Compact Lecture

Multimedia Coding: Methods & Applications

Part 4: Video Coding Fundamentals

4.1: Motion Estimation and Compensation

Dr. Klaus Illgner

Dr. Uwe Rauschenbach

What is „Video“?

g_{n-3} g_n

g_{n-3}

g_n

Video is sequence of images {g},

where the images have a ordered relationship in time

Key Feature:

Difference between images mainly caused by motion

What can be done for Efficient Coding?

Resolution of standard TV:

720 x 576, 25 Hz, 4:2.0	→	165,9 Mbps	(90 min → 112 GB)
still image compression 10:1	→	16,6 Mbps	(90 min → 11,2 GB)

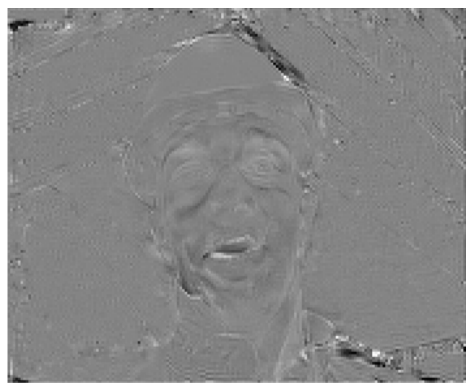
→ amount of data even for SDTV too large for transmission and storage

Approach for coding:

→ transmit only <u>*modified*</u> image areas

→ extend still image coding into temporal domain (kind of “3D”)

No compensation
H = 6.4bit

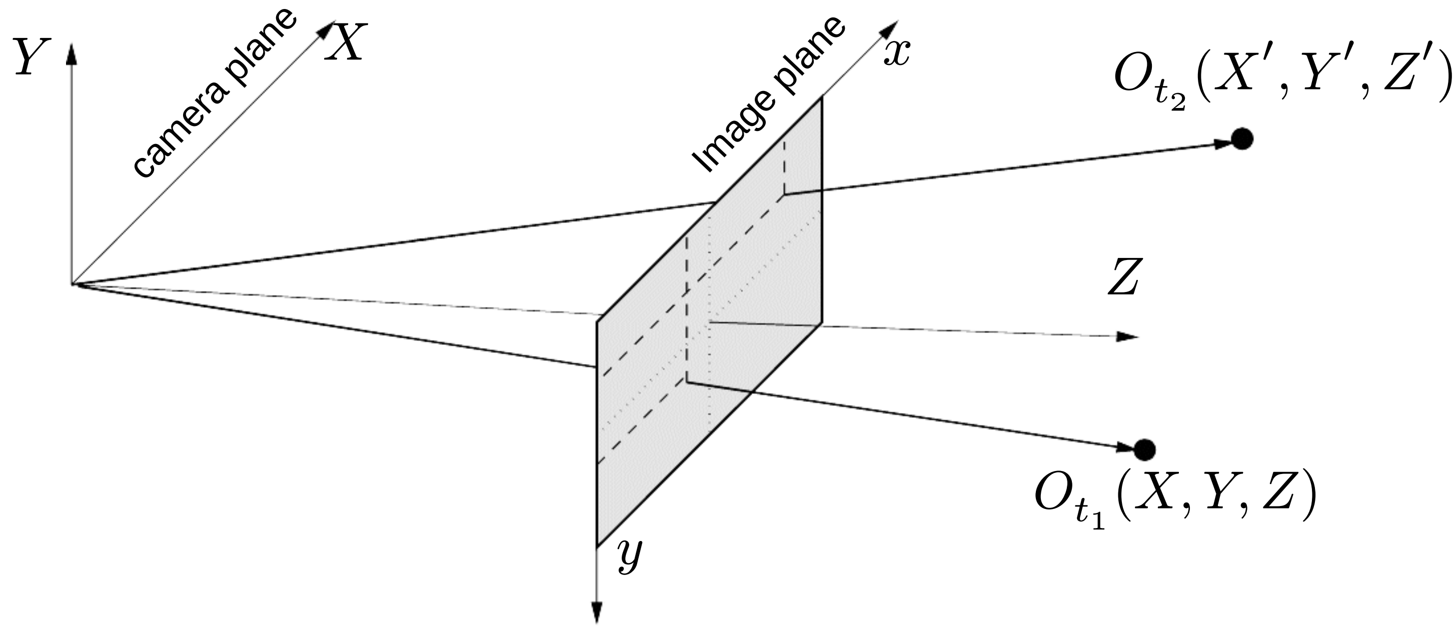


Motion compensated
H = 4.4bit

Approach:
Estimate motion
(reason for changes of the image)

Problem:
How to describe “motion”?

Image Generation



Mapping 3D world → 2D image plane:

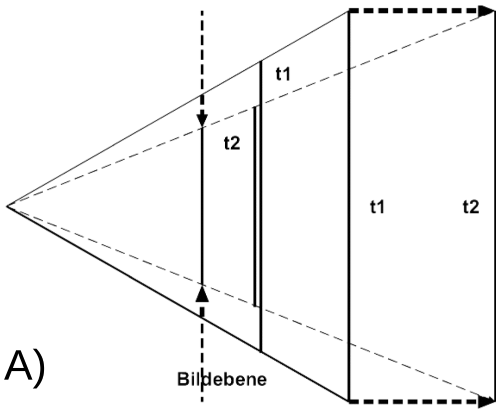
Geometrical optics for modeling

Motion:

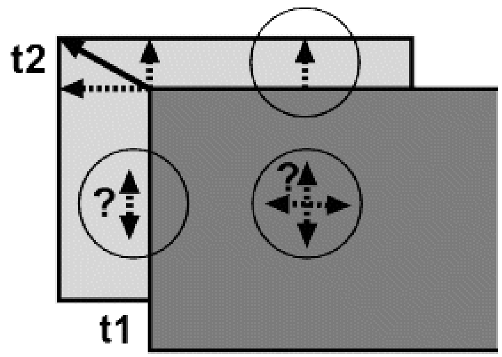
- Projection onto image plane is ***time variable***
- 3D object movement → moving of 2D regions

