Chapter 3
Image Registration

Distributed Algorithms
for Image and Video Processing

Introduction (1)

Definition: Image Registration

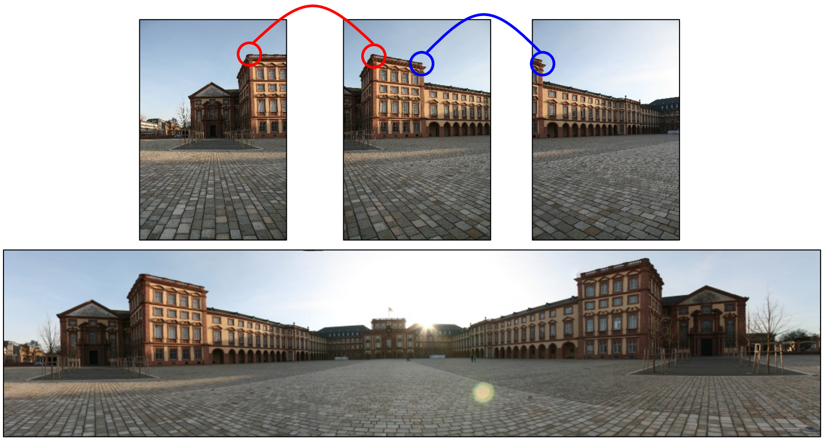
- Input: 2 images of the same scene but taken from different perspectives
- Goal: Identify transformation to present both images in the same coordinate system
- The alignment of images is called *Image Registration*

Image Registration techniques are used

- to detect camera motion
- for cut detection
- for object segmentation
- to generate panoramic images

Introduction (2)

Example: panoramic image



Content

- Identification of characteristic image features
  - Requirements
  - Moravec
  - SIFT
- Application: Analysis of camera motion
  - Camera motion model
  - Motion vectors
  - Calculation of camera model parameters
  - Validation of camera parameters
  - Description of camera motion

Characteristic image features (1)

Requirements concerning image features

- Unique description of a pixel / an image region
- Invariance to
  - Affine transform
    - Scaling
    - Rotation
    - Shear
    - Translation (shift)
    - Mirroring
    - Anisotropic scaling (different scaling factors for both dimensions)
  - Perspective transform
  - Noise
  - Luminance / contrast changes

→ Corners are good feature points in most cases

Characteristic image features (2)

***Moravec* corner detector**

- For each pixel, analyze small region (4 adjacent pixels)
- Shift (move) region one pixel and compare it with original region (sum of squared differences)
- Shift regions into 4 directions
  - Small differences in all cases
    → homogeneous region (no edges or corners)
  - Large difference in some directions
    → edge
  - Large difference in all directions
    → corner

Characteristic image features (3)

SIFT: Scale Invariant Feature Transforms

- Goal: Identify „stable " feature points in images
- Scaling, rotation, or noise should not change the feature points significantly
- Get a large number of stable feature points

Characteristic image features (4)

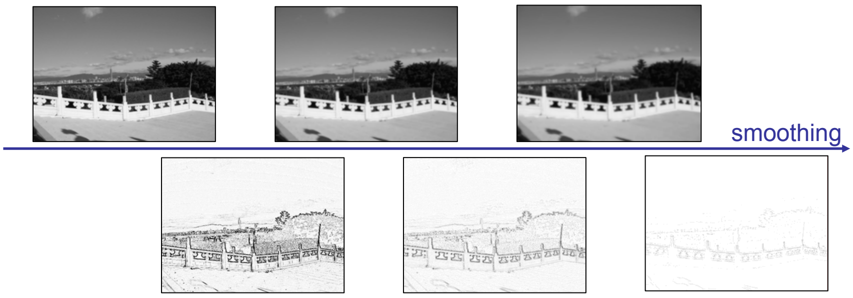
SIFT algorithm

1. Detect scale space extrema
2. Identify feature points
3. Analyze orientation of feature points
4. Get characteristic description for each feature point

Characteristic image features (5)

1. **Detect scale space extrema**

- Create several smoothed images of different intensity (based on Gaussian filter)
- Calculate difference of two smoothed images



