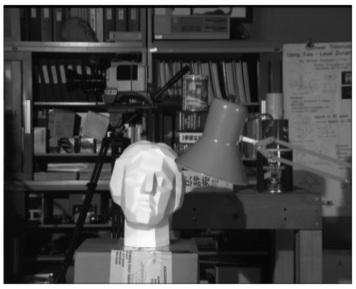
Lecture 2:
Block Matching, Correlation Techniques Interest Points, Feature-Based Methods

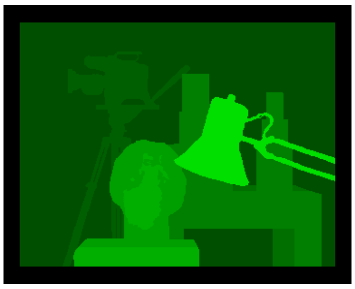
Contents

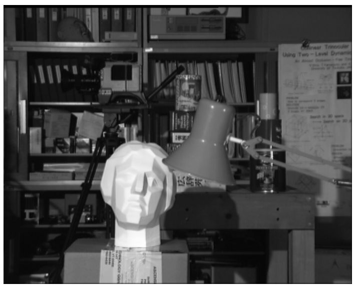
Basic Approach

In this lecture we do not focus on special types of correspondence problems. We consider the following general case:

Given: Two images $\mathbf{f}$ and $\mathbf{g}$, where $f_{i,j}$ is the grey value of image $\mathbf{f}$ at pixel i, j.
Wanted: Displacement field $\mathbf{u}$ between both frames.


(a) Left: Frame 1.


(b) Middle: Displacement Field.


(c) Right: Frame 2.

(Taken from the Middlebury Stereo Evaluation)

Basic Approach (2)

How Can We Retrieve Pixels?

- Constraints on the Data $\rightarrow$ Constancy assumptions
  - There may be certain features of a pixel that do not change over time
  - These features should be as unique as possible (to avoid multiple matches)
  - For our basic approach let us consider the **grey value constancy**:

$$f_{i,j} - g_{i+u,j+v} = 0$$

For certain pixels the solution may be non-unique or even non-existing !

How Can We Always Find a Solution?

- Minimisation Problem with Cost Function $\rightarrow$ Allow deviations from constancy
  - Select solution that is as optimal as possible, i.e. that minimises

$$\mathbf{u}_{i,j} = \underset{u,v}{\operatorname{argmin}} \, (f_{i,j} - g_{i+u,j+v})^2$$

Basic Approach (3)

How Can We Restrict the Search Space?

- Constraints on the Solution $\rightarrow$ Only search in a certain neighbourhood
  - If one knows that the displacements are limited by $|u_i|, |v_i| \leq d$, one can minimise

$$\mathbf{u}_{i,j} = \underset{u,v \in \mathcal{S}_d}{\operatorname{argmin}} \quad (f_{i,j} - g_{i+u,j+v})^2$$

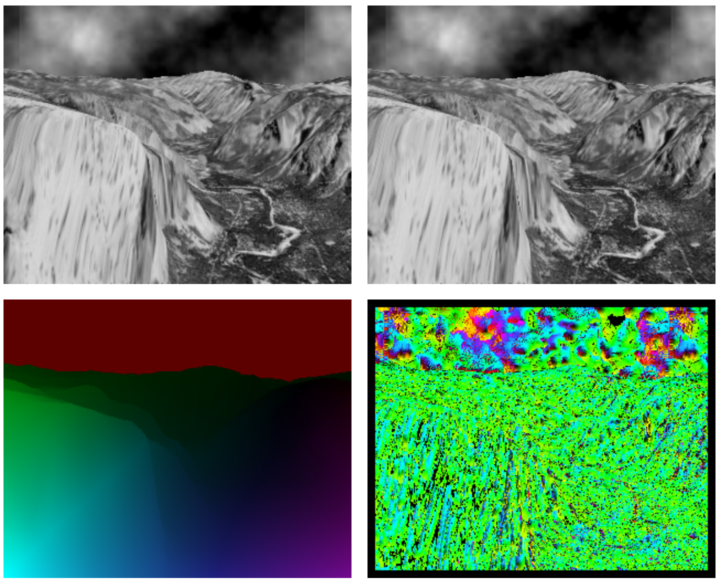
  - $\mathcal{S}_d$: Search window of size $(2d+1) \times (2d+1)$ around the pixel i, j

How Do We Find the Optimal Solution?

- Minimisation Strategy $\rightarrow$ Exhaustive search
  - For each pixel i, j check all $(2d+1)^2$ possible displacements
  - Linear complexity, i.e. $O(\underbrace{(2d+1)^2}_{\text{large constant}} NM)$, but still poor efficiency

Basic Approach (4)

Results for the Basic Matching Approach



Matching results for the Yosemite Sequence with clouds (L. Quam). **(a) Upper Left:** Frame 8. **(c) Upper Right:** Frame 9. **(d) Lower Left:** Ground truth. **(f) Lower Right:** Basic approach ($d = 7$).