Mapping of Motion

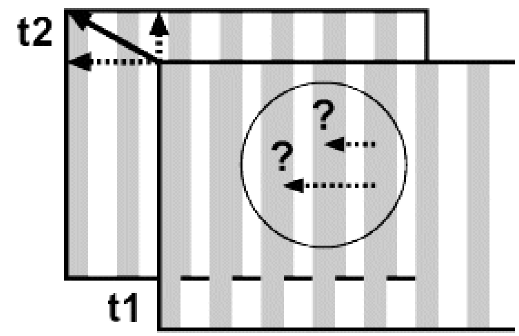
Problem: motion in the image plane is not unique
(no one-to-one mapping between 2D and 2D world)



A)



B)



C)

A) change of size caused by shortening (lengthening), change of depth, rotation

B) aperture problem → locale motion description

C) correspondence problem, in particular for periodic structures → aliasing

→ a unique description requires to assume a certain model

Modeling Motion

- **Consistency of objects:**

  opaque, diffuse reflecting, geometrical form

- **Motion of objects:**

  translation, rotation, deformation

  a) estimating physical parameters → motion analysis

  model: parametric description

  b) finding correspondences → coding

  model: displacement

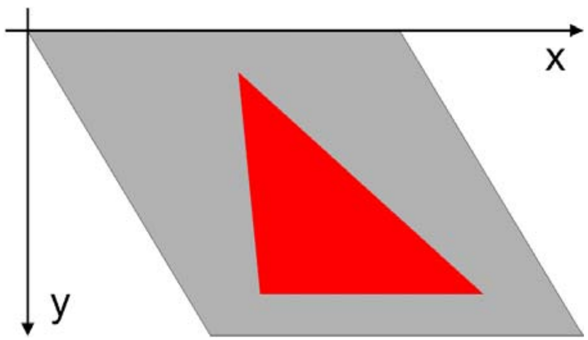
- **Movements of the camera**

  zoom, pan, rotation

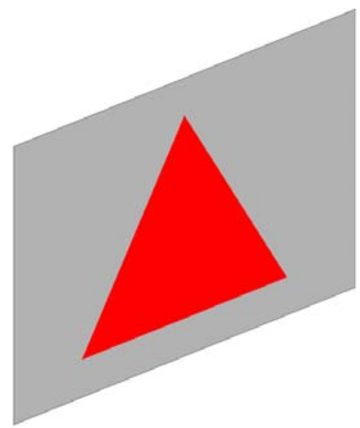
2D Affine Mapping

Transforming coordinates and coordinate systems

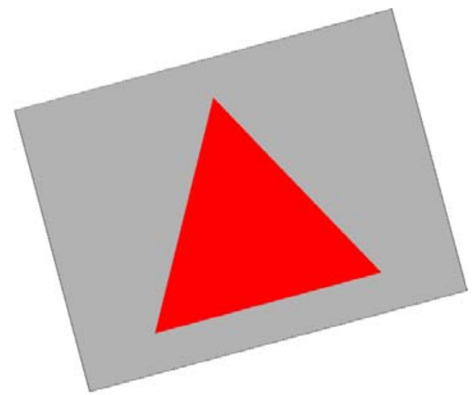
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$$



$$\begin{pmatrix} 1 & 0.5 \\ 0 & 1 \end{pmatrix}$$



$$\begin{pmatrix} 1 & 0 \\ -0.5 & 1 \end{pmatrix}$$



$$\begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}$$

Parametric Motion Model (1)

3D Objektbewegung (3D affin) $\mathbf{X'} = \mathbf{AX} + \mathbf{X_0}$

$\mathbf{A}$ → rotation, deformation

$\mathbf{X_0} \in \mathbb{R}^\mathbf{3}$ → translation

$\mathbf{X}, \mathbf{X'}$ → coordinates in the 3D space

Mapping a point of an object assuming entral projection

$$x' = x\frac{Z}{Z'} + X_0\frac{F}{Z'} \qquad y' = y\frac{Z}{Z'} + Y_0\frac{F}{Z'}$$

Resulting 2D motion of 3D moving of a rigid plane in space (3D):

$$x' = \frac{a_1 x + a_2 y + a_3}{a_7 x + a_8 y + 1} \qquad y' = \frac{a_4 x + a_5 y + a_6}{a_7 x + a_8 y + 1}$$

Parametric Motion Model (2)

Modell	Parameter	Object form	Object motion	Projection
2D translation	2	arbitrary	2D translational	parallel
2D affine 3D affine	6 8	planar planar	3D affine 3D affine	parallel central
2D flexible	2N	2D linear in sections	2D flexible in sections	arbitrary

Estimating the parameters:

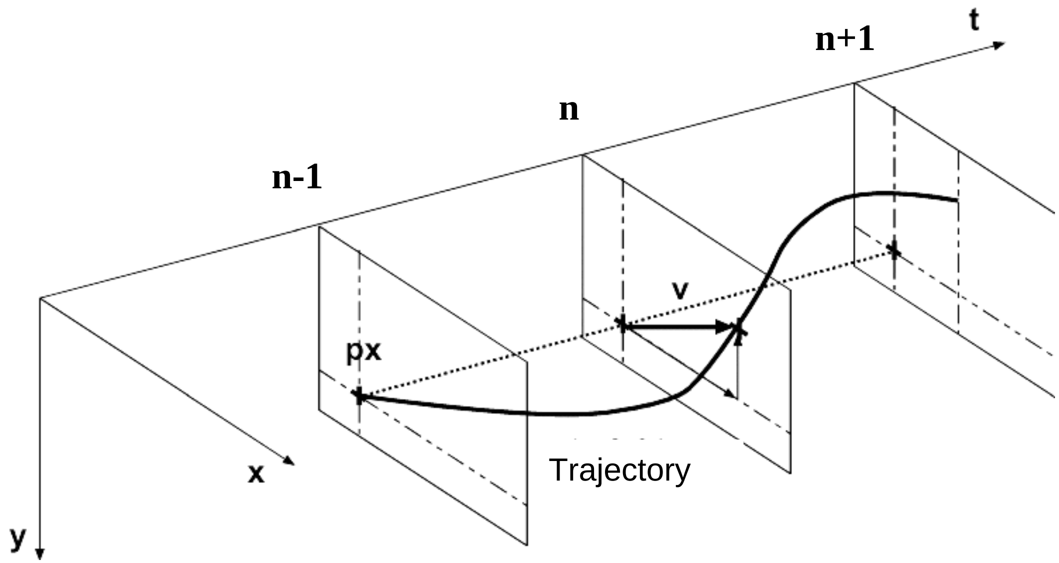
- directly via feature points

  → requires to identify N measurement points (minimum 1 point / parameter)
- indirectly via a displacement vector field