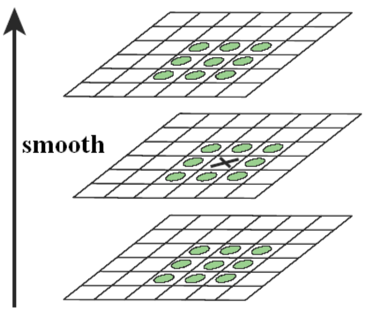
Characteristic image features (6)

1. **Detect scale space extrema**

   Validate, if pixel is a local extremum:

   – Compare pixel with 8 adjacent pixels of the same image, and

   – Compare pixel with 9 pixels of the two adjacent (differently smoothed) images



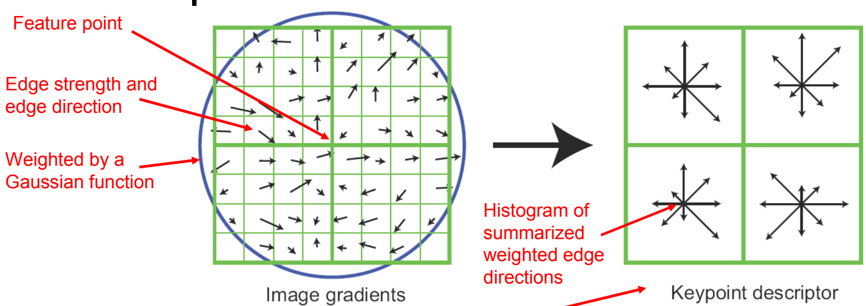
Source:
Lowe: Distinctive Image Features from Scale-Invariant Keypoints

Characteristic image features (7)

2. **Identify feature points**
   - Remove pixel with low contrast
   - Remove edge pixel (see 3.)
3. **Analyze orientation of feature points**
   - Calculate edge strength (see *Canny*)
   - Calculate edge direction (see *Canny*)
   - Get histogram for each feature point based on the edge strength and edge direction of adjacent pixel values
   - Identify dominant edge direction (largest value in histogram)
   - Normalize histogram based on the dominant edge direction

Characteristic image features (8)

4. **Get characteristic description for each feature point**



Keypoint descriptor aggregates all histograms
Original approach: 4x4 regions (in this example: 2x2) with 16 elements in each region
→ A SIFT feature vector has 128 dimensions

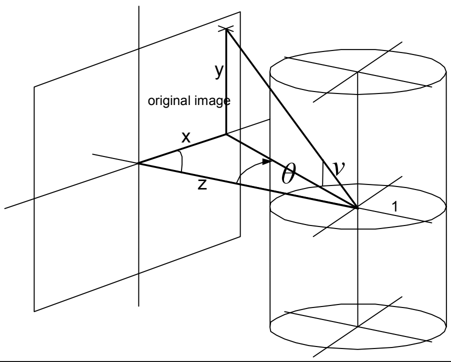
Source:
Lowe: Distinctive Image Features from Scale-Invariant Keypoints

Application: Camera motion

- Camera motion model
  - *Cylindrical model*
  - *Spherical model*
  - *8 parameter model*
- Motion vectors
- Calculation of camera model parameters
- Validation of camera parameters
- Analysis of camera motion

Cylindrical camera model (1)

- Map two images into one coordinate system
- The cylindrical model supports a horizontal rotation of the camera
- The transformed image is represented by cylindrical coordinates



