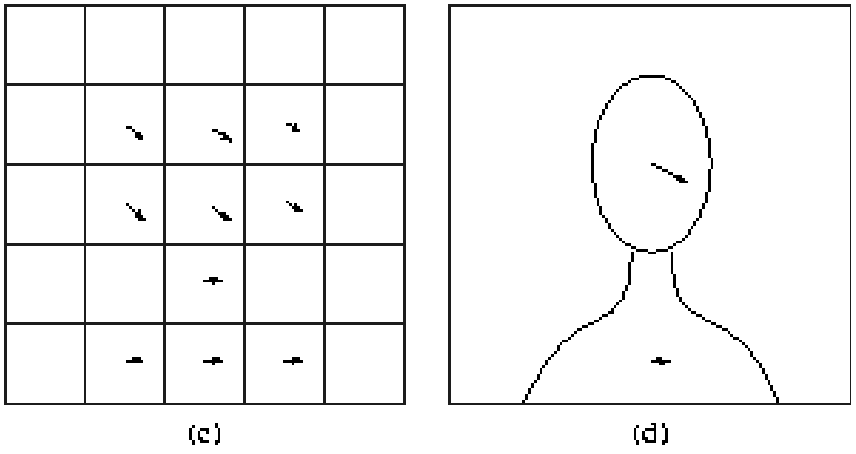
Region of support for representation of motion

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



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.



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.

©Yao Wang, 2003

Motion estimation algorithms

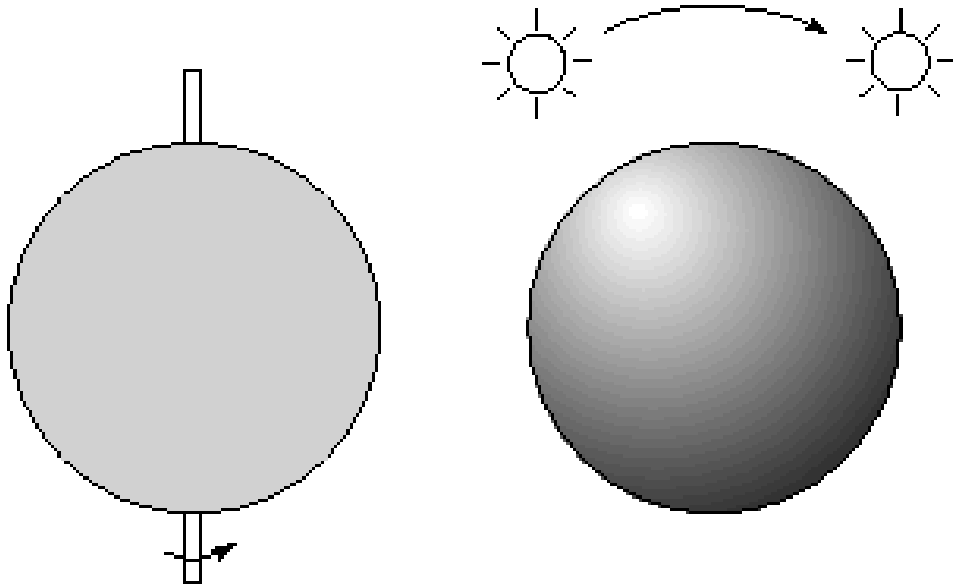
- Motion representation
  - Pixel-based, block-based, mesh, global motion, ..
- Optimization criteria
  - Minimize displaced frame difference, optical flow, while subject to constraints ..
- Optimization strategies
  - Gradient descent, exhaustive search, ..

Motion estimation outline

- 2D motion and optical flow
- Motion estimation
  - General methodologies
  - Pixel-based
  - Block-based

Apparent motion, or optical flow

- 2D velocity (object motion projected onto the image plane) is **NOT** the same as optical flow
- Example 1: constantly lit sphere rotating
- Example 2: still sphere with changing lighting



©Yao Wang, 2003

Optical flow derivation (1)

- Constant intensity assumption (ambient and diffuse lighting)

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

- Taylor's expansion

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

- So clearly

$$\frac{\partial \psi}{\partial x} d_x + \frac{\partial \psi}{\partial y} d_y + \frac{\partial \psi}{\partial t} d_t = 0$$

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

Assuming d_t is small so that $v_x = d_x/d_t$

Optical flow derivation (2)

- Constant intensity assumption

$$\frac{d\psi(x,y,t)}{dt}=0$$

- Apply chain rule

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

Two approaches, same assumption, same answer

Flow equation

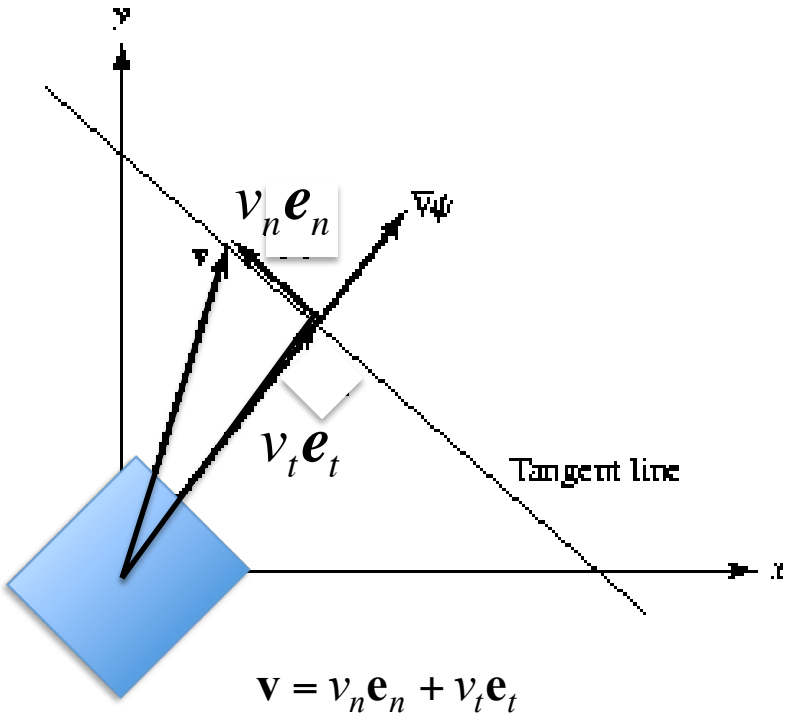
$$\frac{\partial \psi}{\partial x} v_x + \frac{\partial \psi}{\partial y} v_y + \frac{\partial \psi}{\partial t} = 0 \quad \text{or} \quad \nabla \psi^T \mathbf{v} + \frac{\partial \psi}{\partial t} = 0$$

- Can only estimate motion in direction of the spatial gradient

- Applies to a single point only

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



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

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

Inaccuracies

- Object boundaries
  - Motion estimation more reliable around strong edges, but strong edges are likely to be where two objects move differently
- Occlusion
  - No correspondence exists for (un)covered background
- The aperture problem
  - This is an underconstrained problem; one equation, two unknowns. Tangent direction is undetermined
  - “Aperture” must contain at least 2 different gradient directions

Flow equation

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

- May not hold exactly for real images
  - Noise, aliasing, illumination variations, ..
- Instead, minimize some function of

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

Two categories of approaches for Motion Estimation

- Feature based (more often used in object tracking, 3D reconstruction from 2D)
  - Find motion only for sparse points
  - Impose a motion model to estimate a dense field
- Intensity based (based on constant intensity assumption) (more often used for motion compensated prediction, required in video coding, frame interpolation)

General Considerations for Motion Estimation

- Three important questions
  - How to represent the motion field? (ex: dense or sparse? Region or)
  - What optimization criteria to use to estimate motion parameters?
    - Depends on the application; compression minimize average prediction error; motion-compensated interpolation minimize maximum interpolation error
  - How to search for the best motion parameters?

Mix-and-match

- Motion representation
- Optimization criteria
- Optimization strategies

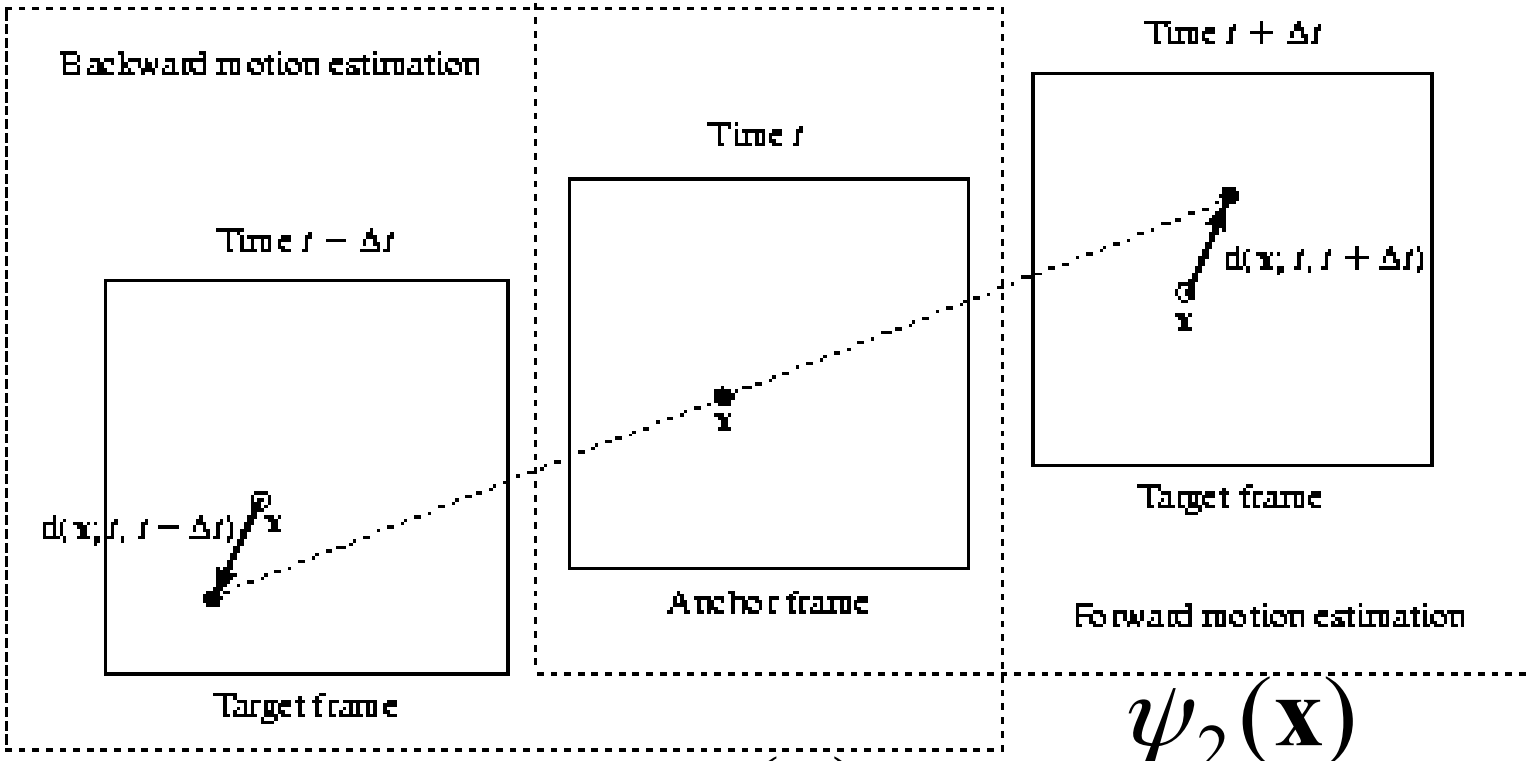
- Examples:
  - Pixel-based representation, DFD, gradient descent
  - Pixel-based representation, OF, least-square