Describing Motion as Displacement

Assumption: $g(\mathbf{x}) \mapsto O(\mathbf{X}), \qquad \forall\ \mathbf{x} \in \mathbb{R}^2,\ \mathbf{X} \in \mathbb{R}^3$

Moving of O(X) → trace in the image sequence → motion trajectory



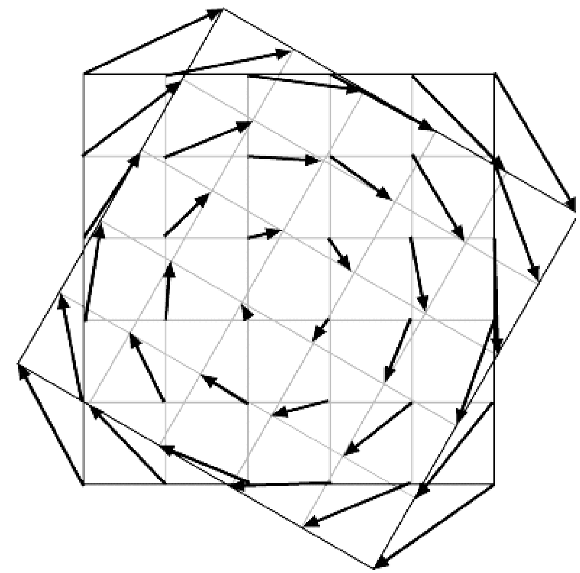
Displacement vector v(x): describes motion in the 2D image plane
Motion estimation: estimating the displacement v(x)

Displacement (Motion) Vector Field (MVF)

Vector field is discrete in space

$$\mathbf{v}_n = \{\mathbf{v}_n[\mathbf{x}] \in^2, \mathbf{x} \in^2\}$$

Characteristics:



- Each vector only describes translational motion
- Dynamic change of vectors
- Describing other types of motion (incl. object deformation) exploiting the <u>change</u> of vectors over time AND the <u>context</u>

Implementation:

- Describing motion discrete in space (location)
  - MVF is samples in space on a sub-grid of the image grid
  - MVF is termed <u>*dense*</u>, if there is a vector for each pixel
- Quantization and thresholding of the vector amplitude

Approaches for Motion Estimation

Motion Estimation (ME) $\mathbf{v}_n = \mathrm{ME}(g_n, g_{n-1})$

Assumptions:

- Unique assignment of motion to 2D plane $g[x] \leftrightarrow O(X)$
- Intensity of pixel remains unaltered over time $\Longrightarrow$ $g_n[\mathbf{x}] = g_{n-1}[\mathbf{x} - \mathbf{v}[\mathbf{x}]]$

Algorithms

- Matching approach

  minimizing an error criteria / maximizing a similarity criterion
- Gradient approach

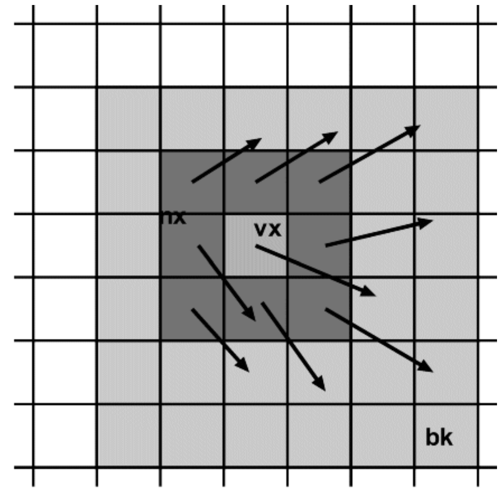
  evaluate the continuity equation → solving an equation system
- Statistical approach

  MVF is the realization of a random process; maximizing the probability

Correspondences in Space

Assuming dense MVF

$$\forall_{\mathbf{y}\in\mathcal{N}_{\mathbf{x}}^2} \mathbf{v}[\mathbf{x}] \approx \mathbf{v}[\mathbf{y}]$$



Motion model

2D motion can approximated as
Locally translational

$$\forall_{\mathbf{y}\in\mathcal{N}_{\mathbf{x}}^2} \mathbf{v}[\mathbf{x}] = \mathbf{v}[\mathbf{y}]$$

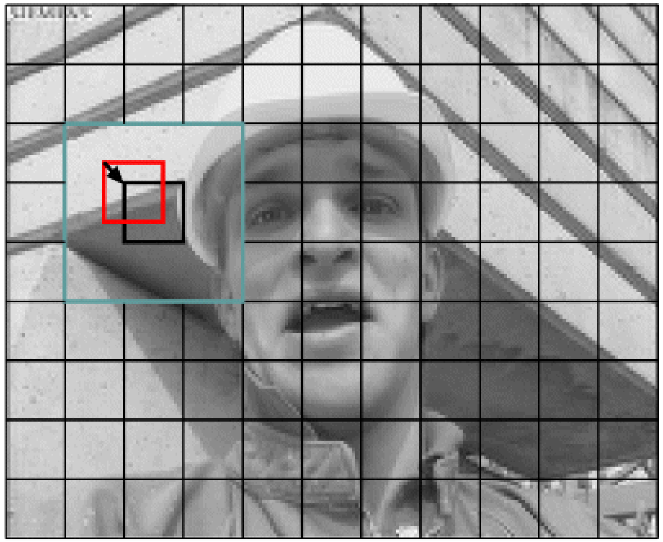
„Matching“ principle

- Partitioning the image into regions, e.g.. blocks
- Minimizing a distance criterion e() for each region