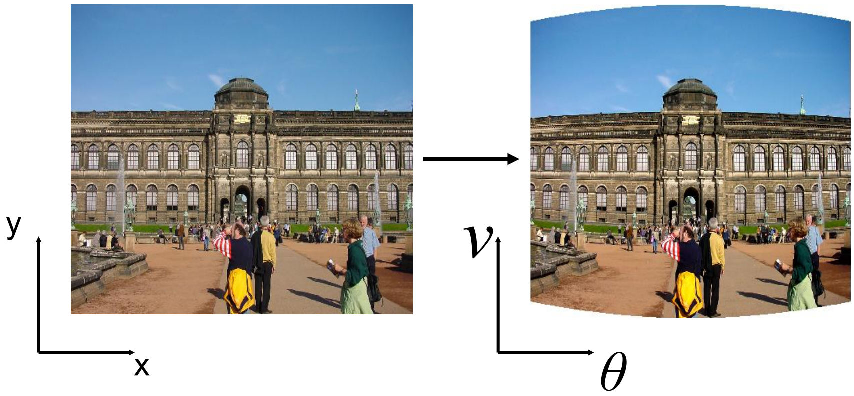
angle: $\theta = \arctan(x/z)$

height: $v = y/\sqrt{x^2+z^2}$

Focal length: *z*

Cylindrical camera model (2)

Example



Cylindrical camera model (3)

Calculate horizontal / vertical shift of two transformed images

Shift one image until pixel differences of overlapping pixels are minimized

Cylindrical camera model (4)

Estimate translation

- Unknown: Vector (t_x, t_y)
- Error E between images f and g is minimized:

$$E(t_x, t_y) = \sum_{x,y} | f(x,y) - g(x+t_x, y+t_y) |^2$$

- Complexity: full search with all possible translations

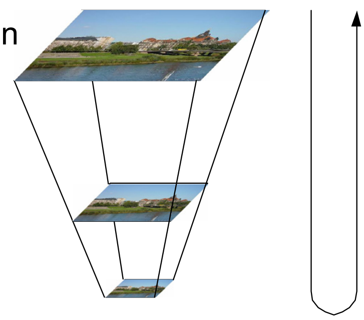
$$O(N^2 M^2) \approx O(N^4)$$ → large effort

image size: N x N, search radius: M x M

Cylindrical camera model (5)

Hierarchical approach

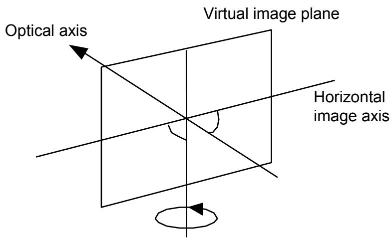
- Build pyramid of images in different resolutions
- Calculate translation for image with smallest resolution
- Scale shift parameters to next higher level, and calculate small delta of shift parameters in the current image



Cylindrical camera model (6)

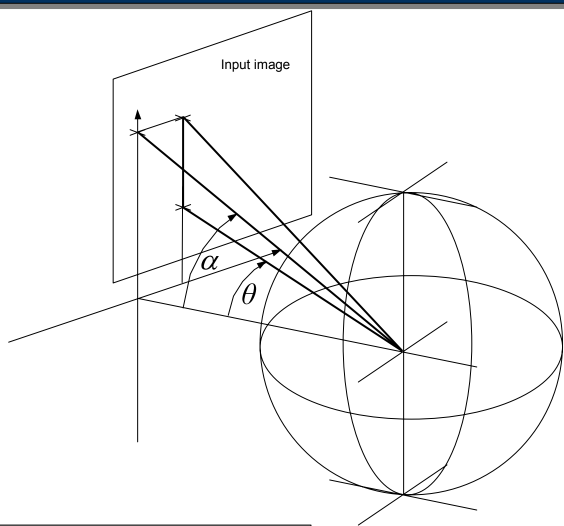
General disadvantages

- Straight lines become curved
- The focal length (zoom) must be known in advance
- Only horizontal rotations can be handled



Spherical camera model

- Mapping of the image to a sphere
- Model supports horizontal and vertical camera rotations



8 Parameter model (1)

Assumption

- Image is painted on a flat sheet: Which motion of the sheet is possible?