Block Matching (1)

Block Matching

- *Idea:* Consider small neighbourhoods instead of single features for matching

	$i-1$	i	$i+1$	$i+2$	$i+3$
$j-1$	**22**	**21**	**26**	28	42
j	**20**	**3**	**22**	36	31
$j+1$	**23**	**22**	**34**	31	37
$j+2$	25	22	31	38	38
$j+3$	31	34	34	31	33

Image **f**

	$i-1$	i	$i+1$	$i+2$	$i+3$
$j-1$	32	31	35	34	31
j	41	**23**	**23**	**46**	28
$j+1$	33	**24**	**4**	**32**	36
$j+2$	34	**25**	**54**	**24**	31
$j+3$	35	23	22	31	48

Image **g**

- Compares all grey values of a reference block in the first image to the corresponding values of differently shifted candidate blocks in the second image
- Two degrees of freedom
  - Block size m that specifies $f_{i+\Delta i, j+\Delta j}$ with $\Delta i, \Delta j \in \{-m, ..., +m\}$
  - Max displacement d that limits the search space via $u, v \in \{-d, ..., +d\}$

Block Matching (2)

Sum of Squared Differences Model (SSD)

- Mathematical formulation with quadratic cost function

$$\mathbf{u}_{i,j} = \underset{u,v \in \mathcal{S}_d}{\operatorname{argmin}} \sum_{\substack{\Delta i,\,\Delta j \\ \in \mathcal{N}_m}} \left(f_{i+\Delta i, j+\Delta j} - g_{(i+u)+\Delta i,(j+v)+\Delta j}\right)^2$$

  - $\mathcal{N}_m$: Neighbourhood of size m, i.e. patch of size $(2m+1) \times (2m+1)$
  - $\mathcal{S}_d$: Search space for u, v limited by the maximum displacement d

- *Step 1:* Compute the cost map for each pixel i, j
  - Example for $d = 2$

$v_{i,j} \diagdown u_{i,j}$	-2	-1	0	1	2
-2	72	71	76	78	92
-1	70	53	72	86	81
0	73	72	84	81	87
1	75	72	81	88	88
2	81	84	84	81	83

Block Matching (3)

Sum of Squared Differences Model (SSD)

- *Step 2:* Select displacement $\mathbf{u}_{i,j}$ with minimal cost
  - Example for $d = 2$

$\mathbf{u}_{i,j} = (-1, -1)^\top$

$v_{i,j} \diagdown u_{i,j}$	-2	-1	0	1	2
-2	72	71	76	78	92
-1	70	**53**	72	86	81
0	73	72	84	81	87
1	75	72	81	88	88
2	81	84	84	81	83

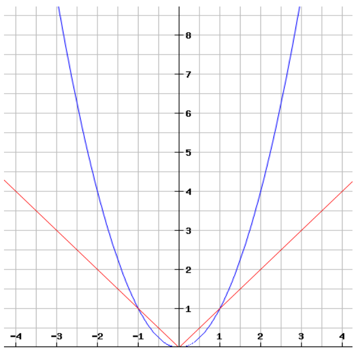
- Advantages
  - Simple approach with straightforward implementation, robust under noise
- Drawbacks
  - Non-unique solution: multiple best matches (e.g. in homogeneous areas)
  - Poor quality: no sub-pixel precision, no illumination invariance
    mainly works for translations, no discontinuity preservation, ...
  - Poor efficiency: 2-D search space too large $\rightarrow O(\underbrace{(2m+1)^2(2d+1)^2}_{\text{large constant}} NM)$

Block Matching (4)

Can we improve the performance with respect to outliers?

- *Idea:* Reduce the influence of outliers

- Less-than-quadratic penalisation
  Example: quadratic vs. linear



- General model with positive increasing penaliser function $\Psi(s^2)$

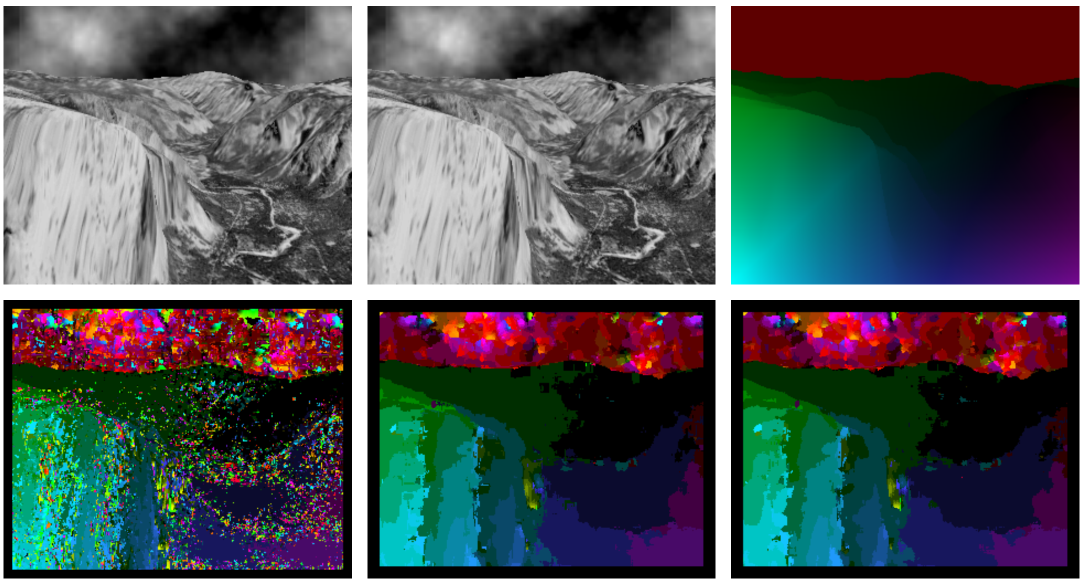
$$\mathbf{u}_{i,j} = \underset{u,v \in \mathcal{S}_d}{\operatorname{argmin}} \sum_{\substack{\Delta i,\, \Delta j \\ \in \mathcal{N}_m}} \Psi\Big(\big(f_{i+\Delta i, j+\Delta j} - g_{(i+u)+\Delta i,(j+v)+\Delta j}\big)^2 \Big)$$