  - Block-based representation, DFD, exhaustive search
  - Block-based representation, DFD, hierarchical search

  - Mesh representation, DFD, iterative search
  - Global motion

Notations

Motion parameters: $\mathbf{a}$ Motion field: $\mathbf{d}(\mathbf{x};\mathbf{a}), \mathbf{x} \in \Lambda$



$\psi_2(\mathbf{x})$ $\psi_1(\mathbf{x})$ $\psi_2(\mathbf{x})$

Mapping function:

$$\mathbf{w}(\mathbf{x};\mathbf{a}) = \mathbf{x} + \mathbf{d}(\mathbf{x};\mathbf{a}), \mathbf{x} \in \Lambda$$

©Yao Wang, 2003

Motion Estimation Criteria

- Minimize the displaced frame difference (DFD)

$$E_{\mathrm{DFD}}(\mathbf{a}) = \sum_{x\in\Lambda} \left|\psi_2(\mathbf{x}+\mathbf{d}(\mathbf{x};\mathbf{a})) - \psi_1(\mathbf{x})\right|^p \rightarrow \min$$

$$p = 1: \mathrm{MAD}; \quad P = 2: \mathrm{MSE}$$

- Satisfy the optical flow (OF) equation

$$E_{\mathrm{OF}}(\mathbf{a}) = \sum_{x\in\Lambda} \left|\left(\nabla\psi_1(\mathbf{x})\right)^T \mathbf{d}(\mathbf{x};\mathbf{a}) + \psi_2(\mathbf{x}) - \psi_1(\mathbf{x})\right|^p \rightarrow \min$$

- Impose additional smoothness constraint using regularization technique (Important in pixel- and block-based representation)

$$E_s(\mathbf{a}) = \sum_{\mathbf{x}\in\Lambda} \sum_{\mathbf{y}\in N_x} \left\|\mathbf{d}(\mathbf{x};\mathbf{a}) - \mathbf{d}(\mathbf{y};\mathbf{a})\right\|^2$$

$$w_{DFD} E_{\mathrm{DFD}}(\mathbf{a}) + w_s E_s(\mathbf{a}) \rightarrow \min$$

©Yao Wang, 2003

Optimization Strategies to find Min. or Max.

- Exhaustive search
  - Typically used for the DFD criterion with p=1 (MAD)
  - Guarantees reaching the global optimal
  - Computation required may be unacceptable when there are many parameters to search simultaneously!
  - Fast search algorithms reach sub-optimal solution in shorter time
- Gradient-based search
  - Typically used for the DFD or OF criterion with p=2 (MSE)
    - the gradient can often be calculated analytically
    - When used with the OF criterion, closed-form solution may be obtained
  - Reaches the local optimal point closest to the initial solution
- Multi-resolution search
  - Search from coarse to fine resolution, faster than exhaustive search
  - Less likely to be trapped into a local minimum

©Yao Wang, 2003

High-level Framework

- Motion representation
- Optimization criteria
- Optimization strategies

- Mix and match
  - Pixel-based representation, DFD, gradient descent
  - Pixel-based representation, OF, least-squares

  - Block-based representation, DFD, exhaustive search
  - Block-based representation, DFD, hierarchical search

  - Mesh representation, DFD, iterative search
  - Global motion

Block Matching Algorithm

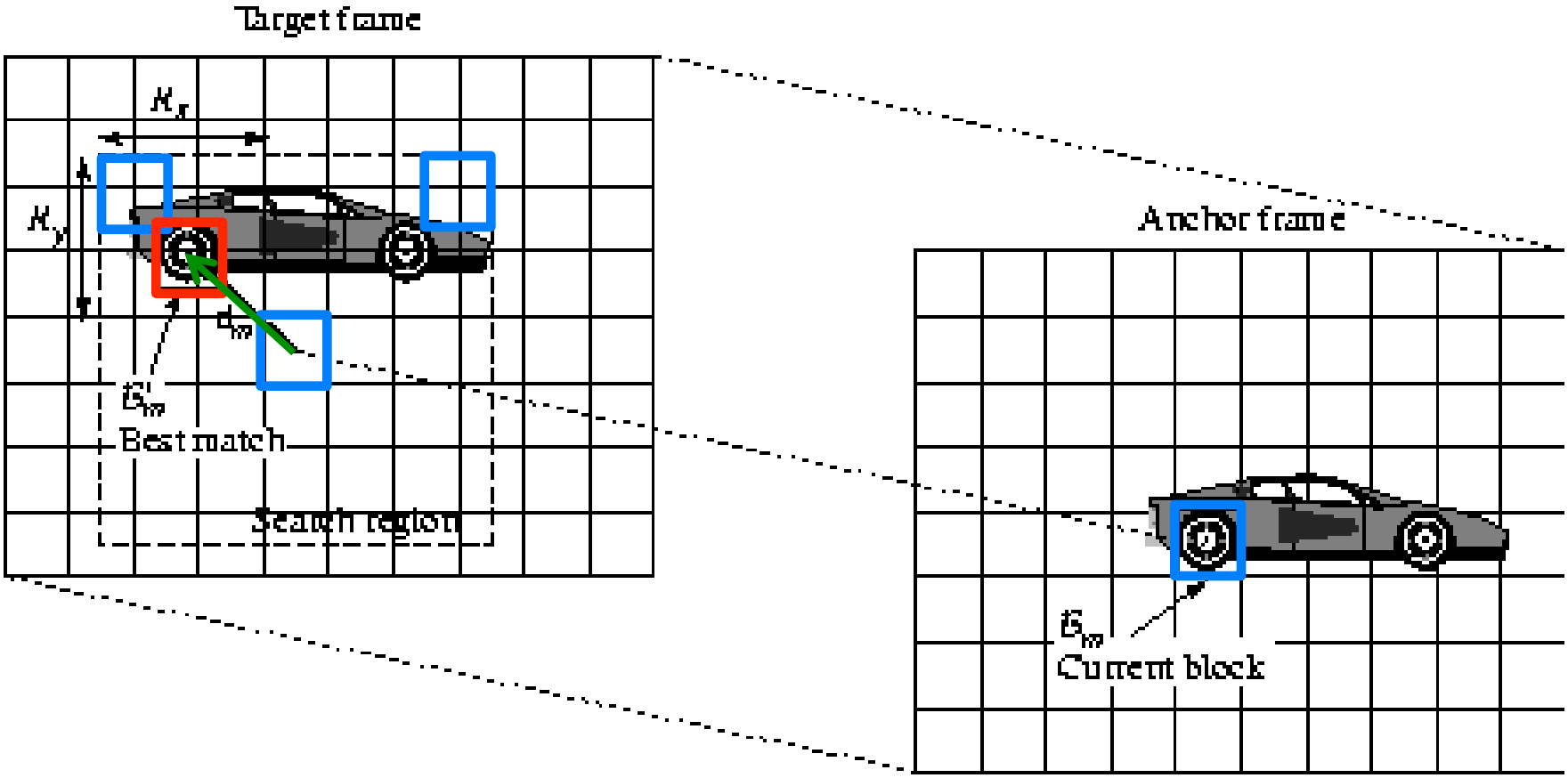
- Overview:
  - Assume all pixels in a block undergo a translation, denoted by a single MV
  - Estimate the MV for each block independently, by minimizing the DFD error over this block
- Minimizing function:

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

- Optimization method:
  - Exhaustive search (feasible as one only needs to search one MV for all pixels in the block), using MAD criterion (p=1)
  - Fast search algorithms
  - Integer vs. fractional pel accuracy search

©Yao Wang, 2003

Exhaustive Block Matching Algorithm (EBMA)



Complexity of Integer-Pel EBMA

- Assumption
  - Image size: MxM
  - Block size: NxN
  - Search range: (-R,R) in each dimension
  - Search stepsize: 1 pixel (assuming integer MV)
- Operation counts (1 operation=1 "-", 1 "+", 1 "*"):
  - Each candidate position: *N^2*
  - Each block going through all candidates: *(2R+1)^2 N^2*
  - Entire frame: *(M/N)^2 (2R+1)^2 N^2=M^2 (2R+1)^2*
    - Independent of block size!
- Example: M=512, N=16, R=16, 30 fps
  - Total operation count = 2.85x10^8/frame =8.55x10^9/second
- Regular structure suitable for VLSI implementation
- Software-only implementation slow