2D

- A 6 parameter model specifies *affine motion* (translation, rotation, scaling, [mirroring, shear]):

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

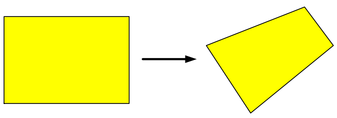
Rotation
Scaling
[Shear]
[Mirroring]

Translation

8 Parameter model (2)

3D

- The parameters $a_{i,j}$ *and* t_x, t_y describe affine motion
- Additionally: Perspective transformation



- 2 additional parameters b_1 *and* b_2 describe perspective deformations

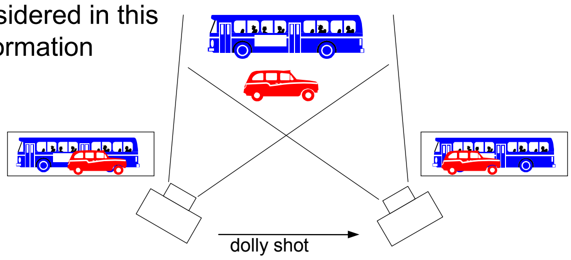
$$x' = \frac{a_{11}x + a_{12}y + t_x}{b_1 x + b_2 y + 1} \qquad y' = \frac{a_{21}x + a_{22}y + t_y}{b_1 x + b_2 y + 1}$$

8 Parameter model (3)

- Horizontal / vertical shift (horizontal pan or vertical tilt): t_x, t_y
- Zoom and rotation along the optical axis: $a_{i,j}$
- Perspective transformation: b_1, b_2
- Dolly shots (camera changes position) are not considered in this model (3D scene information would be required)

$$x' = \frac{a_{11}x + a_{12}y + t_x}{b_1 x + b_2 y + 1}$$

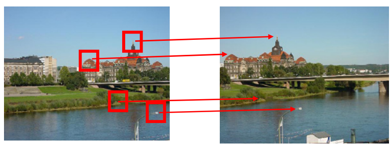
$$y' = \frac{a_{21}x + a_{22}y + t_y}{b_1 x + b_2 y + 1}$$



8 Parameter model (4)

Estimation of camera parameters

- Detect 4 similar blocks in both images



- A motion vector describes the horizontal and vertical shift of each block (pixel in the center)
- The exact shift (x and y coordinate) of 4 blocks sufficient to calculate the 8 parameters of the camera model

Motion vectors (1)

- A **motion vector** describes the horizontal and vertical shift of a pixel between two images
- **Optical flow** describes the motion of all pixels

Estimation of motion vectors from image *i* to *j* based on **block matching**

1. Define rectangular regions in image *i*
2. For each region, identify the most similar region in image *j*
3. For a block at position *(x, y)* in image *i*, we consider all pixels *(x', y')* in image *j* with a maximum horizontal and vertical distance *r* :

$$\begin{aligned} x - r &\leq x' \leq x + r \\ y - r &\leq y' \leq y + r. \end{aligned}$$

Motion vectors (2)

Similarity of blocks

- Sum of absolute differences of two blocks (e.g., 16 x 16 pixels):

$$SAD = \sum_{\Delta x=-8}^{7} \sum_{\Delta y=-8}^{7} |I_i(x + \Delta x, y + \Delta y) - I_j(x' + \Delta x, y' + \Delta y)|.$$

- The position (x′, y′) minimizes the value of *SAD*. The motion vector of this block starts at (x, y) and ends at (x′, y′).

Disadvantage

- High computational effort
- Inaccurate motion vectors in regions with low texture

Motion vectors (3)

Improvements

- Identify characteristic features in the first image and locate them in the second:
  - Moravec: A block is selected for each detected corner
  - SIFT: Features are characterized by a feature vector of 128 dimensions (keypoint descriptor)
- The motion vector describes the position change of a feature

Motion vectors (4)

Assignment of corners (Moravec)

1. Analyze all combinations of corners in both images (N x M combinations)
2. A corner defines the center of a block (*SAD* is used as image similarity metric)
3. The two most similar corners are assigned to each other, and removed from the list (they define a motion vector)
4. Additional motion vectors are defined based on the residual corners (repeat step 3)
5. The algorithm stops if *SAD* is above a certain threshold (we assume that additional assignments are no longer correct)

Motion vectors (5)

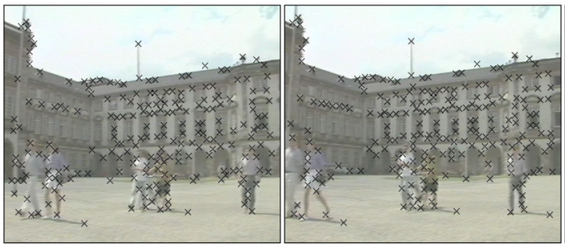
Example :
Block matching (block size: 16x16)

