Institute of Informatics
Eötvös Loránd University
Budapest, Hungary

Basic Algorithms for Digital Image Analysis: a course

Dmitrij Csetverikov

with help of Attila Lerch, Judit Verestóy, Zoltán Megyesi, Zsolt Jankó

http://visual.ipan.sztaki.hu

Lecture 5: Finding Patterns in Images

- Neighbourhood processing: A summary
  - Adaptive neighbourhood selection
  - Frequency domain: low-, high- and band-pass filters
- Matching and correspondence in computer vision
- Template matching
  - Similarity and dissimilarity measures
  - Interior matching versus contour matching
  - Invariance
  - Speed

Adaptive neighbourhood selection

All filters considered up to now are non-adaptive (position-independent):

- Fixed neighbourhood selection procedure
- Fixed function that calculates output value

Adaptivity: Using local context to improve performance of noise filters

- Goal: Avoid artefacts (undesirable effects)
    - ‘Averaging across edges’ by the mean filter
    - Rounding of corners by the median filter
- Main cause of artefacts: Pixels belonging to different classes (distributions) are mixed by filter.

Basic idea: Try to separate

- object pixels from background pixels
- relevant greyvalues from noise

How can we do this making the filter adaptive?

Components of a noise filter:

- Neighborhood selection: Selecting relevant pixels in a vicinity of central pixel.
  - Until now, we used all pixels of the window.
  - Now, we are going to select certain pixels.
- Computation of the output value based on the selected relevant pixels
  - Until now, we used fixed functions: mean, median, etc.
  - This will not change.

Alternative ways of neighborhood selection in a $n \times n$ window:

- Standard neighborhood: Use all n^2 pixels.
- k-nearest neighbors (kNN): Select those k pixels that are closest in grey value to the central pixel. A possible value of k is

$$k = n \times \left[\frac{n}{2}\right] + (n-1)$$

  Example: For $n = 3$, $k = 5$

- Sigma-nearest neighbors:

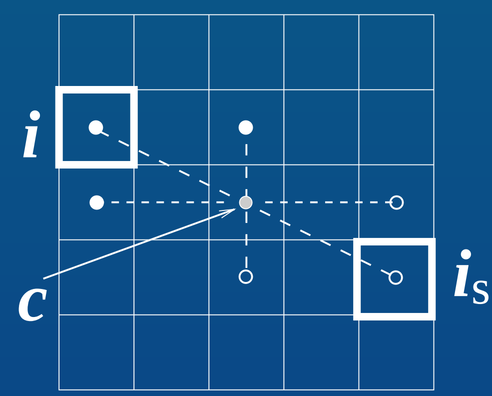
$$\text{select pixels } i : |I(i) - I(c)| < k \cdot \sigma_{ns}$$

  - Usually, $k = 2$.
  - σ_{ns} is the standard deviation of noise
    - estimated in a flat (non-variyng) region of image.

Symmetric nearest neighbors:

$$\text{select } i \text{ if } |I(i) - I(c)| < |I(i_s) - I(c)|$$

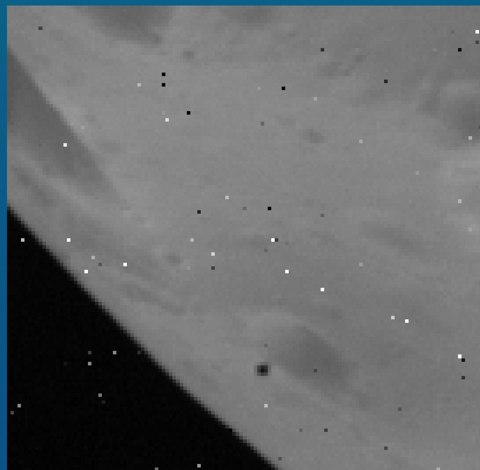
- $I(m)$: intensity of pixel m; c: central pixel; $\{i, i_s\}$: symmetric pixels



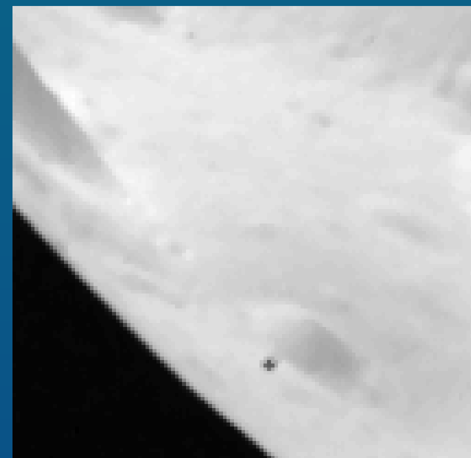
Examples of symmetric pixel pairs in a window.