©Yao Wang, 2003

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

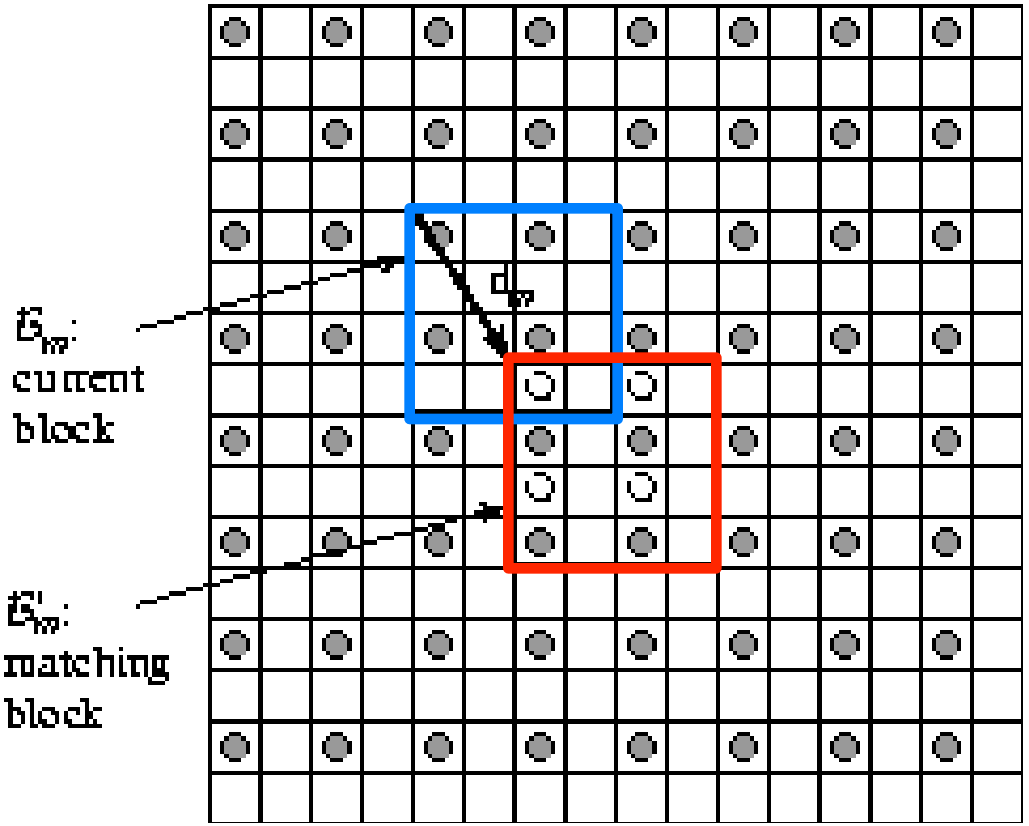
Note: A real working program needs to check whether a pixel in the candidate matching block falls outside the image boundary and such pixel should not count in MAD. This program is meant to illustrate the main operations involved. Not the actual working matlab script.

©Yao Wang, 2003

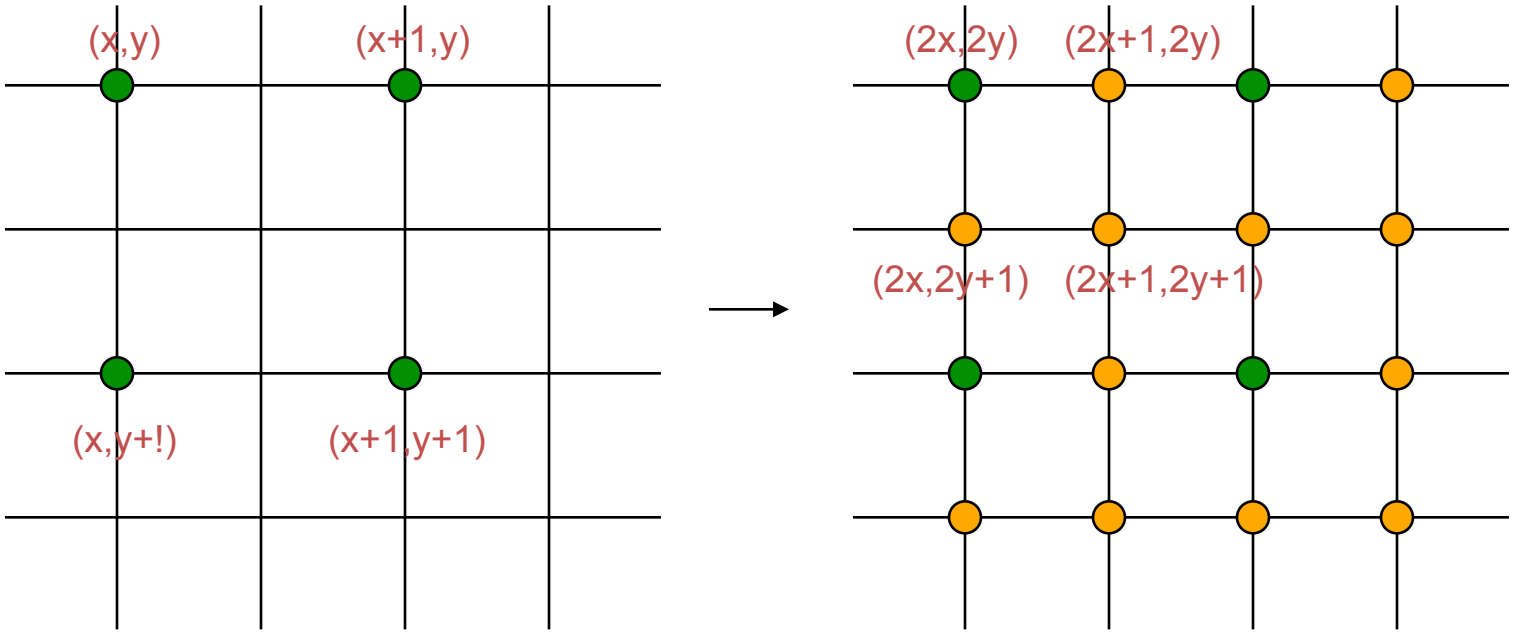
Fractional Accuracy EBMA

- 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 dimension
- Difficulty:
  - Target frame only have 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+1,y])/2
O[2x+1,2y+1]=(I[x,y]+I[x+1,y]+I[x,y+1]+I[x+1,y+1])/4

©Yao Wang, 2003

target frame

anchor frame

Motion field

Predicted anchor frame (29.86dB)

Example: Half-pel EBMA

Problems with EBMA

- Motion field is chaotic
  - Each block's motion vector is computed independently
  - Many possible matches, especially in smooth regions
- DFD is not uniformly small within block
  - Poor motion model:
    - Block may contain multiple motions
    - Block does not undergo translation
    - Illumination changes
- DFD is not uniformly small across block boundaries
  - Poor motion model: Adjacent pixels can have very different motions
- Slow

Minimizing problems with EBMA

- Motion field is chaotic
  - Use hierarchical search
  - Impose smoothness constraints (including mesh-based model)
- DFD is not uniformly small within block
  - Improve motion model (Deformable and mesh-based models; region-based estimation; compensate for variable illumination)
- DFD is not uniformly small across block boundaries
  - Mesh-based motion models; compute pixel-based motion
- Slow
  - Use fast algorithms and hierarchical search

Fast Algorithms for BMA

- Two key ideas to reduce the computation in EBMA
  - Reduce # of search candidates:
    - Only search for those that are likely to produce small errors.
    - Predict possible remaining candidates, based on previous search result
  - Reduce the computation for each candidate by simplifying the DFD error measure
    - Subsample and don't compute DFD on all possible pixels
- Many many fast algorithms
  - Three-step
  - 2D-log
  - Conjugate direction

©Yao Wang, 2003

Summary (so far this class)

- Constraints for 2D motion
  - Optical flow equation
  - Derived from constant intensity and small motion assumption
  - Ambiguity in motion estimation
- Estimation criterion:
  - DFD (constant intensity)
  - OF (constant intensity+small motion)
- Search method:
  - Exhaustive search, gradient-descent, multi-resolution
- Pixel-based motion estimation
  - Most accurate representation, but also most costly to estimate
- Block-based motion estimation
  - Good trade-off between accuracy and speed
  - EBMA and its fast but suboptimal variants are widely used in video coding for motion-compensated temporal prediction.

Computer assignments

- **Due Monday 2/6/17 at 7am**
  - See assignment 1 posted online

Optimization Strategies to find Min. or Max.

- Exhaustive search
  - Typically used for the DFD criterion with p=1 (MAD)
  - Guarantees reaching the global optimal
  - Computation required may be unacceptable when there are many parameters to search simultaneously!
  - Fast search algorithms reach sub-optimal solution in shorter time
- Gradient-based search
  - Typically used for the DFD or OF criterion with p=2 (MSE)
    - the gradient can often be calculated analytically
    - When used with the OF criterion, closed-form solution may be obtained
  - Reaches the local optimal point closest to the initial solution
- Multi-resolution search
  - Search from coarse to fine resolution, faster than exhaustive search
  - Less likely to be trapped into a local minimum