2 images of a video sequence

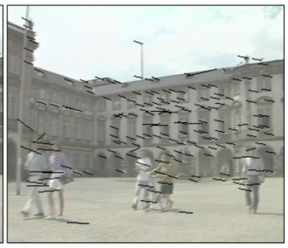
motion vectors (block matching)

Motion vectors (6)

Example: Assignment of corners



detected corners



motion vectors

Problems

- Incorrect assignment of corners
- Object motion

Calculation of the camera model (1)

Goal

- Identify (choose) motion vectors, that describe the camera motion (not motion caused by foreground objects)

Approach

- *Assumption*: at least half of the vectors describe background motion
- Calculate camera motion using a robust regression technique (*least trimmed squares*)
  1. Randomly select 4 motion vectors from the list of all motion vectors
  2. Use these vectors to solve the linear system of equations of the camera model.
     Solution of the linear system: 8 parameters that describe the camera model between two frames

Calculation of the camera model (2)

3. The error function e_i classifies the error for each motion vector *i* between the real position of pixel *(x', y')* of the second image and the estimated position of the camera model $(\hat{x}, \hat{y})$.

$$e_i \;=\; (x'_i - \hat{x}_i)^2 + (y'_i - \hat{y}_i)^2.$$

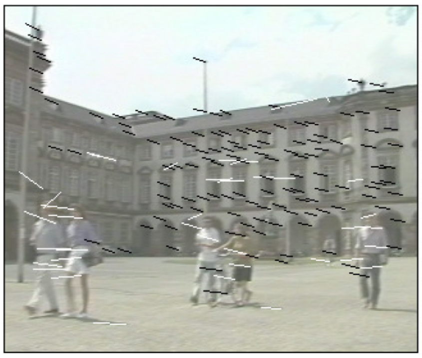
4. Vectors of foreground objects should not have a large impact on the error function.
   Idea: Sort errors based on difference values (vectors with lowest errors come first)
5. Calculate total error *E* based on 50 percent of the best matching motion vectors:

$$E \;=\; \sum_{i=1}^{N/2} e_i \qquad \text{mit } e_1 \leq \cdots \leq e_N.$$

6. Continue with step 1, until the total error is below a threshold, or until a predefined number of iterations has passed.

Calculation of the camera model (3)

Example



bright vectors: significant deviation from model
dark vectors: minor deviation from model

calculated camera model

Typical errors

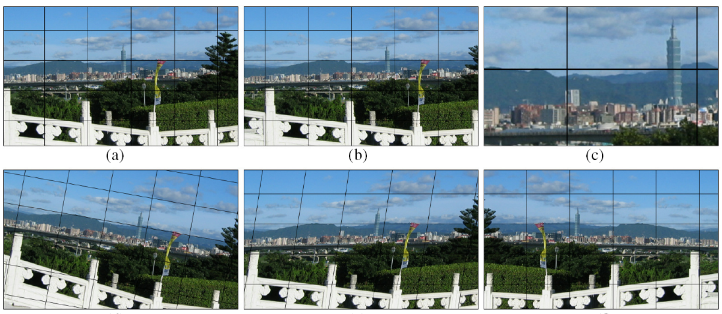
The camera model is not correct in case of

- regions without structure
  → block matching errors
- low contrast
  → small number of corners
- periodic structures in the image
  → motion vectors are not correct
- large foreground objects
  → more than 50 % of the motion vectors describe foreground objects

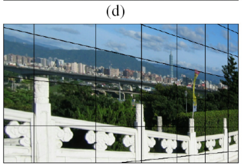
Validation of the parameters (1)