- Local context: intensity and geometry are taken into account.

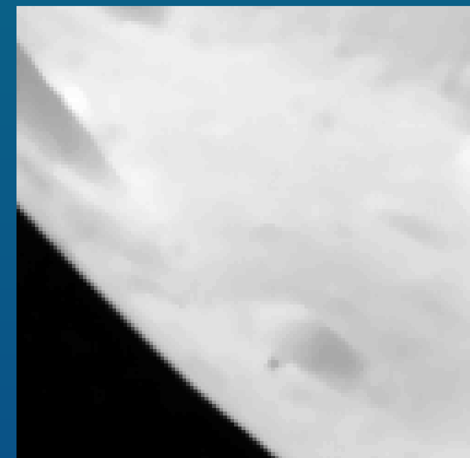
- Useful in case of edges
  - Selects pixels on the same side of an edge.
  - Averaging across edge is avoided.



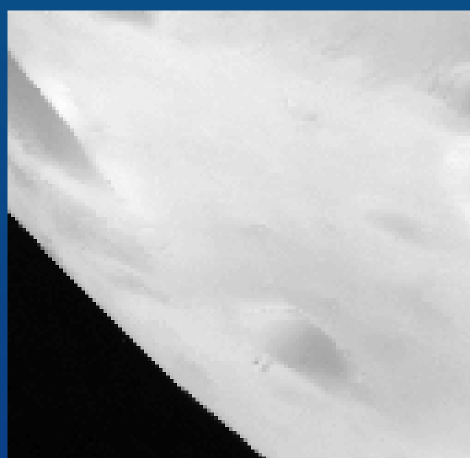
Original image