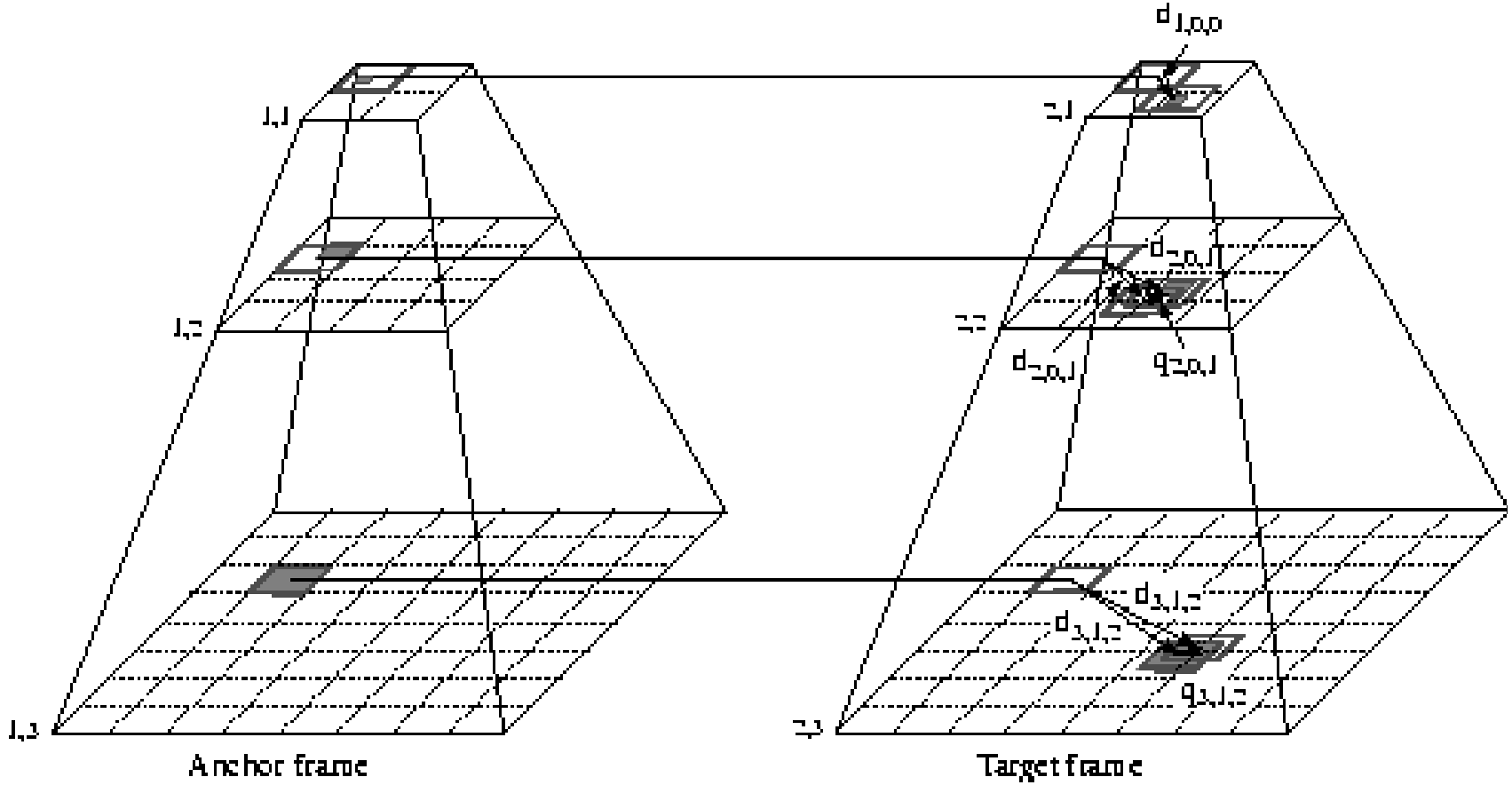
©Yao Wang, 2003

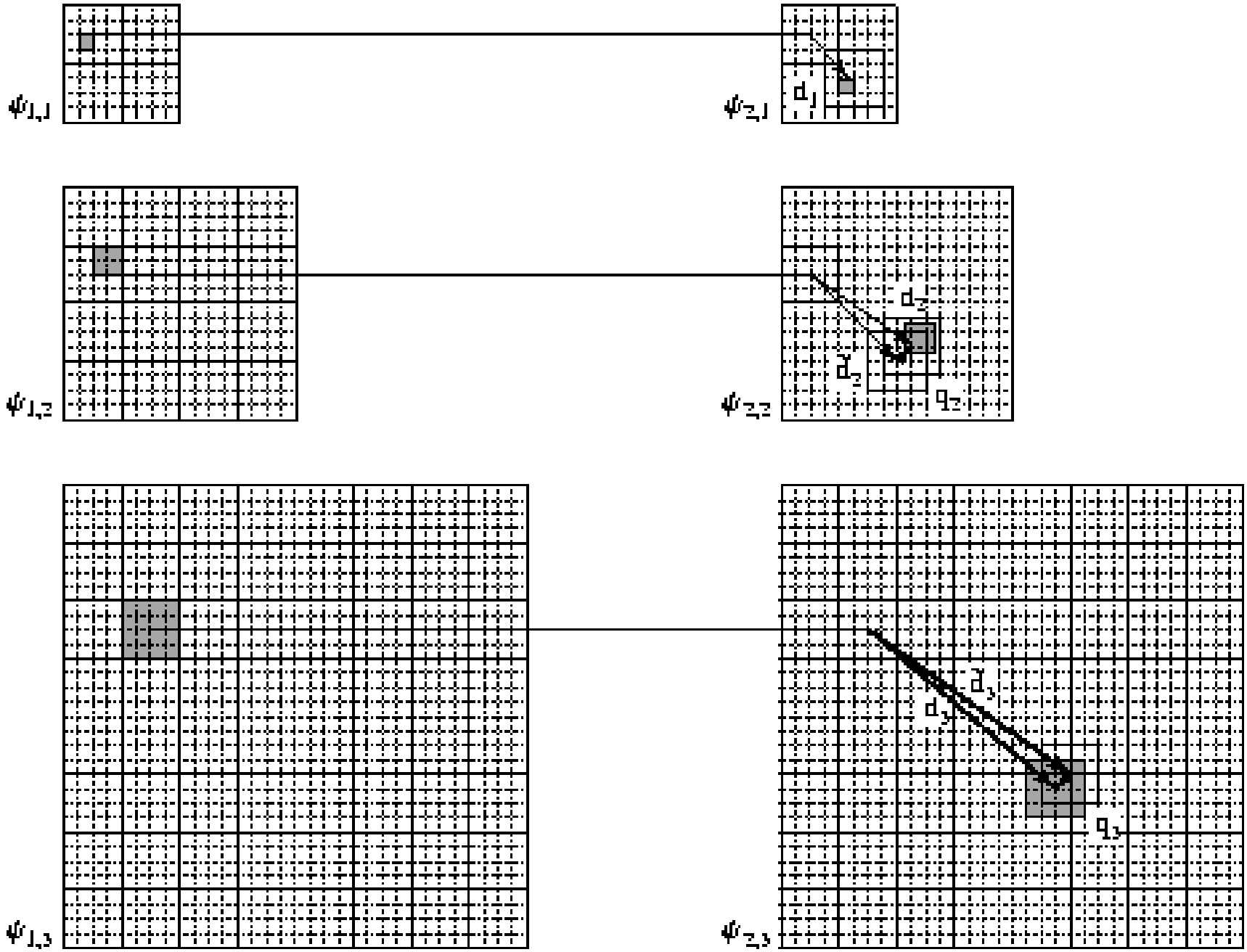
Multi-resolution Motion Estimation

- Goal: Reduce computation and approach globally minimal solution
- First: Estimate the motion in a coarse resolution over low-pass filtered, down-sampled image pair
  - May lead to a solution closer to the true motion field
- Second: Modify the initial solution in successively finer resolution within a small search range
  - Reduces the amount of computation
- Can be applied to different motion representations, but we will focus on its application to BMA

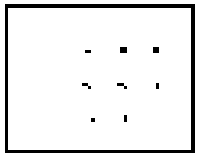
©Yao Wang, 2003

Hierarchical Block Matching Algorithm (HBMA)



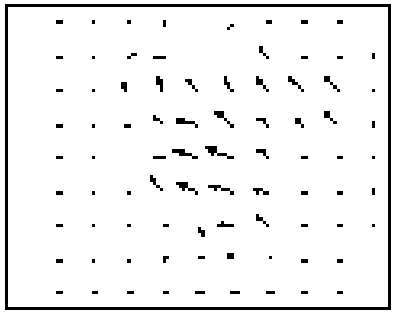


©Yao Wang, 2003 Anchor frame Target frame



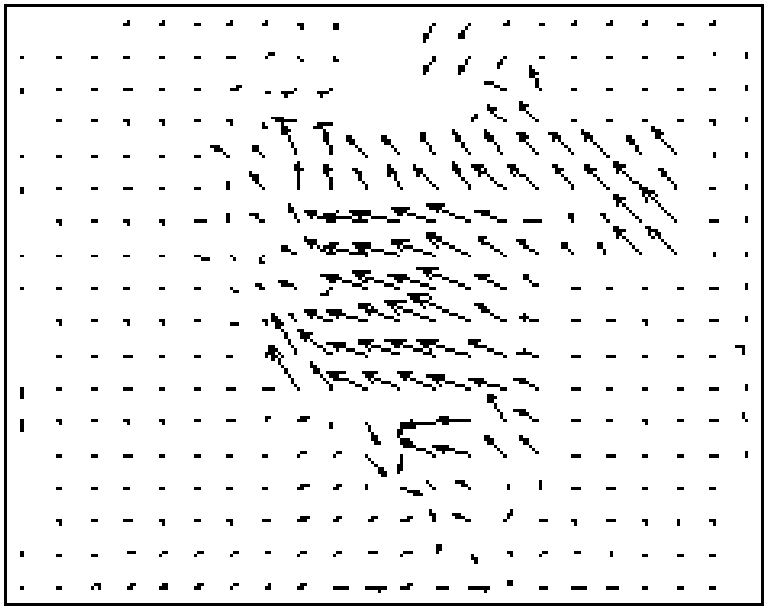
(a)

(b)



(c)

(d)



(e)

(f)

Predicted anchor frame (29.32dB)

©Yao Wang, 2003

Example: Three-level HBMA

target frame

anchor frame

Motion field

Predicted anchor frame (29.86dB)

©Yao Wang, 2003

Example: Half-pel EBMA

Computation Requirement of HBMA

- Definitions
  - Image size: MxM; Block size: NxN at every level; Levels: L
  - Search range:
    - 1st level: R/2^(L-1) (Equivalent to R in L-th level)
    - Other levels: R/2^(L-1) (could be smaller – since motion error is likely to be small)
- Operation counts for EBMA
  - image size M, block size N, search range R
  - # operations: $M^2(2R+1)^2$
- Operation counts at L-th level (Image size: M/2^(L-l))

$$\left(M/2^{L-l}\right)^2\left(2R/2^{L-1}+1\right)^2$$

- Total operation count

$$\sum_{l=1}^{L}\left(M/2^{L-l}\right)^2\left(2R/2^{L-1}+1\right)^2 \approx \frac{1}{3}4^{-(L-2)}4M^2R^2$$

- Saving factor: $3\cdot 4^{(L-2)} = 3(L=2);\ 12(L=3)$

Optimization Strategies to find Min. or Max.