Camera motion	t_x, t_y	a_{11}, a_{22}	a_{12}, a_{21}	p_x, p_y
Static camera	0	1	0	0
Translation	$\neq 0$	1	0	0
Scaling - zoom in - zoom out	 0 0	 $0 < a_{11} = a_{22} < 1$ $a_{11} = a_{22} > 1$	 0 0	 0 0
Rotation, angle θ	0	$a_{11} = a_{22} = \cos\theta$	$a_{12} = -a_{21} = \sin\theta$	0
Shear - horizontal - vertical	 0 0	 1 1	 $a_{12} \neq 0$ $a_{21} \neq 0$	 0 0
Mirroring - horizontal - vertical	 0 0	 $a_{11} = -1$ $a_{22} = -1$	 0 0	 0 0
Persp. transform	0	1	0	$\neq 0$

Validation of the parameters (2)



(a) (b) (c) (d) (e) (f)



(g)

(a) no transformation (static)
(b) translation ($t_x = -0,1 \cdot$ width)
(c) scaling Zoom–In ($a_{11} = a_{22} = 0,33$)
(d) rotation: 10 degrees ($a_{11} = a_{22} = 0,985$, $a_{12} = -a_{21} = 0,174$)
(e) shear (not possible in videos) ($a_{12} = 0,15$)
(f) mirroring (not possible in videos) ($a_{22} = -1$)
(g) persp. transform ($p_x = -4 \cdot 10^{-5}$, $p_y = 0$)

Validation of the parameters (3)

Typical values of the parameters

Camera motion	t_x, t_y	a_{11}, a_{22}	a_{12}, a_{21}	p_x, p_y
static camera	$0 \pm 0{,}8$	$1 \pm 0{,}01$	$0 \pm 0{,}01$	$0 \pm 1 \cdot 10^{-6}$
horizontal pan	$0 \pm \frac{1}{5}W$	$1 \pm 0{,}02$	$0 \pm 0{,}02$	$0 \pm 2 \cdot 10^{-4}$
zoom	$0 \pm 0{,}8$	$1 \pm 0{,}08$	$0 \pm 0{,}08$	$0 \pm 1 \cdot 10^{-5}$
rotation (max. $\theta = 5^o$)	$0 \pm 0{,}8$	$1 \pm 0{,}01$	$0 \pm 0{,}09$	$0 \pm 1 \cdot 10^{-5}$

W: image width

Validation of the parameters (4)

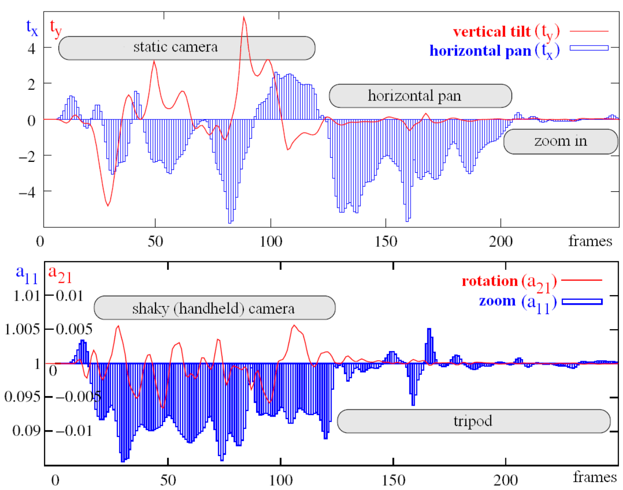
Recognition rates

Camera model	%	Reason
correct	94,8 %	
edges not detected	0,3 %	low contrast
wrong assignment of corners	0,1 %	hard cuts, sudden changes in video
incorrect camera model	4,8 %	hard and soft cuts, object motion

Description of camera motion (1)

- The camera model parameters are used to describe the start, length, and strength of a camera operation
- Camera rotation gives feedback about the way of recording (e.g., handheld camera or tripod)

Description of camera motion (2)



Description of camera motion (3)

Comparison of 9 test sequences

	horizontal pan	vertical pan	incoming zoom	outgoing zoom
Documentation	31	12	12	21
News	40	18	14	30
Action movie	32	4	15	33
Talkshow	41	9	28	48
Series	18	11	19	24
Animated film	3	1	2	16
Sports	81	7	13	28
Music clip	27	10	10	24
Advertisments	18	19	18	20
Sum	**301**	**88**	**123**	**254**

Questions ?