- Special instance for $\Psi(s^2) = |s|$: Sum of Absolute Differences Model (SAD)

$$\mathbf{u}_{i,j} = \underset{u,v \in \mathcal{S}_d}{\operatorname{argmin}} \sum_{\substack{\Delta i,\, \Delta j \\ \in \mathcal{N}_m}} \big| f_{i+\Delta i, j+\Delta j} - g_{(i+u)+\Delta i,(j+v)+\Delta j} \big|$$

Block Matching (5)

Block Matching Results



Block matching results for the Yosemite Sequence with clouds (L. Quam). **(a) Upper Left:** Frame 8. **(b) Upper Middle:** Frame 9. **(c) Upper Right:** Ground truth. **(d) Lower Left:** SSD ($n = 1, d = 7$). **(e) Lower Middle:** SSD ($n = 4, d = 7$). **(f) Lower Right:** SAD ($n = 4, d = 7$).

Correlation Techniques

- *Idea:* Consider only the mixed term in SSD that describes a scalar product

$$\sum (f-g)^2 = \sum f^2 - 2\sum fg + \sum g^2$$

- Cross Correlation

$$\mathbf{u}_{i,j} = \underset{u,v\in\mathcal{S}_d}{\operatorname{argmax}} \sum_{\substack{\Delta i,\Delta j\\ \in \mathcal{N}_m}} f_{i+\Delta i,j+\Delta j} \cdot g_{(i+u)+\Delta i,(j+v)+\Delta j}$$

- Mean Compensated Cross Correlation

$$\mathbf{u}_{i,j} = \underset{u,v\in\mathcal{S}_d}{\operatorname{argmax}} \sum_{\substack{\Delta i,\Delta j\\ \in \mathcal{N}_m}} (f_{i+\Delta i,j+\Delta j} - \bar{f}_{i,j}) \cdot (g_{(i+u)+\Delta i,(j+v)+\Delta j} - \bar{g}_{i+u,j+v})$$

where $\bar{f}_{i,j}$ and $\bar{g}_{i+u,j+v}$ are the mean of the reference and the candidate block

Normalised Cross Correlation (NCC)

- Mean compensated cross correlation with additional normalisation
(e.g. Musmann/Pirsch/Grallert 1985)

$$\mathbf{u}_{i,j} = \underset{u,v\in\mathcal{S}_d}{\operatorname{argmax}} \frac{\sum_{\substack{\Delta i,\Delta j\\ \in \mathcal{N}_m}} (f_{i+\Delta i,j+\Delta j} - \bar{f}_{i,j}) \cdot (g_{(i+u)+\Delta i,(j+v)+\Delta j} - \bar{g}_{i+u,j+v})}{\sqrt{\sum_{\substack{\Delta i,\Delta j\\ \in \mathcal{N}_m}} (f_{i+\Delta i,j+\Delta j} - \bar{f}_{i,j})^2}\sqrt{\sum_{\substack{\Delta i,\Delta j\\ \in \mathcal{N}_m}} (g_{(i+u)+\Delta i,(j+v)+\Delta j} - \bar{g}_{i+u,j+v})^2}}$$

- Geometric interpretation: $\frac{\mathbf{a}^\top \mathbf{b}}{|\mathbf{a}||\mathbf{b}|} = \cos(\alpha_{\angle(\mathbf{a},\mathbf{b})})$
  - Normalised inner product, where $\mathbf{a}$ is the mean compensated reference block and $\mathbf{b}$ is the mean compensated candidate block. Range of values $[-1,1]$.
- Statistical interpretation: $\frac{\operatorname{cov}(\mathbf{a},\mathbf{b})}{\sqrt{\operatorname{var}(\mathbf{a})}\sqrt{\operatorname{var}(\mathbf{b})}}$
  - Ratio, where cov is the covariance between the reference block $\mathbf{a}$ and the candidate block $\mathbf{b}$ and var denotes the variance. Range of values $[-1,1]$.