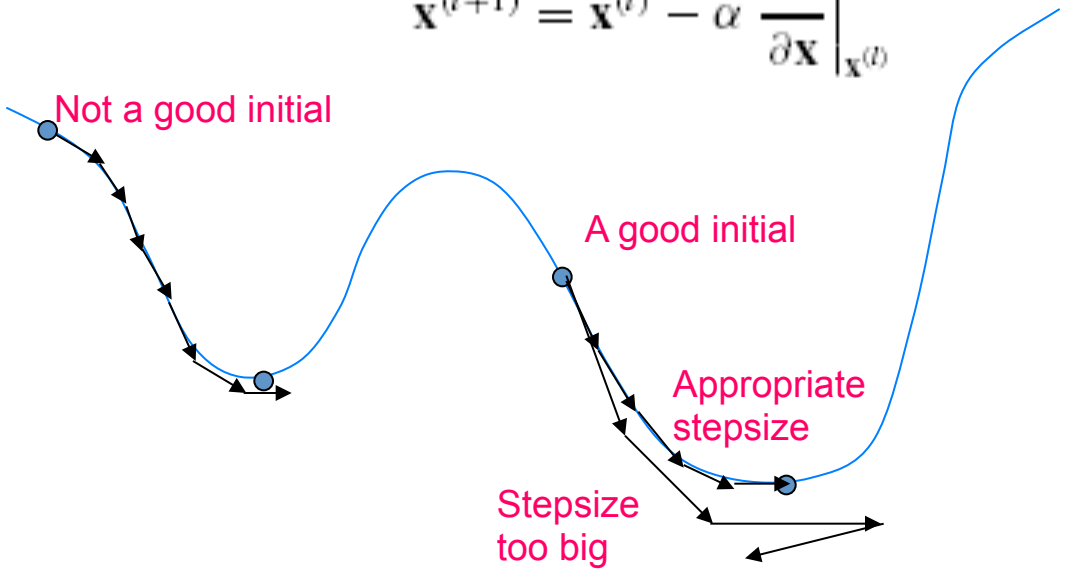
- Exhaustive search
  - Typically used for the DFD criterion with p=1 (MAD)
  - Guarantees reaching the global optimal
  - Computation required may be unacceptable when there are many parameters to search simultaneously!
  - Fast search algorithms reach sub-optimal solution in shorter time
- Gradient-based search
  - Typically used for the DFD or OF criterion with p=2 (MSE)
    - the gradient can often be calculated analytically
    - When used with the OF criterion, closed-form solution may be obtained
  - Reaches the local optimal point closest to the initial solution
- Multi-resolution search
  - Search from coarse to fine resolution, faster than exhaustive search
  - Less likely to be trapped into a local minimum

©Yao Wang, 2003

Gradient Descent Method

- Iteratively update the current estimate in the direction opposite the gradient direction.

$$\mathbf{x}^{(l+1)} = \mathbf{x}^{(l)} - \alpha \left. \frac{\partial J}{\partial \mathbf{x}} \right|_{\mathbf{x}^{(l)}}$$



- The solution depends on the initial condition. Reaches the local minimum closest to the initial condition
- Choice of step side:
  - Fixed stepsize: Stepsize must be small to avoid oscillation, requires many iterations
  - Steepest gradient descent (adjust stepsize optimally)

©Yao Wang, 2003

Newton's Method

- Newton's method

$$\mathbf{x}^{(l+1)} = \mathbf{x}^{(l)} - \alpha\left[\mathbf{H}\left(\mathbf{x}^{(l)}\right)\right]^{-1} \left.\frac{\partial J}{\partial \mathbf{x}}\right|_{\mathbf{x}^{(l)}}$$

$$[\mathbf{H}(\mathbf{x})] = \frac{\partial^2 J}{\partial \mathbf{x}^2} = \begin{bmatrix} \frac{\partial^2 J}{\partial x_1 \partial x_1} & \frac{\partial^2 J}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 J}{\partial x_1 \partial x_K} \\ \frac{\partial^2 J}{\partial x_2 \partial x_1} & \frac{\partial^2 J}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 J}{\partial x_2 \partial x_K} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 J}{\partial x_K \partial x_1} & \frac{\partial^2 J}{\partial x_K \partial x_2} & \cdots & \frac{\partial^2 J}{\partial x_K \partial x_K} \end{bmatrix}$$

  - Converges faster than 1st order method (I.e. requires fewer number of iterations to reach convergence)
  - Requires more calculation in each iteration
  - More prone to noise (gradient calculation is subject to noise, more so with 2nd order than with 1st order)
  - May not converge if \alpha >=1. Should choose \alpha appropriate to reach a good compromise between guaranteeing convergence and the convergence rate.

Reminder: high-level framework

- Motion representation
- Optimization criteria
- Optimization strategies

- Mix and match
  - Pixel-based representation, DFD, gradient descent
  - Pixel-based representation, OF, least-squares

  - Block-based representation, DFD, exhaustive search
  - Block-based representation, DFD, hierarchical search

  - Mesh representation, DFD, iterative search
  - Global motion

Pixel-Based Motion Estimation

- Multipoint neighborhood method
  - Assuming every pixel in a small block surrounding a pixel has the same MV
- Horn-Schunck (1981) method
  - OF + smoothness criterion
- Pel-recursive method
  - MV for a current pel is updated from those of its previous pels, so that the MV does not need to be coded
  - Developed for early generation of video coder

©Yao Wang, 2003

Multipoint Neighborhood Method

- Estimate the MV at each pixel independently, by minimizing the optimization criterion over a neighborhood surrounding this pixel
- Every pixel in the neighborhood B(x) is assumed to have the same MV
- Case 1: Gradient descent with DFD criterion

$$E_{\mathrm{DFD}}(\mathbf{d}_{\mathrm{n}}) = \sum_{\mathbf{x}\in B(\mathbf{x}_n)} w(\mathbf{x})\left|\psi_2(\mathbf{x}+\mathbf{d}_n) - \psi_1(\mathbf{x})\right|^2 \rightarrow \min$$

- Case 2: Least-squares with OF criterion

$$E_{\mathrm{OF}}(\mathbf{d}_{\mathrm{n}}) = \sum_{\mathbf{x}\in B(\mathbf{x}_n)} w(\mathbf{x})\left|\left(\nabla\psi_1(\mathbf{x})\right)^T \mathbf{d}_n + \psi_2(\mathbf{x}) - \psi_1(\mathbf{x})\right|^2 \rightarrow \min$$

©Yao Wang, 2003

Example: Gradient Descent Method

$$E_{\mathrm{DFD}}(\mathbf{d}_{\mathrm{n}}) = \sum_{\mathbf{x}\in B(\mathbf{x}_n)} w(\mathbf{x})\left|\psi_2(\mathbf{x}+\mathbf{d}_n) - \psi_1(\mathbf{x})\right|^2 \rightarrow \min$$

$$\mathbf{g}(\mathbf{d}_{\mathrm{n}}) = \frac{\partial E}{\partial \mathbf{d}_{\mathrm{n}}} = \sum_{\mathbf{x}\in B(\mathbf{x}_n)} w(\mathbf{x}) e(\mathbf{x}+\mathbf{d}_n) \frac{\partial \psi_2}{\partial \mathbf{x}}\bigg|_{\mathbf{x}+\mathbf{d}_n}$$

First order gradient descent:

$$\mathbf{d}_{\mathrm{n}}{}^{(l+1)} = \mathbf{d}_{\mathrm{n}}{}^{(l)} - \alpha\, \mathbf{g}(\mathbf{d}_{\mathrm{n}}{}^{(l)})$$

Example: Gradient Descent Method

$$E_{\mathrm{DFD}}(\mathbf{d}_{\mathrm{n}}) = \sum_{\mathbf{x}\in B(\mathbf{x}_n)} w(\mathbf{x})\left|\psi_2(\mathbf{x}+\mathbf{d}_n) - \psi_1(\mathbf{x})\right|^2 \rightarrow \min$$

$$\mathbf{g}(\mathbf{d}_{\mathrm{n}}) = \frac{\partial E}{\partial \mathbf{d}_{\mathrm{n}}} = \sum_{\mathbf{x}\in B(\mathbf{x}_n)} w(\mathbf{x}) e(\mathbf{x}+\mathbf{d}_n) \frac{\partial \psi_2}{\partial \mathbf{x}}\bigg|_{\mathbf{x}+\mathbf{d}_n}$$

$$\left[\mathbf{H}(\mathbf{d}_{\mathrm{n}})\right] = \frac{\partial^2 E}{\partial \mathbf{d}_{\mathrm{n}}{}^2} = \sum_{\mathbf{x}\in B(\mathbf{x}_n)} w(\mathbf{x}) \frac{\partial \psi_2}{\partial \mathbf{x}} \left(\frac{\partial \psi_2}{\partial \mathbf{x}}\right)^T \bigg|_{\mathbf{x}+\mathbf{d}_n} + w(\mathbf{x}) e(\mathbf{x}+\mathbf{d}_n) \frac{\partial^2 \psi_2}{\partial \mathbf{x}^2}\bigg|_{\mathbf{x}+\mathbf{d}_n}$$

$$\approx \sum_{\mathbf{x}\in B(\mathbf{x}_n)} w(\mathbf{x}) \frac{\partial \psi_2}{\partial \mathbf{x}} \left(\frac{\partial \psi_2}{\partial \mathbf{x}}\right)^T \bigg|_{\mathbf{x}+\mathbf{d}_n}$$

First order gradient descent:

$$\mathbf{d}_{\mathrm{n}}{}^{(l+1)} = \mathbf{d}_{\mathrm{n}}{}^{(l)} - \alpha\, \mathbf{g}(\mathbf{d}_{\mathrm{n}}{}^{(l)})$$

Newton - Raphson method:

$$\mathbf{d}_{\mathrm{n}}{}^{(l+1)} = \mathbf{d}_{\mathrm{n}}{}^{(l)} - \alpha \left[\mathbf{H}(\mathbf{d}_{\mathrm{n}}{}^{(l)})\right]^{-1} \mathbf{g}(\mathbf{d}_{\mathrm{n}}{}^{(l)})$$