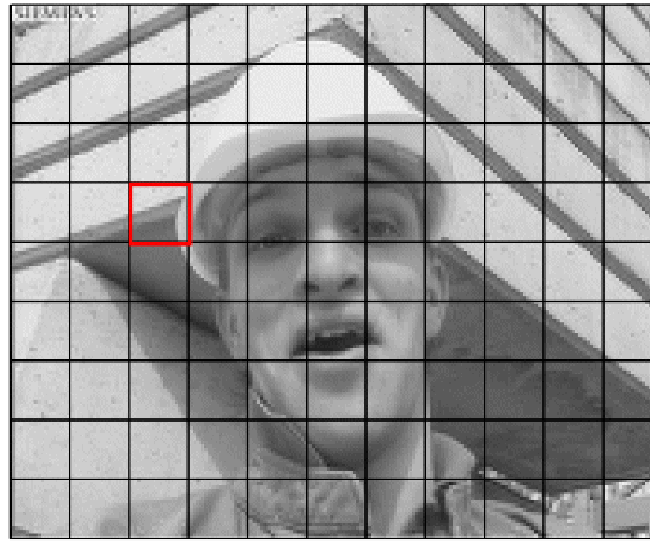
$$e(\mathbf{x}_k, \mathbf{v}) = \sum_{\mathbf{x}\in\mathcal{B}_k} f\Big(g_n[\mathbf{x}], g_{n-1}[\mathbf{x} - \mathbf{v}[\mathbf{x}]]\Big) \rightarrow \min$$

f(): function to weight each pixel error

Block Matching



g_{n-3}



g_n

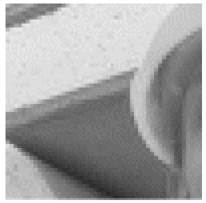
Error criterion:

$$e(\mathbf{x}_k, \mathbf{v}) = \sum_{\mathbf{x} \in \mathcal{B}_k} \Big| g_n[\mathbf{x}] - g_{n-3}[\mathbf{x} - \mathbf{v}] \Big|^{\alpha}, \qquad \alpha \in \{1, 2\}$$

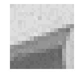
Estimation criterion:

$$\mathbf{v}[\mathbf{x}_k] = \operatorname*{argmin}_{\mathbf{v}_i \in \mathcal{V}} \{ e(\mathbf{x}_k, \mathbf{v}_i) \} \qquad \mathcal{V}\text{: set of test vectors}$$

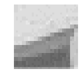
Progression of the Error Criterion (Example)



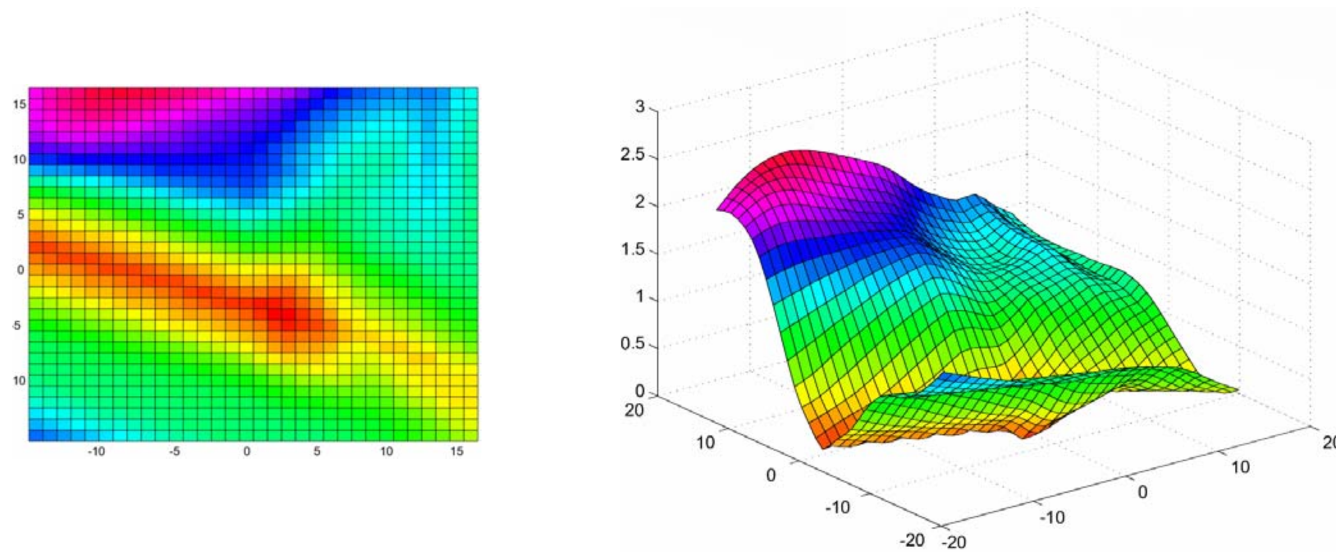
Search area



Reference block
from current image



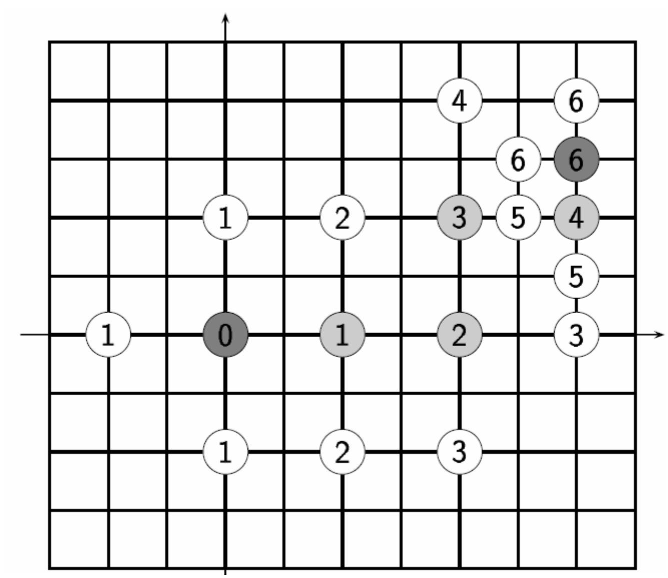
Most similar block
from previous image



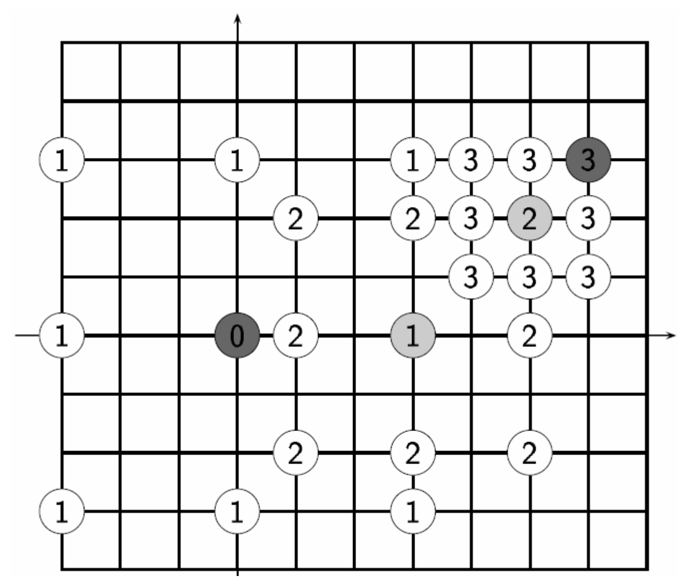
Progression of quadratic error e() over the displacement v