Correlation Techniques (3)

Normalised Cross Correlation (NCC)

- Same advantages and shortcomings as the SSD and SAD approach. However, the NCC is invariant under global linear illumination changes:

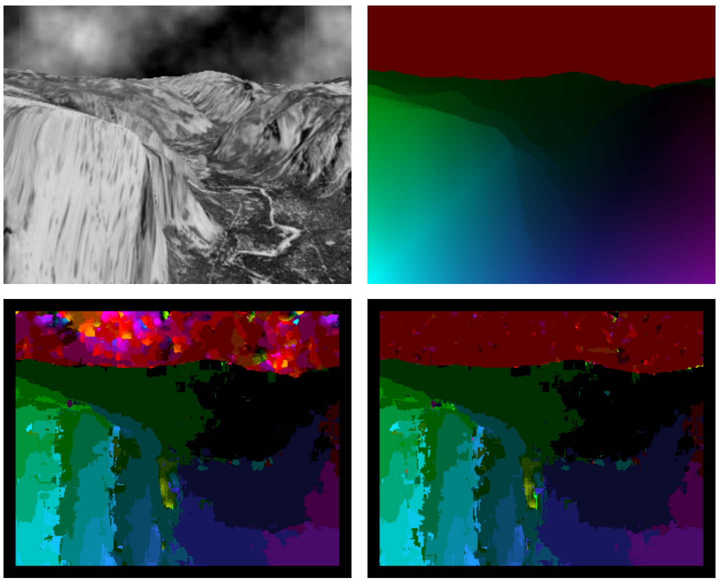
$$f^*_{i,j} = a f_{i,j} + b$$

  - Mean compensation removes the additive component b
  - Normalisation cancels out the multiplicative component a

- There is also a price to pay for this invariance
  - NCC cannot be used with a mask size of 0, i.e. it is not possible to consider the central pixel only
  - NCC is not defined in homogeneous areas, i.e in areas where the variance of one of the patches becomes zero

Cross Correlation (4)

Normalised Cross Correlation Results



Block matching results for the Yosemite Sequence with clouds (L. Quam). **(a) Upper Left:** Frame 8. **(c) Upper Right:** Ground Truth. **(d) Lower Left:** SSD ($n=4, d=7$). **(f) Lower Right:** Normalised Cross Correlation ($n=4, d=7$).

Sub-Pixel Refinement (1)

M I A 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34

Sub-Pixel Refinement

How Can We Obtain Sub-Pixel Precision from Integer Displacements?

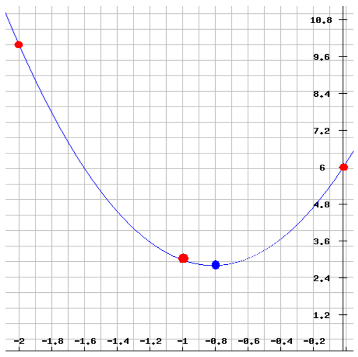
- *Idea:* Fit a continuous function to the cost map and recompute the minimum
- Common choice for SSD in 1-D: Fit the parabola $f(x) = ax^2 + bx + c$

Displacement u_i	-2	-1	0	1	2
Matching Cost	10	**3**	6	15	22
	c_{-1}	c_0	c_{+1}		

c_0 : cost for the best match u_i

c_{-1}: left neighbour of c_0 in the cost map

c_{+1}: right neighbour of c_0 in the cost map



Sub-Pixel Refinement (2)

M I A 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34

Sub-Pixel Refinement in 1-D

- The parabola through c_{-1}, c_0 and c_1 centered at c_0 is given by the second order Taylor approximation

$$f(x) = \underbrace{\frac{c_{+1} - 2c_0 + c_{-1}}{2}\, x^2}_{\approx \frac{1}{2}\, c''(0)(x-0)^2} + \underbrace{\frac{c_{+1} - c_{-1}}{2}\, x}_{\approx c'(0)(x-0)} + c_0$$

- Instead of u_i, we then obtain the new location of the minimum at $u_i + \Delta u_i$, where the sub-pixel correction reads

$$\Delta u_i = \frac{c_{-1} - c_{+1}}{2(c_{+1} - 2c_0 + c_{-1})}$$

- In our example we obtain $\Delta u_i = 0.2$. This shifts the minimum from $u_i = -1$ to $u_i = -0.8$.

Sub-Pixel Refinement (3)

How Can This Idea Be Extended to 2-D?