©Yao Wang, 2003

Least-square solution: OF Criterion

$$E_{\mathrm{OF}}(\mathbf{d}_{\mathrm{n}}) = \sum_{\mathbf{x}\in B(\mathbf{x}_n)} w(\mathbf{x})\left|\left(\nabla\psi_1(\mathbf{x})\right)^T\mathbf{d}_n + \psi_2(\mathbf{x}) - \psi_1(\mathbf{x})\right|^2 \rightarrow \min$$

$$\frac{\partial E}{\partial \mathbf{d}_{\mathrm{n}}} = 2\sum_{\mathbf{x}\in B(\mathbf{x}_n)} w(\mathbf{x})\left(\left(\nabla\psi_1(\mathbf{x})\right)^T\mathbf{d}_n + \psi_2(\mathbf{x}) - \psi_1(\mathbf{x})\right)\nabla\psi_1(\mathbf{x}) = 0$$

Unique minimum. Solve directly.

$$\mathbf{d}_{\mathrm{n,opt}} = \left(\sum_{\mathbf{x}\in B(\mathbf{x}_n)} w(\mathbf{x})\nabla\psi_1(\mathbf{x})\left(\nabla\psi_1(\mathbf{x})\right)^T\right)^{-1}\left(\sum_{\mathbf{x}\in B(\mathbf{x}_n)} w(\mathbf{x})\left(\psi_1(\mathbf{x}) - \psi_2(\mathbf{x})\right)\nabla\psi_1(\mathbf{x})\right)$$

The solution is good only if the actual MV is small. When this is not the case, one should iterate the above solution, with the following update:

$$\psi_2{}^{(l+1)}(\mathbf{x}) = \psi_2(\mathbf{x} + \mathbf{d}_{\mathrm{n}}{}^{(l)})$$

$$\mathbf{d}_{\mathrm{n}}{}^{(l+1)} = \mathbf{d}_{\mathrm{n}}{}^{(l)} + \Delta_n{}^{(l+1)}$$

where $\Delta_n{}^{(l+1)}$ denote the MV found at that iteration

Intuitively, this takes the target image and shifts it by the best known vector. This makes the small-motion approximation more valid

©Yao Wang, 2003

Horn and Schunck (1981)

- Pixel-based motion
- Combine flow equation with smooth-motion constraint

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

- All gradients approximated with local differences
- Iterate; eventually information from regions with strong gradient infiltrate both
  - Into regions with nearly zero gradient
  - Across object boundaries

Optical flow criterion and gradient descent

- OF criterion is good only if motion is small.
- When the motion is not small, can iterate the solution based on the OF criterion to satisfy the DFD criterion.
- OF criterion can often yield closed-form solution as the objective function is quadratic in MVs.

©Yao Wang, 2003

High-level Framework

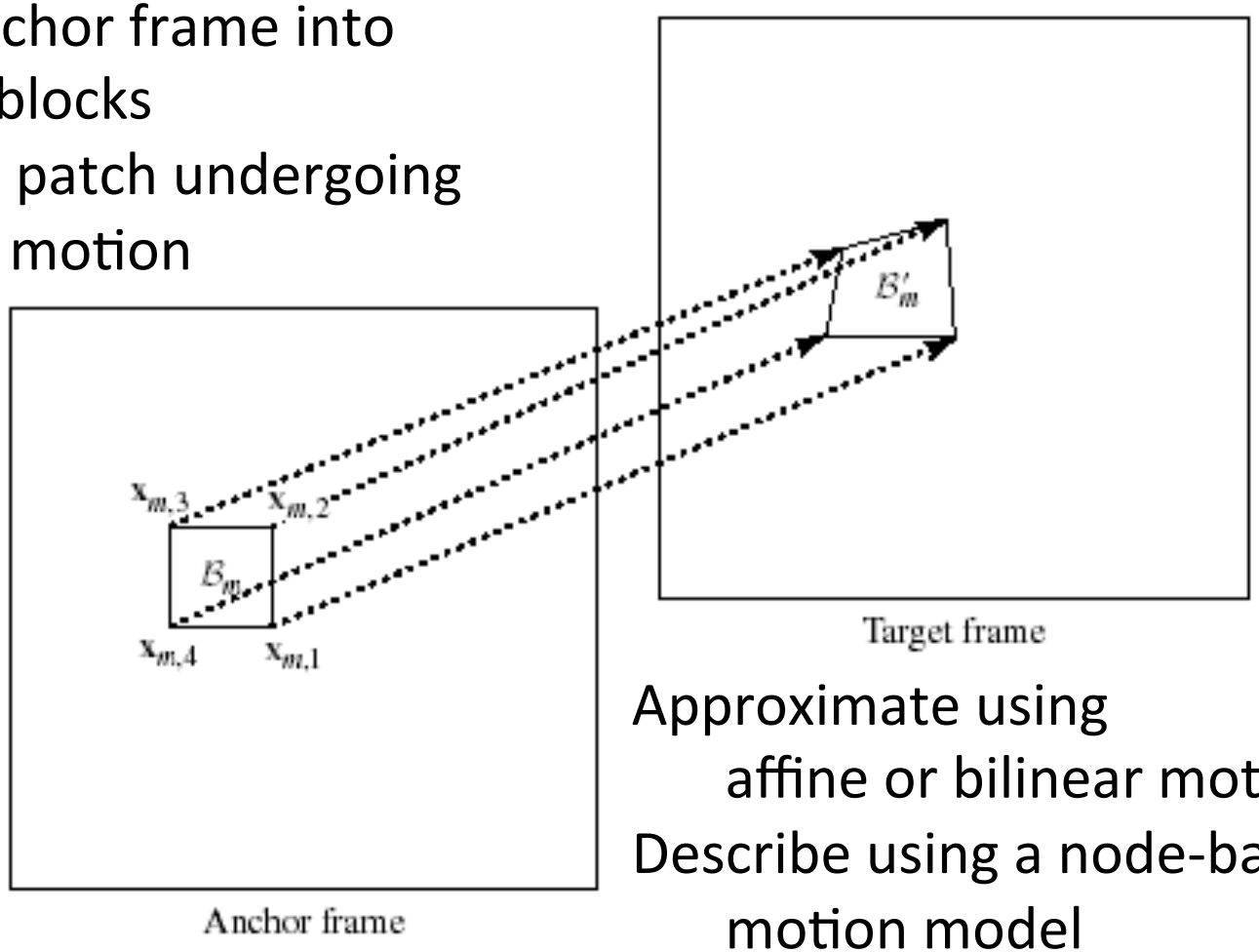
- Motion representation
- Optimization criteria
- Optimization strategies

- Mix and match
  - Pixel-based representation, DFD, gradient descent
  - Pixel-based representation, OF, least-square
  - Block-based representation, DFD, exhaustive search
  - Block-based representation, DFD, hierarchical search
  - Deformable block and mesh representations
  - Global motion

Deformable Block Matching Algorithm

Partition anchor frame into regular blocks

Assume flat patch undergoing rigid 3D motion



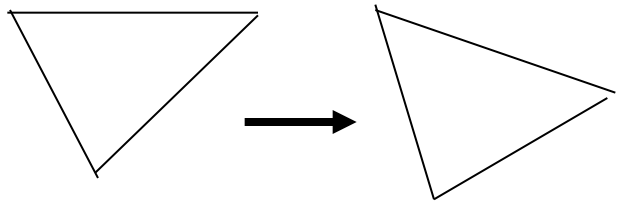
Approximate using affine or bilinear motion

Describe using a node-based motion model

Affine and Bilinear Model

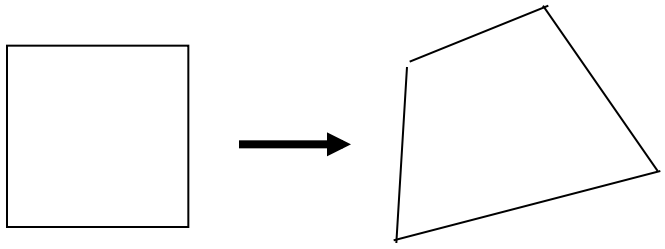
- Affine (6 parameters):
  - Good for mapping triangles to triangles

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



- Bilinear (8 parameters):
  - Good for mapping blocks to quadrangles

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



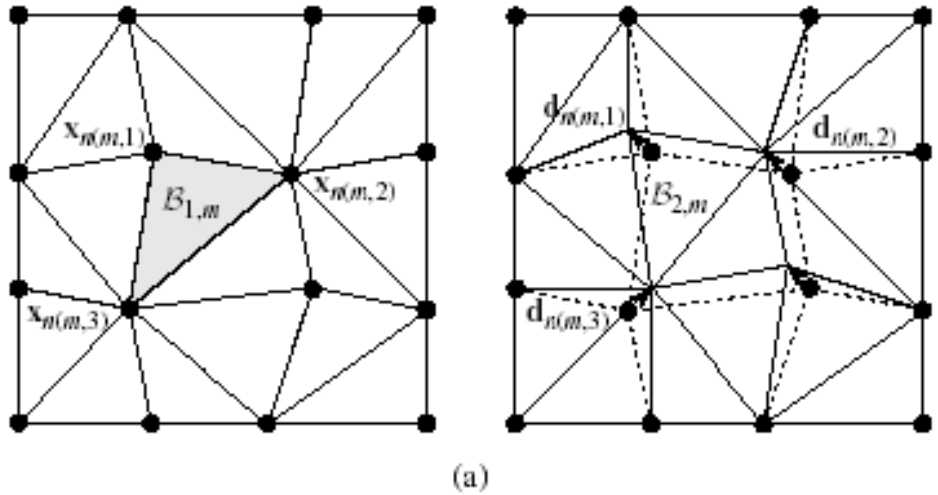
Representing Deformable Blocks

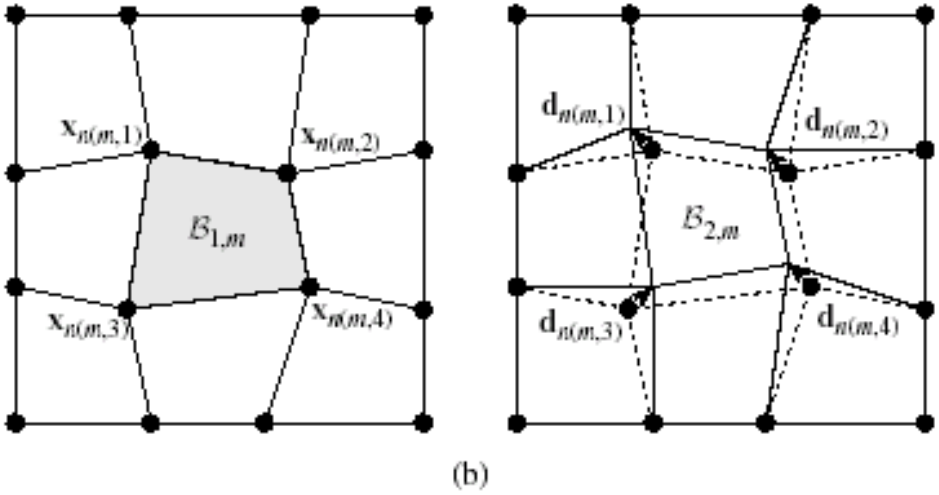
1. Represent with polynomial coefficients of the model
2. Represent with nodal motion
   - Motion vector for each node
   - Interpolation kernel for pixels inside node (kernel depends on the motion model)