Search Strategies

1. full search: computational expensive, guarantees a minimal error
2. logarithmic search (assumption: convex error progrssion)
3. search in multiple steps (three step with decreasing step size)
4. fast search based on Schwarz Inequality



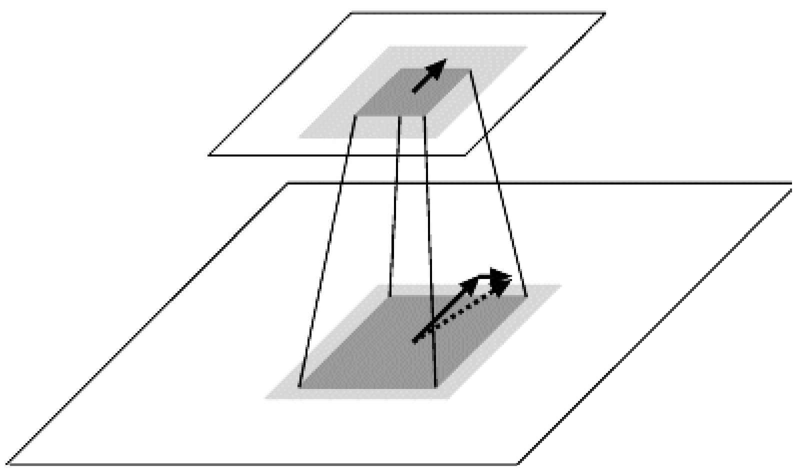
Logarithmic search



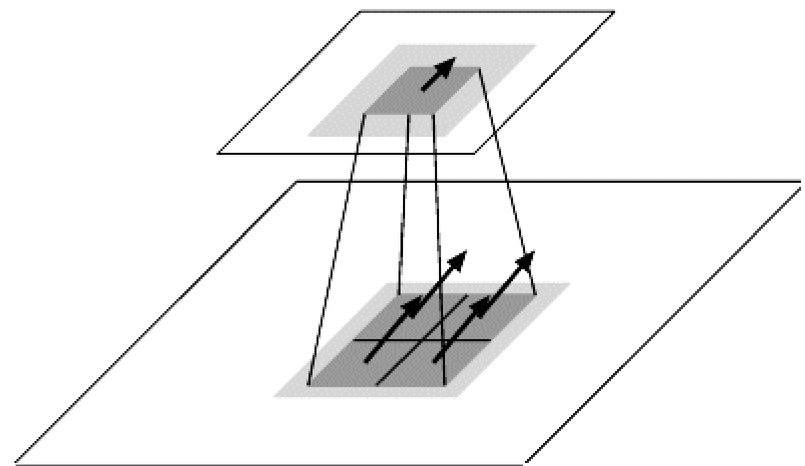
Multiple step search

Hierarchical Search

4. Hierarchical Search
   a) approximation on image of reduced resolution
      using search algorithm out of 1….4
   b) successive refinement on images of higher resolution
      reduced search area → reduced computational complexity



Identical image area per block
→ Enlarged block size



Identical block size
→ Refined description

Extensions (1)

Displacement vector amplitudes are quantized to resolution of image grid:

$$\mathbf{v}[\mathbf{x}] \in \Lambda$$

Increasing the amplitude resoution: $k \cdot \mathbf{v}_n[\mathbf{x}] \in \mathbb{Z}^2$

→ motion estimation on <u>interpolated</u> images

$$\mathbf{v}_n = \mathrm{ME}\Big(\mathrm{E}(g_n), \mathrm{E}(g_{n-1})\Big)$$

Interpolation operator:

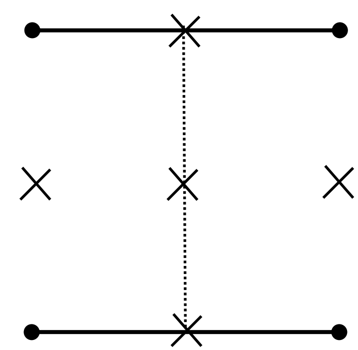
$$\mathrm{E} : \tilde{g} = \mathrm{E}(g) = [g]_{\uparrow k} * h_E$$

Typical interpolation by factor 2 (half-pel) or 4 (quarter-pel)

Example:

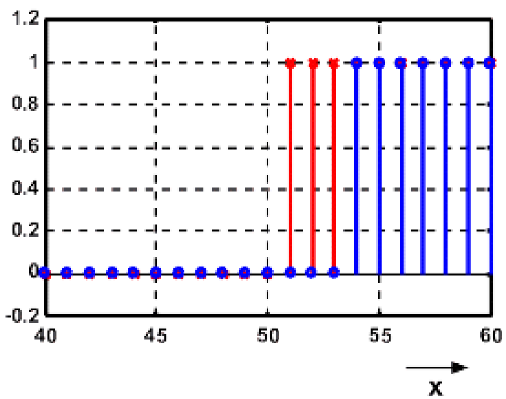
- k = 2
- bilinear interpolation (separable filter kernel)

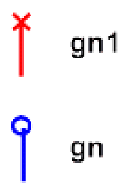
$$h_E[x] = \tfrac{1}{2}\delta[x-1] + \delta[x] + \tfrac{1}{2}\delta[x+1]$$