- Analogously for SSD in 2-D: Fit a paraboloid to the local cost map
  - Instead of a parabola $f(x) = ax^2 + bx + c$, we now consider the paraboloid

$$f(x,y) = ax^2 + bxy + cy^2 + dx + ey + f$$

  - To fit this paraboloid to the local cost map we need six cost values
  - However, typically neighbourhood sets of five or nine pixels are used in 2-D

$v_{i,j} \setminus u_{i,j}$	-2	1	0	1	2
-2	22	**71**	76	78	92
-1	**70**	**53**	**72**	86	81
0	73	**72**	84	81	87
1	75	72	81	88	88
2	81	84	84	81	83

Five-point-stencil

$v_{i,j} \setminus u_{i,j}$	-2	1	0	1	2
-2	**72**	**71**	**76**	78	92
-1	**70**	**53**	**72**	86	81
0	**73**	**72**	**84**	81	87
1	75	72	81	88	88
2	81	84	84	81	83

Nine-point-stencil

Sub-Pixel Refinement (4)

How Can We Solve The Problem Which Neighbours to Chose in 2-D?

- Instead of performing a least squares fit with nine values, we simplify the model by setting the mixed term bxy to zero with $b = 0$

$$f(x,y) = ax^2 + cy^2 + dx + ey + f$$

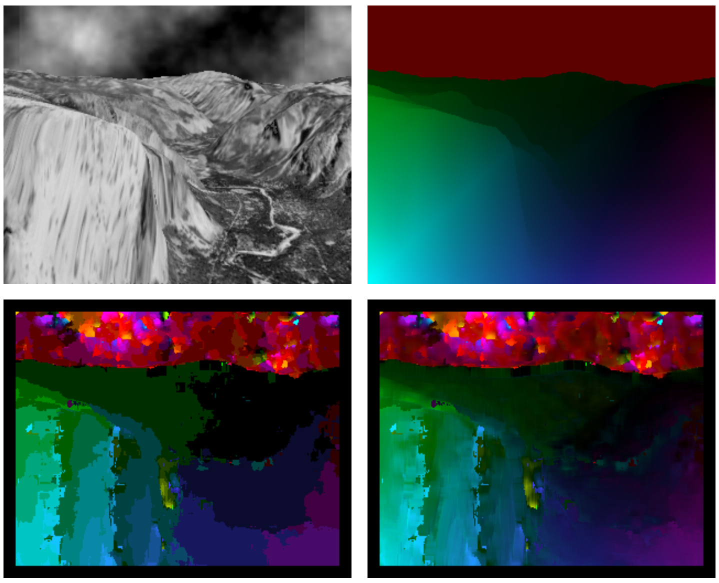
- Then, for a neighbourhood set of five pixels, only two orthogonal parabolas through $c_{0,0}$ must be fitted

$$\begin{aligned} f(x,0) &= ax^2 + dx + f \\ f(0,y) &= cy^2 + ey + f \end{aligned}$$

- Apply formula for the 1-D case twice:
  - with $c_{0,0}$, $c_{+1,0}$ and $c_{-1,0}$ for sub-pixel displacement in x-direction
  - with $c_{0,0}$, $c_{0,+1}$ and $c_{0,-1}$ for sub-pixel displacement in y-direction

Sub-Pixel Refinement (4)

Block Matching Results with Sub-Pixel Refinement



Block matching results for the Yosemite Sequence with clouds (L. Quam). **(a) Upper Left:** Frame 8. **(c) Upper Right:** Ground truth. **(d) Lower Left:** SSD $(n=4, d=7)$. **(f) Lower Right:** SSD + Subpixel Refinement $(n=4, d=7)$.

Occlusion Detection (1)

Occlusion Detection

How Can Occlusions Be Identified?

- Observations for Occluded Pixels
  - The cost for the best match is very high (poor match)
  - The best match is not very discriminative (w.r.t. to the other matches)
- Example (best match in **red**, second best match in **blue**)

$v_{i,j} \backslash u_{i,j}$	-2	1	0	1	2
-2	72	71	76	78	92
-1	**70**	**53**	72	86	81
0	73	72	84	81	87
1	75	72	81	88	88
2	81	84	84	81	83

Dominant match/Low cost

$v_{i,j} \backslash u_{i,j}$	-2	1	0	1	2
-2	87	89	89	**84**	87
-1	85	**83**	88	86	91
0	87	85	94	89	87
1	95	95	81	88	88
2	85	86	94	87	86

Non-dominant match/High cost

Occlusion Detection (2)

M I A 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34

How can we exploit this knowledge?

- Forward/Backward-Check
  - Compute forward displacements $\mathbf{u}^{\text{fw}}$ from image $\mathbf{f}$ to image $\mathbf{g}$
  - Compute backward displacements $\mathbf{u}^{\text{bw}}$ from image $\mathbf{g}$ to image $\mathbf{f}$
  - Evaluate for each pixel i, j the magnitude of the difference vector $\Delta\mathbf{u}_{i,j}$ between both displacements given by