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

- Advantages of second representation
  - More efficient communication (polynomial coefficients need high precision)
  - Easier to define both search range and search stepsize
  - All parameters are equally important
  - All parameters need same degree of precision

Mesh-Based Motion Estimation





Partition frame into non-overlapping polygons

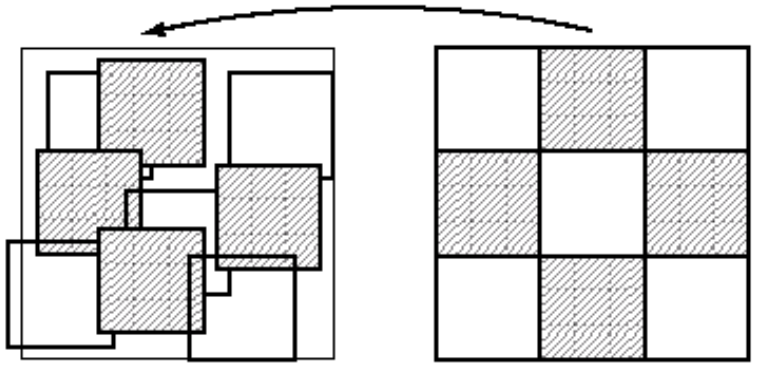
Describe using nodal motion

- Motion of the nodes
- Interpolation kernel

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

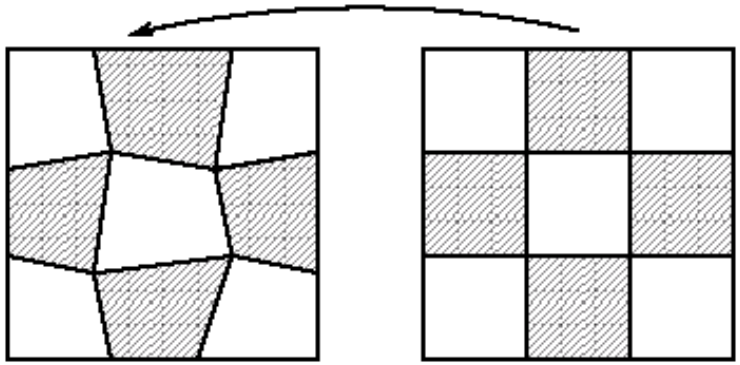
When they move,
nodes must stay in a mesh

Mesh-based vs. block-based motion estimation



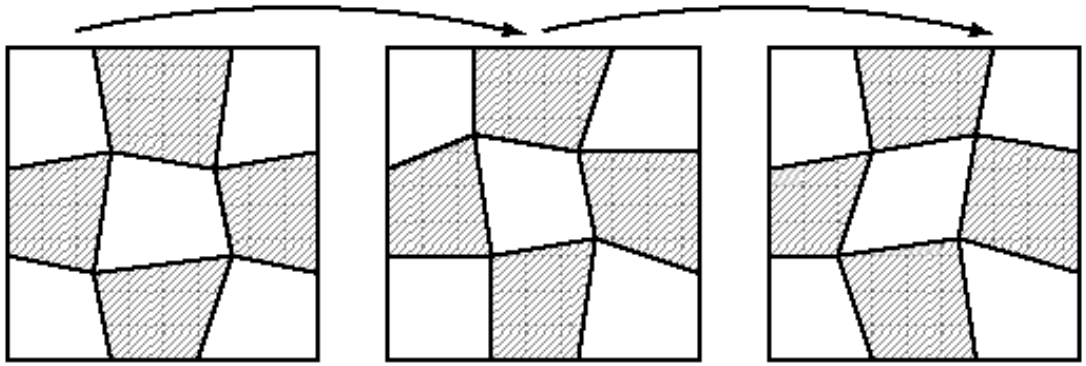
(a)

(a) block-based backward ME



(b)

(b) mesh-based backward ME



(c)

(c) mesh-based forward ME

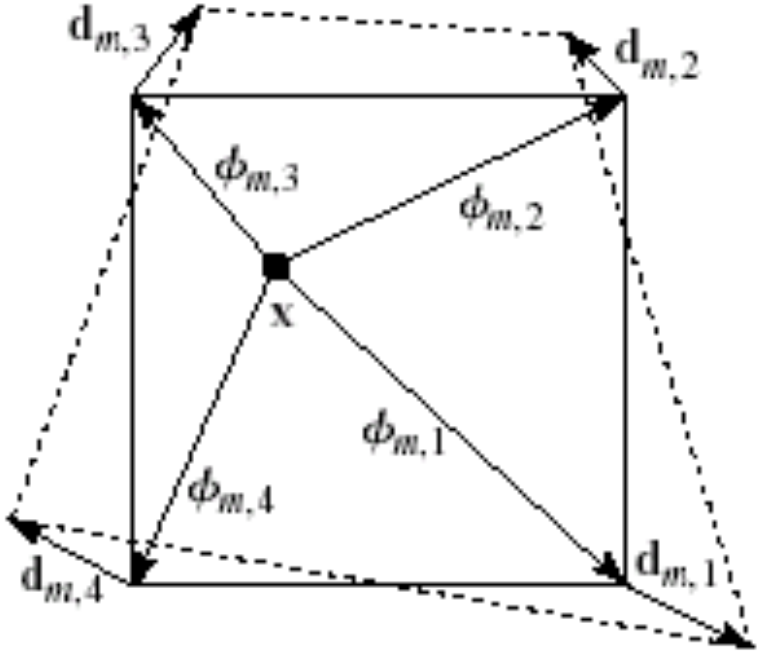
Node-Based Motion Model

Control nodes in this example: Block corners

Motion in other points are interpolated from the nodal MVs $\mathbf{d}_{m,k}$

Control node MVs can be described with integer or half-pel accuracy, all have same importance

Translation, affine, and bilinear are special case of this model



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

©Yao Wang, 2003

Interpolation Kernels

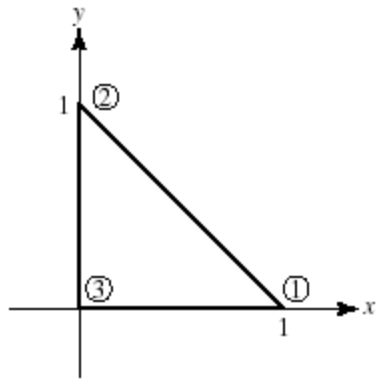
- Requirement:

$$0 \le \phi_{m,k}(\mathbf{x}) \le 1, \quad \sum_k \phi_{m,k}(\mathbf{x}) = 1, \quad \forall\, \mathbf{x} \in \mathcal{B}_{1,m},$$

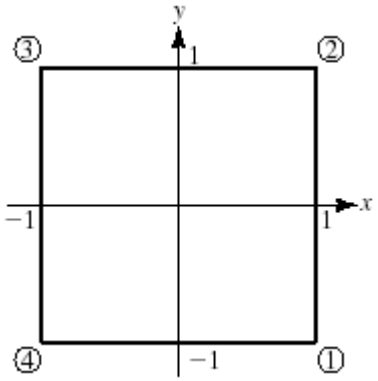
$$\phi_{m,k}(\mathbf{x}_l) = \delta_{k,l} = \begin{cases} 1 & k = l, \\ 0 & k \neq l. \end{cases}$$

- For standard triangular element:

$$\phi_1^t(x, y) = x, \quad \phi_2^t(x, y) = y, \quad \phi_3^t(x, y) = 1 - x - y.$$



- For standard quadrilateral element:



$$\phi_1^q(x, y) = (1 + x)(1 - y)/4, \quad \phi_2^q(x, y) = (1 + x)(1 + y)/4,$$
$$\phi_3^q(x, y) = (1 - x)(1 + y)/4, \quad \phi_4^q(x, y) = (1 - x)(1 - y)/4.$$

©Yao Wang, 2003

Some details about DBMA

- DBMA: each node has 4 possible MV (each block has 4 MV)
- A practical algorithm:
  - First, apply EBMA to all blocks
    - Blocks with small EBMA errors have *translational motion*
    - Blocks with large EBMA errors may have *non-translational motion*
  - Next, apply DBMA to blocks with large EBMA
    - Blocks still having large errors are *non-motion compensatible*
  - [Ref] O. Lee and Y. Wang, Motion compensated prediction using nodal based deformable block matching. J. Visual Communications and Image Representation (March 1995), 6:26-34