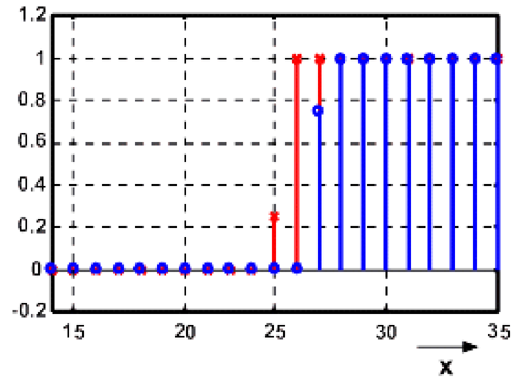
Example for Half-pel Motion Estimation

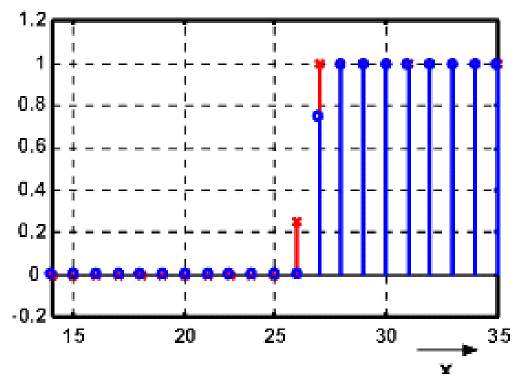
Reference signal





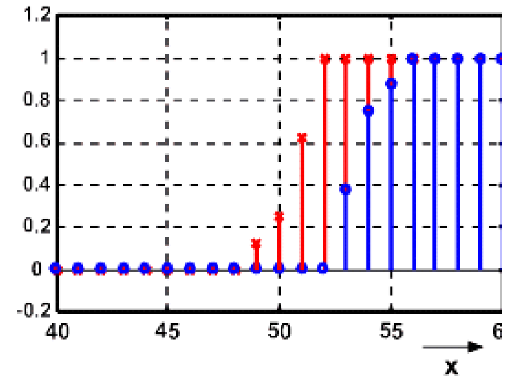
available signals

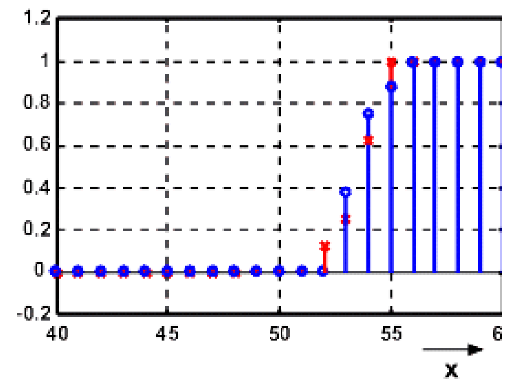




Shifted signal by v=1

interpolated signals





Shifted signal by v=3

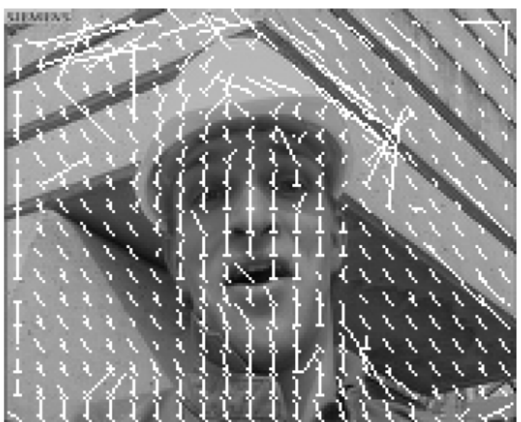
Extensions (2)

Regular vector fields → smoothness constraint:
Assumption: neighbored vectors describe similar motion
(homogeneous motion, rigid bodies)

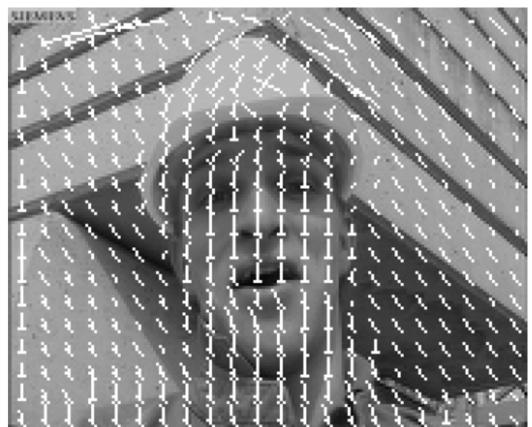
→ Motion estimation taking neighbored vectors into account

$$\mathbf{v}[\mathbf{x}_k] = \underset{\mathbf{v}_i \in \mathcal{V}}{\operatorname{argmin}} \left\{ \Psi(e(\mathbf{x}_k, \mathbf{v}_i)) + \lambda \sum_{\mathbf{y} \in \mathcal{N}_{\mathbf{x_k}}} |\mathbf{v}_i - \mathbf{v}[\mathbf{y}]| \right\}, \quad \lambda \in \mathbb{R}$$

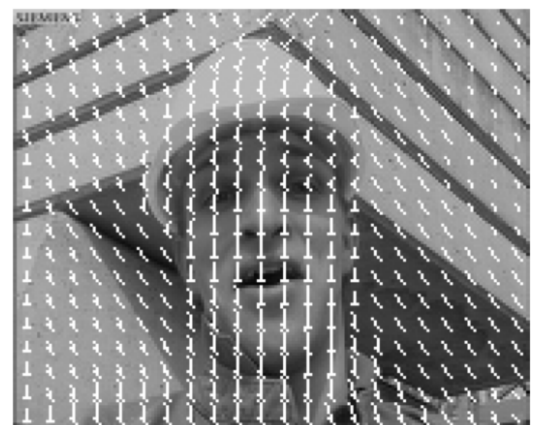
$\Psi(\cdot)$ Weight function



$\lambda = 0$



$\lambda = 0.05$



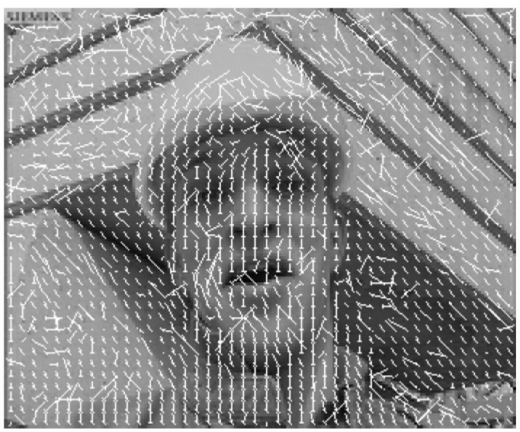
$\lambda = 0.5$

Extensions (3)