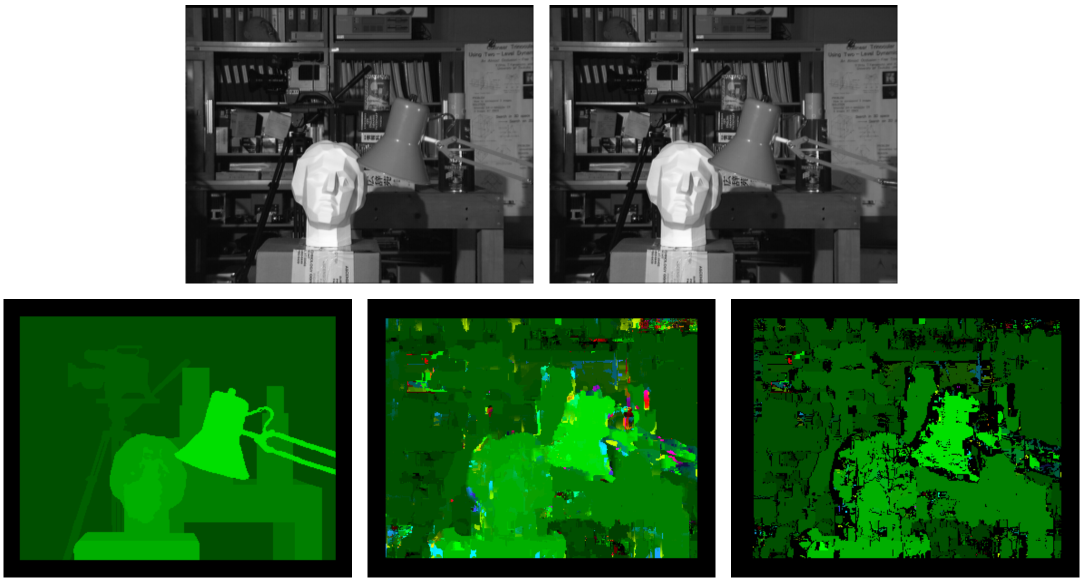
$$|\Delta\mathbf{u}_{i,j}| = \sqrt{\left(u^{\text{fw}}_{i,j} + u^{\text{bw}}_{i+u^{\text{fw}}_{i,j},\,j+v^{\text{fw}}_{i,j}}\right)^2 + \left(v^{\text{fw}}_{i,j} + v^{\text{bw}}_{i+u^{\text{fw}}_{i,j},\,j+v^{\text{fw}}_{i,j}}\right)^2}$$

  - Consider all pixels with $|\Delta\mathbf{u}|$ larger than a treshold T_{occ} as occluded
  - In practise often $T_{\text{occ}} = 0$ is used, i.e. consistency is enforced
  - One can further improve the results by discarding matches with high matching costs

Occlusion Detection (3)

M I A 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34

Block Matching Results with Occlusion Detection



Block matching results for the Tsukuba stereo pair (Middlebury stereo evaluation). **(a) Upper Left:** Left frame. **(b) Upper Right:** Right frame.. **(c) Lower Left:** Ground truth. **(d) Lower Center:** SSD $(n{=}4, d{=}16)$. **(f) Lower Right:** SSD + Occlusion Detection $(n{=}4, d{=}16, T_{\text{occ}} = 0)$.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34

Interest Points

Are there image features that are more characteristic than the grey value?

- *Idea:* Select points that have a higher information content or are better localised
- Typical selection: Corner points
- Corner points can be computed via the structure tensor
  (Förstner/Gülch 1987, Harris/Stevens 1988)

$$J_\rho = K_\rho * (\nabla f \nabla f^\top) = \begin{pmatrix} K_\rho * f_x^2 & K_\rho * f_x f_y \\ K_\rho * f_x f_y & K_\rho * f_y^2 \end{pmatrix}$$

- ∇f: spatial gradient $(f_x, f_y)^\top$ with $f_x = \frac{\partial}{\partial x} f$
- K_ρ: gaussian kernel of standard deviation ρ
- $*$: convolution operator
- derivatives approximated e.g. by central differences, i.e. $(f_x)_i = \frac{f_{i+1,j} - f_{i-1,j}}{2hx}$

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34

Interpretation of the Structure Tensor

- Integrates gradient information from a neighbourhood of fixed size
- Eigenvalue decomposition with eigenvalues $\lambda_1 \geq \lambda_2$ and corresponding eigenvectors $\mathbf{e}_1$ and $\mathbf{e}_2$ gives information on the local structure of this neighbourhood:

$$J_\rho = (\mathbf{e}_1, \mathbf{e}_2) \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} (\mathbf{e}_1, \mathbf{e}_2)^\top$$

• $\lambda_1 \approx 0,\ \lambda_2 \approx 0$	–	no local structure	$\rightarrow$ homogeneous area
• $\lambda_1 \gg 0, \lambda_2 \approx 0$	–	variation in one direction	$\rightarrow$ edge
• $\lambda_1 \gg 0, \lambda_2 \gg 0$	–	variation in both directions	$\rightarrow$ corner

- Measures for detecting corners
  - $c = \lambda_2 \overset{!}{=} \max$ *(Tomasi/Kanade 1991)*
  - $c = \det J_\rho = \lambda_1 \lambda_2 \overset{!}{=} \max$ *(Rohr 1987)*
  - $c = \det J_\rho - k \operatorname{tr}^2 J_\rho = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 \overset{!}{=} \max$ *(Förstner 1986, Harris 1988)*

Interest Points (3)

Which Information to Consider at Interest Points?