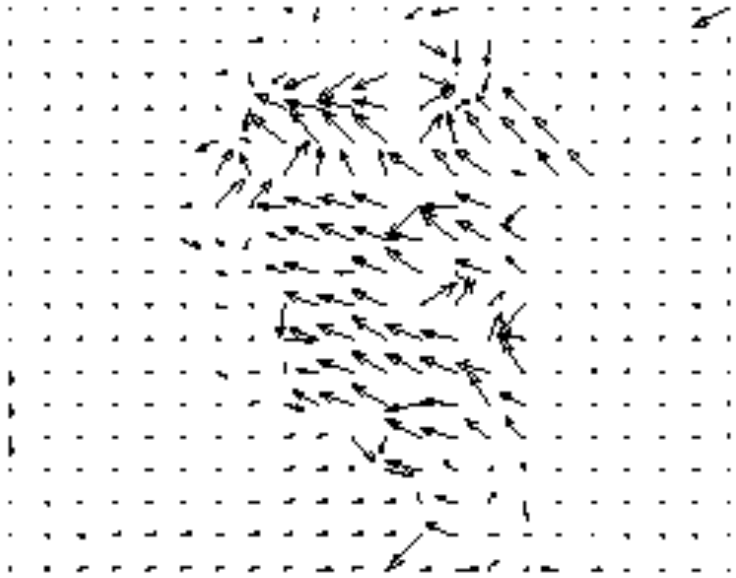
target frame

anchor frame

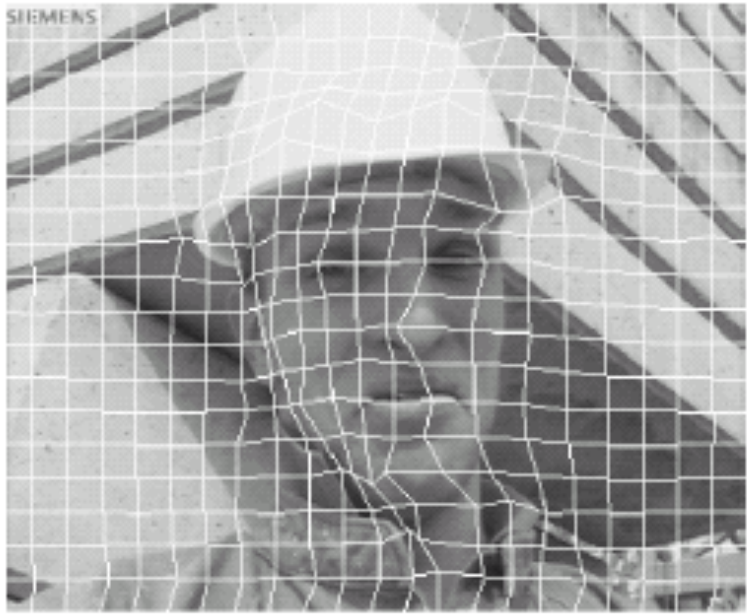
Motion field

Predicted anchor frame (29.86dB)

Example: Half-pel EBMA



EBMA (29.86dB)



mesh-based method (29.72dB)

EBMA vs. Mesh-based Motion Estimation

Global Motion Estimation

- Global motion:
  - Camera moving over a stationary scene
    - Most projected camera motions can be captured by affine mapping
  - Assumes the scene moves in its entirety – very rare
  - Can decompose scene into several major regions, each moving differently (region-based motion estimation)
- Determine global motion parameters for all pixels:
  - Direct estimation
  - Indirect estimation
- Exempt some pixels from the global motion estimation:
  - Iteratively determine the motion parameters and the set of pixels
  - Robust estimator

Direct Estimation

- Parameterize the DFD error in terms of the motion parameters, and estimate these parameters by minimizing the DFD error

$$E_{\mathrm{DFD}} = \sum_{n \in \mathcal{N}} w_n |\psi_2(\mathbf{x}_n + \mathbf{d}(\mathbf{x}_n; \mathbf{a})) - \psi_1(\mathbf{x}_n)|^p$$

Weighting w_n coefficients depend on the importance of pixel $\mathbf{x}_n$.

Ex: Affine motion:

$$\begin{bmatrix} d_x(\mathbf{x}_n;\mathbf{a}) \\ d_y(\mathbf{x}_n;\mathbf{a}) \end{bmatrix} = \begin{bmatrix} a_0 + a_1 x_n + a_2 y_n \\ b_0 + b_1 x_n + b_2 y_n \end{bmatrix}, \quad \mathbf{a} = [a_0, a_1, a_2, b_0, b_1, b_2]^T$$

Exhaustive search or gradient descent method can be used to find **a** that minimizes E_{DFD}

Indirect Estimation

- First find the dense motion field using pixel-based or block-based approach (e.g. EBMA)
- Then parameterize the resulting motion field using the motion model through least squares fitting

$$E_{fit} = \sum w_n (\mathbf{d}(\mathbf{x}_n;\mathbf{a}) - \mathbf{d}_n)^2$$

Affine motion :

$$\mathbf{d}(\mathbf{x}_n;\mathbf{a}) = [\mathbf{A}_n]\mathbf{a},$$

$$[\mathbf{A}_n] = \begin{bmatrix} 1 & x_n & y_n & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_n & y_n \end{bmatrix}$$

$$\frac{\partial E_{fit}}{\partial \mathbf{a}} = \sum w_n [\mathbf{A}_n]^T ([\mathbf{A}_n]\mathbf{a} - \mathbf{d}_n) = 0$$

$$\mathbf{a} = \left(\sum w_n [\mathbf{A}_n]^T [\mathbf{A}_n]\right)^{-1} \left(\sum w_n [\mathbf{A}_n]^T \mathbf{d}_n\right)$$

Weighting w_n coefficients depend on the accuracy of estimated motion at $\mathbf{x}_n$.

©Yao Wang, 2003

Robust Estimator

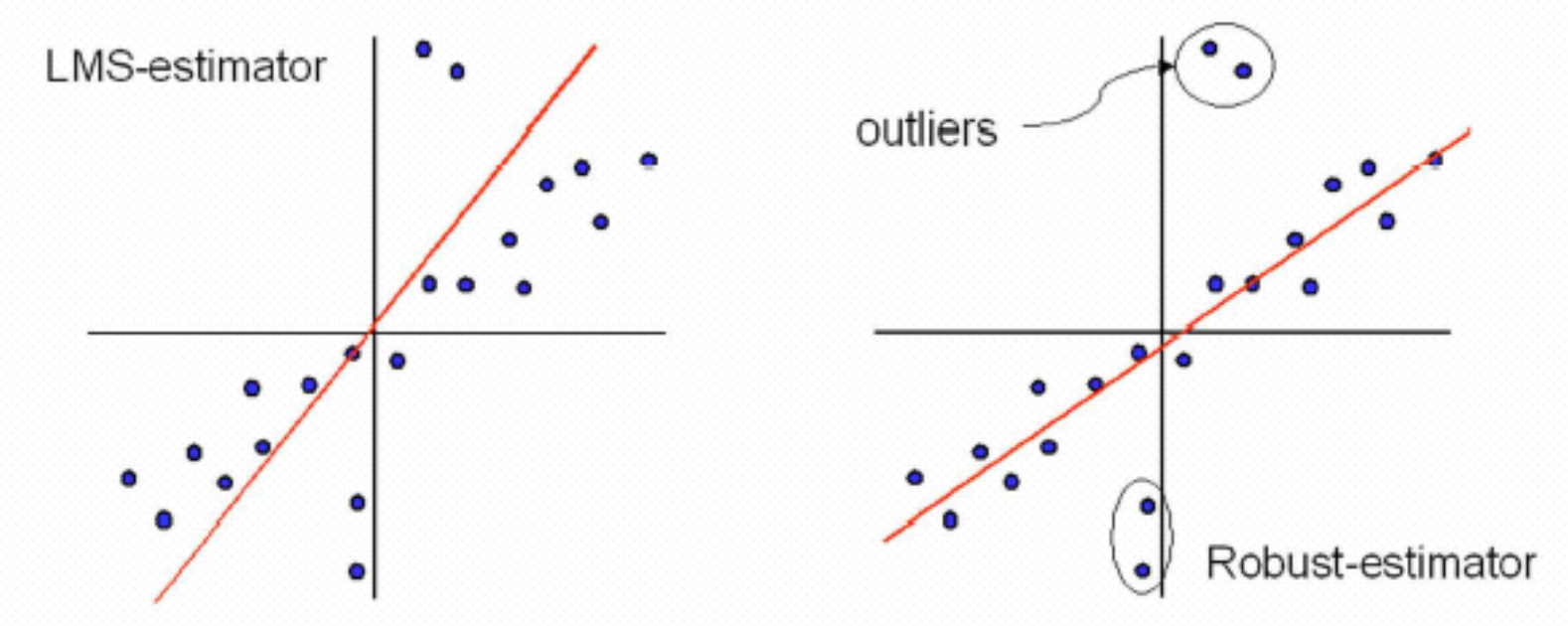
Concept: iteratively removing "outlier" pixels.

1. Initialize the region to include all pixels in a frame
2. Apply the direct or indirect method over all pixels in the region
3. Evaluate errors (E_{DFD} or E_{fit}) at all pixels in the region
4. Eliminate "outlier" pixels with large errors
5. Repeat steps 2-4 for the remaining pixels in the region

More detail in Wang, Ostermann, Zhang

©Yao Wang, 2003

Illustration of Robust Estimator



Fitting a line to the data points by using LMS and robust estimators. Courtesy of Fatih Porikli

Region-Based Motion Estimation

- Assumption: the scene consists of multiple objects, with the region corresponding to each object (or sub-object) having a coherent motion
  - Physically more correct than block-based, mesh-based, global motion model
- Method:
  - Region First: Segment the frame into multiple regions based on texture/ edges, then estimate motion in each region using the global motion estimation method
  - Motion First: Estimate a dense motion field, then segment the motion field so that motion in each region can be accurately modeled by a single set of parameters
  - Joint region-segmentation and motion estimation: iterate the two processes

Summary

- Fundamentals:
  - Optical flow equation
    - Derived from constant intensity and small motion assumption
    - Ambiguity in motion estimation
  - How to represent motion:
    - Pixel-based, block-based, region-based, global, etc.
  - Estimation criterion:
    - DFD (constant intensity)
    - OF (constant intensity+small motion)
  - Search method:
    - Exhaustive search, gradient-descent, multi-resolution

Summary (Cntd)

- Basic techniques:
  - Pixel-based motion estimation
  - Block-based motion estimation
    - EBMA, integer-pel vs. half-pel accuracy, Fast algorithms
- More advanced techniques
  - Multiresolution approach
    - Avoid local minima, smooth motion field, reduced computation
  - Deformable block matching algorithm (DBMA)
    - To allow more complex motion within each block
  - Mesh-based motion estimation
    - To enforce continuity of motion across block boundaries
  - Global motion estimation
    - Good for estimating camera motion
  - Region-based motion estimation
    - More physically correct: allow different motion in each sub-object region

©Yao Wang, 2003