Increasing the <u>spatial resolution</u> of MVF:

• using smaller blocks
  estimation is less reliable due to aperture effects and noise
  → hierarchical block matching

• interpolating motion vector fields
  interpolation requires adaptation according to motion model and consideration of motion discontinuities at boudaries



4x4 independent

16x16

4x4 hierarchical

Other Matching Approaches (1)

Correlation as measure for similarities of functions:

$$\begin{aligned}\varphi_{g_n g_{n-1}}(\mathbf{v}) &= (g_n * g'_{n-1})(\mathbf{v}) \\ &= c \cdot \sum_{\mathbf{x}} g_n[\mathbf{x}] \cdot g_{n-1}[\mathbf{x}-\mathbf{v}]\end{aligned}$$

$$g'[x] = g[-x] \qquad \text{normalization: } c^{-1} = \sum_{\mathbf{x}} g_n[\mathbf{x}] \cdot \sum_{\mathbf{x}} g_{n-1}[\mathbf{x}]$$

Criterion for motion estimation
→ Maximizing the cross correlation function

$$\mathbf{v} = \underset{\mathbf{v}_i \in \mathcal{V}}{\operatorname{argmax}} \left\{ \varphi_{g_n g_{n-1}}(\mathbf{v}_i) \right\}$$

- high computational effort
- robust against illumination changes
  → reduced constraints for motion model

Other Matching Approaches (2)

Correspondences in the frequency domain

$$
\begin{aligned}
g[\mathbf{x}] \quad &\circ\!\!-\!\!\bullet\; G(\mathbf{f_x}) \\
g[\mathbf{x}-\mathbf{v}] \quad &\circ\!\!-\!\!\bullet\; G(\mathbf{f_x}) \cdot \exp(-j2\pi\langle \mathbf{v}, \mathbf{f_x}\rangle)
\end{aligned}
$$

Interpretation: motion results in characteristic shifts of the phase

Characteristics:

- high computational complexity
- phase signal typically have significant high frequency components
  - → Estimation by matching unreliable
- displacements present in the entire image can be identified
- motion can not be assigned to local regions

Region Oriented Motion Estimation (1)

Partitioning into regions:

- Homogeneity of features (texture, motion)
- Correspondence of objects (*a priori* knowledge)

→ Estimating of the region <u>form</u> required

- Describing the contour
- Approximating as set of elementary regions such as blocks or triangles

Ill-posed problem: motions versus regions form

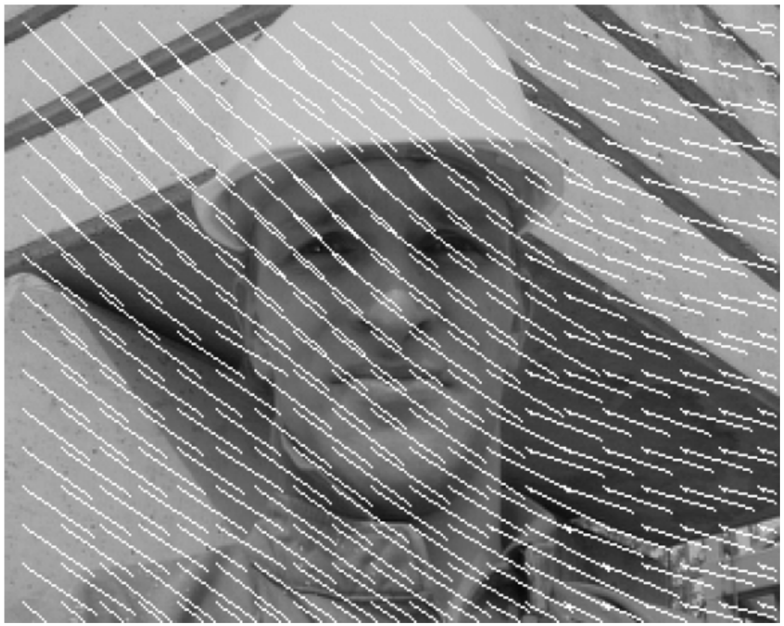
Region Oriented Motion Estimation(2)

Approaches for solving the problem:

- **Alternating estimation**
  - Estimate motion based on existing region partitions (segmentation)
  - Update the segmentation constraint to the feature motion
- **Simplified version**
  - Estimate motion using elementary regions
  - Create segementation by *Split and/or Merge*
- **Criterion for motion estimation includes a region model**
  - Smoothness constraint:
    - motion and texture within regions homogeneous
    - contours of regions are smooth
  - Discontinuities of features (in particular motion) at region boundaries

  → **very high computational complexity**

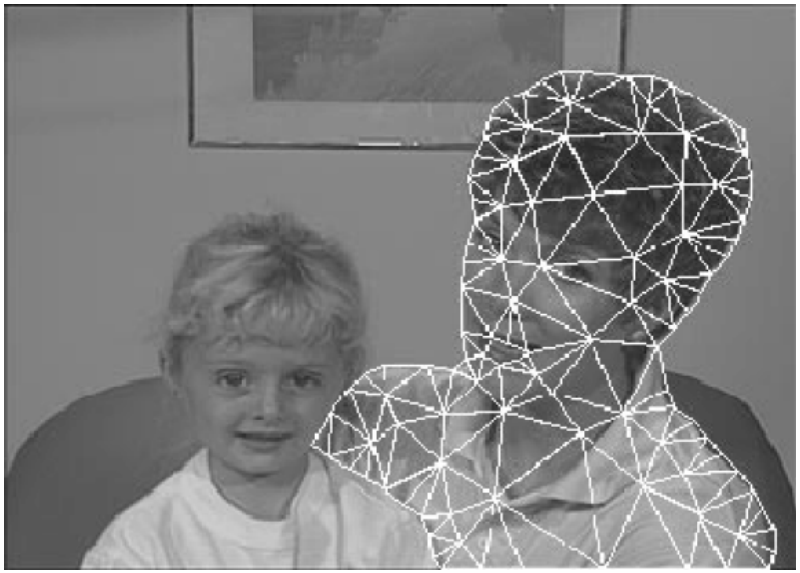
Examples 1:



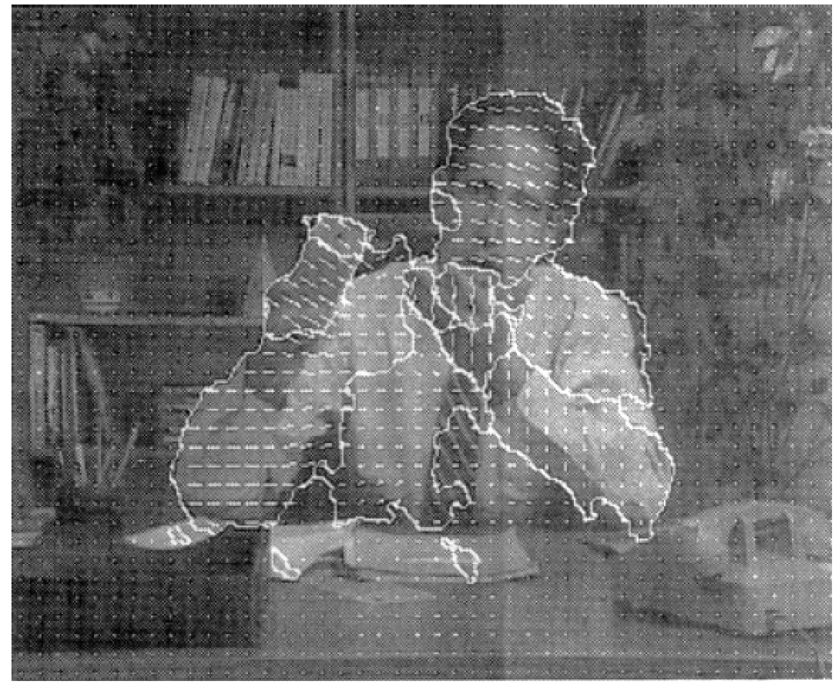
Vector field, calculated by Block matching and regularization

Segmentation derived from the displacement vector field

Example 2:



Estimate „motion“ of
nodes of a triangular mesh



Estimate jointly the
Segmentation and motion
Based on a statistical approach

Motion Compensation

Goal: Coding the difference between images

→ compensation of motion

→ predict an image based on estimated motion

$$\hat{g}_n[\mathbf{x}] = \mathrm{MC}(g_{n-1}, \mathbf{v}_n) = g_{n-1}[\mathbf{x} - \mathbf{v}[\mathbf{x}]], \qquad \mathbf{x} \in \Lambda$$

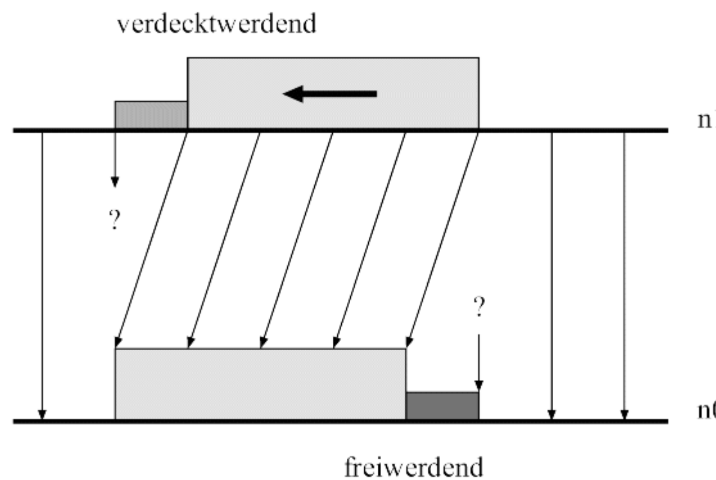
$$\text{mit} \qquad \left| g_n - \hat{g}_n \right|^2 \to \min$$

Error free prediction in reality not possible:

- images are sampled on a grid
- motion model is just an assumption
- VVF sampled and quantised
- border effects
- uncovered background

→ Coding of the **prediction error**

$$d_n = g_n - \hat{g}_n$$



Extensions (1)

Amplitute resolution higher than image grid of image to be compensated

- e.g. block matching with half-pel resolution
- Affine motion parameters

→ subpel compensation required:

$$\hat{g}_n = \mathrm{RD}\Big(\mathrm{MC}\big(\mathrm{EX}(g_{n-1}), \mathbf{v}_n\big)\Big)$$

Reduction operator R:

$$\mathrm{RD} : g = \mathrm{RD}(\tilde{g}) = [g * h_R]_{\downarrow k}$$

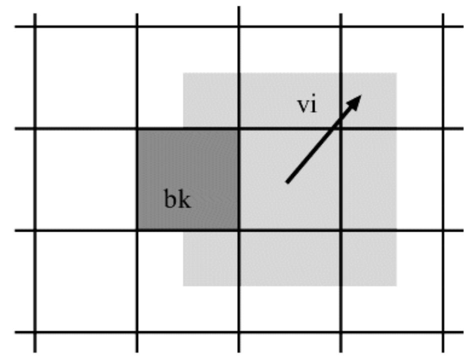
8x8
Pixel-grid
accurate
resolution

8x8
Half-pel accurate
resolution

Extensions (2)

Blocking artifacts due to discontinuities at block boundaries

- post-processing filter
- In-loop filter (Deblocking-filter → H.264)
- Prediction with overlapping blocks (OBMC)



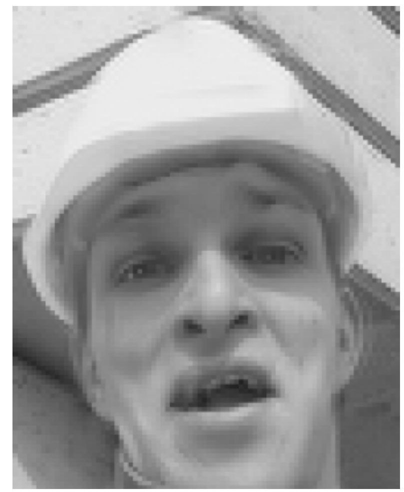
$$\hat{g}_n[\mathbf{x}] = \sum_{\mathbf{v}_i \in \mathcal{N}^2_{\mathbf{x}}} w(i) g_{n-1}[\mathbf{x} - \mathbf{v}_i]$$

w: 2D weighting function with $\sum_{x,y} w[x,y] = 1$

Optimization problem:

- displacement vectors depends on weight window
- weight window depends on displacements vector

→ iterative approach



Pixel accurate resolution with Overlap compensation