- Consider neighbourhood (as in block matching algorithms)
- Not invariant under scaling/zoom, rotation and illumination changes

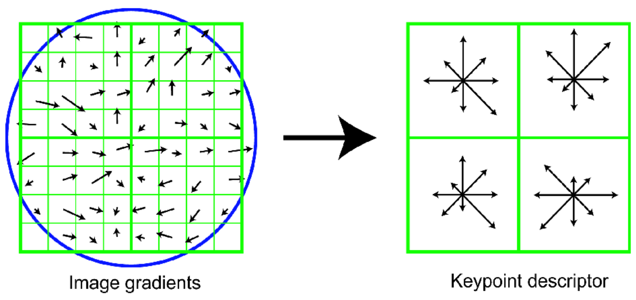
Determine neighbourhood relative to location, scale and orientation

- Scale Invariant Feature Transform (SIFT)
  (Lowe 1999, Lowe 2004)
  - computes precise location and scale of each corner
    (multi-scale corner detection + sub-pixel refinement)
  - computes dominant orientation information for the neighbourhood
    (local statistics on the image gradient ∇f)
  - creates a unit feature vector of size 128 for each corner neighbourhood
    (from image data compensated by location, scale and dominant orientation)

Interest Points (4)

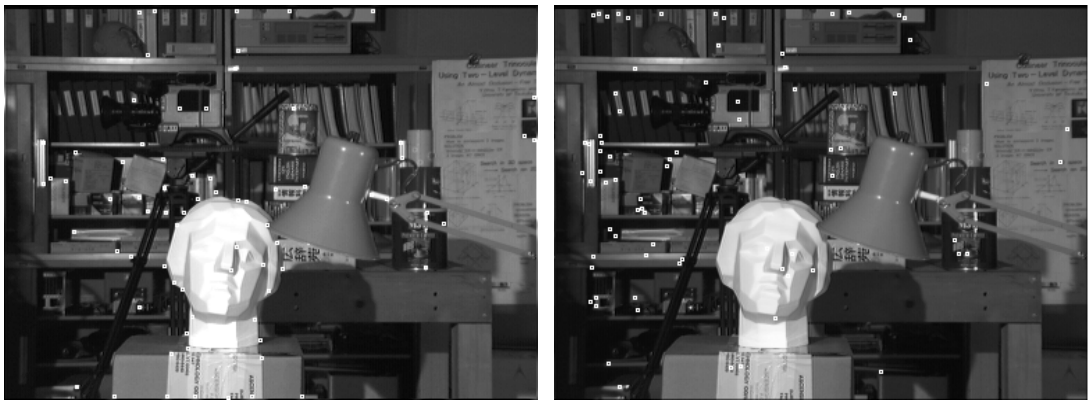
Construction of the SIFT Feature Vector

- Computation of gradient information for each pixel in a 16×16 neighbourhood
- Decreasing weight with distance from center (Gaussian window)
- Accumulation of information of 4×4 sub-patches into a orientation histogram
  (8 directions, accumulated magnitude information)
- 16 histograms with 8 entries each yield a feature vector of size 128
- Normalisation of vector (to obtain a unit vector)



Construction of the SIFT vector from 8×8 neighbourhood orientations. *Author:* D. Lowe (2004).

Structure Tensor vs. SIFT Features



Comparison of interest points found by the structure tensor and the SIFT approach for the left image of the Tsukuba image pair (Middlebury stereo evaluation). **(a) Left:** Structure tensor ($\rho = 3$). **(b) Right:** SIFT-Features ($T = 0.12$).

Feature-Based Methods

- *Idea:* Match interest points to obtain a non-dense but precise displacement field
- For each image interest points and corresponding feature vectors are computed, e.g. the previously discussed SIFT vectors
- Compare all feature vectors from the first image to all feature vectors from the second image (exhaustive search but efficient if number of points is small)
- Difference between two feature vectors $\mathbf{a}_i$ and $\mathbf{b}_j$ of size N defined by a cost function, e.g. SSD:

$$c_{\text{diff}}(\mathbf{a}_i, \mathbf{b}_j) = \sum_{k=1}^{N} ((a_i)_k - (b_j)_k)^2$$

- For each interest point chose the match with minimal cost
- Discard non-unique matches, i.e. matches where the ratio between the costs for the best and the second best match is smaller than a certain threshold T.

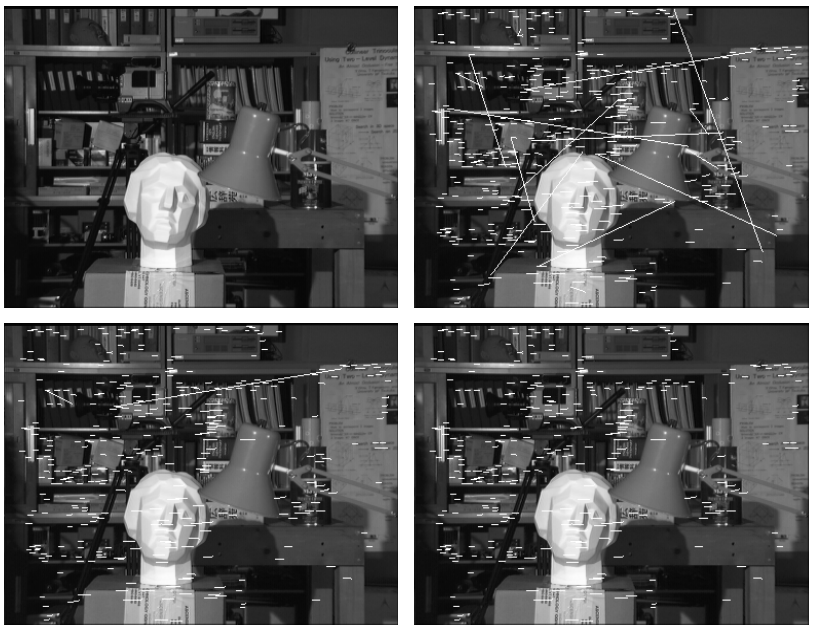
Feature-Based Methods (2)

Advantages and Shortcomings

- Advantages
  - inherit invariant properties from interest points
  - allow the estimation of global motion models with a few parameters (e.g. global translations, rotations or affine motion)
  - allow the estimation of the relation between two camera views (epipolar geometry is described by 7 parameters)
- Drawbacks
  - yield non-dense flow fields which are not useful for many tasks: e.g. stereo reconstruction, medical image registration, ...
  - require post-processing interpolation steps
  - produce outliers that are difficult to handle (statistically robust methods are needed, e.g. RANSAC)

Feature-Based Methods (3)

SIFT Matching Results



SIFT matching results for the Tsukuba stereo pair (Middlebury stereo evaluation). **(a) Upper Left:** Left frame. **(c) Upper Right:** SSD-SIFT ($T = 0.8$). **(d) Lower Left:** SSD-SIFT ($T = 0.65$). **(f) Lower Right:** SSD-SIFT ($T = 0.5$).

Summary (1)

Summary

- Approaches that compare only the grey value of pixels give poor results
- Block matching and correlation-based methods compare neighbourhoods and are thus more reliable (in homogeneous areas the solution is still non-unique)
    - Such methods are robust under noise due to neighbourhood information
    - The NCC even allows to cope with globally varying illumination
    - Although their implementation is simple, they only offer an average performance and a relatively poor efficiency for 2-D search spaces
- Sup-pixel precision can be addressed only a-posteriori by function fitting.
- Occlusions can be detected by a simple forward/backward check.
- Some locations in images are more characteristic than others
- Feature based methods try to match exactly such locations and thus produce sparse but accurate displacement fields

Summary (2)

Literature

- W. Förstner, E. Gülch:
  A fast operator for detection and precise location of distinct points, corners and centres of circular features.
  In *Proc. ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data*, pp. 281–305, 1987.
  (first paper on the structure tensor)
- D. G. Lowe:
  Distinctive image features from scale-invariant keypoints.
  In *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91–110, 2004.
  (introduction to the scale invariant feature transform - SIFT)
- H. G. Musmann, P. Pirsch, H. J Grallert:
  Advances in picture coding.
  In *Proceedings of the IEEE*, Vol. 73, No. 4, pp. 523–548, 1985.
  (normalised cross correlation/block matching for video compression)

Assignment 1

Programming Exercise (Block Matching)

You can download the file copcv09_ex01.tgz from the web page

`http://www.mia.uni-saarland.de/Teaching/copcv09.shtml`

To unpack these data, use `tar xzvf copcv09_ex01.tgz`.

1. Supplement the routine `matching_cost()` in the C programme `blockmatching.c` with the missing code such that it becomes an implementation of the SSD block matching algorithm. Please note that many functions are already available within the routines the accompany the provided code. The front-end is realised with OpenGL and GLUT. In order to compile your programme please use the contained makefile. The compiled programme is then executed by

   ```
   ./frontend <input_image1.pgm> <input_image2.pgm> <zoom_ratio>
   ```

   where the integer parameter `zoom_ratio` is in general set to 1.

2. Use the provided image pairs `tsu1.pgm` and `tsu2.pgm` as well as `yos1.pgm` and `yos2.pgm` and evaluate the performance of the SSD block matching algorithm for different window sizes m. The maximum displacement magnitude for limiting the search space should be chosen $d \leq 7$ for Yosemite (optic flow) and $d \leq 13$ for Tsukuba (stereo).

Assignment 1

Programming Exercise (Block Matching)

3. Supplement the routine `subpixel_refinement()` in the same C programme with the code for parabola fitting from the lecture. Compare your results visually with and without sub-pixel refinement.

4. Finally, supplement the main routine `BLOCK_MATCHING()` in the same C programme with the code for occlusion detection using the forward-backward check. Use the forward flow if the estimation is reliable. If not, i.e. if you have identified an occluded pixel, set the corresponding displacement to zero. Vary the occlusion threshold and take a look at the resulting displacement field. Please note that the programme either allows you to use subpixel refinement or occlusion detection.