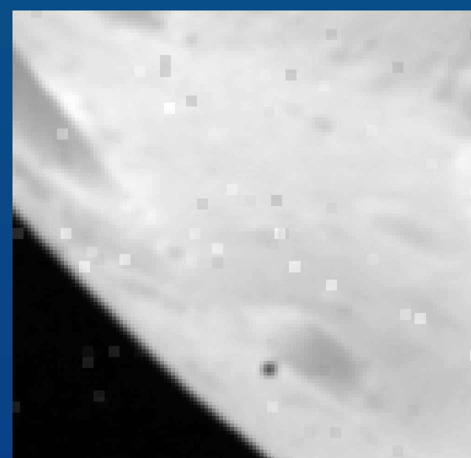
Median filter 3×3



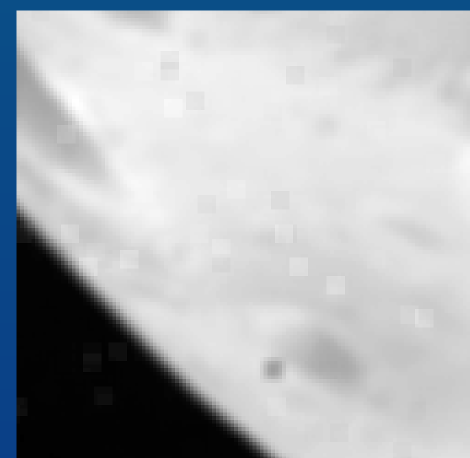
Median filter 5×5



Symmetr. box 5×5



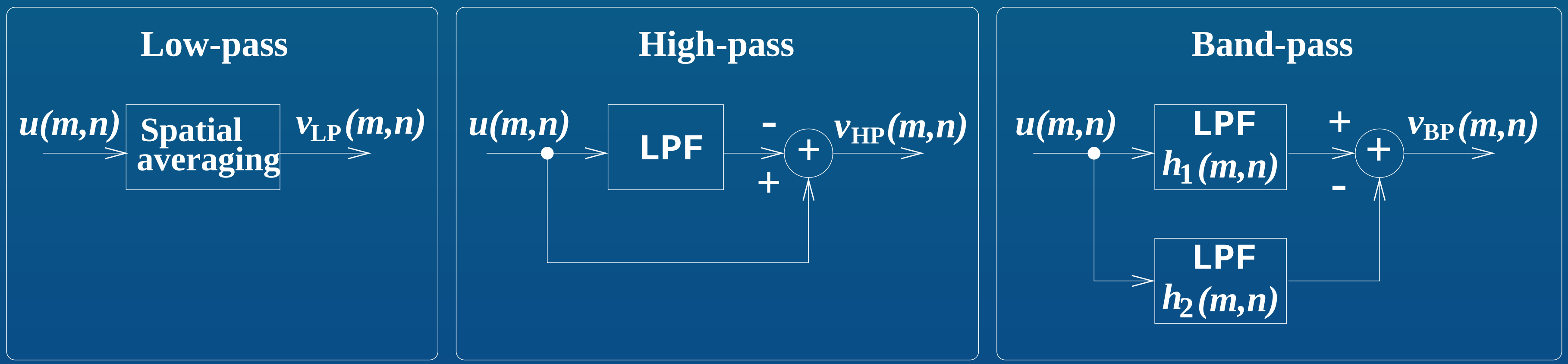
Box filter 3×3



Box filter 5×5

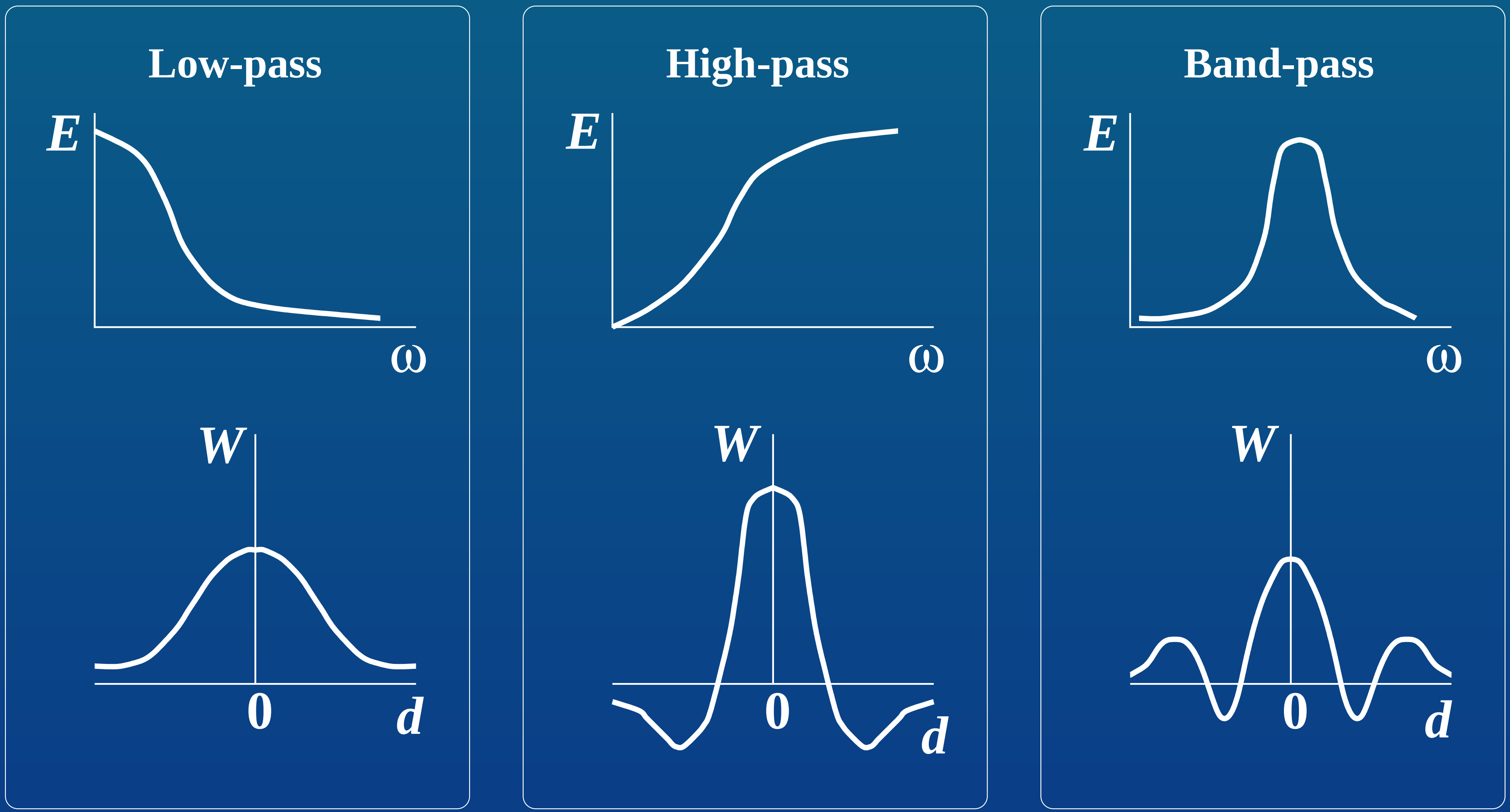
Comparison of the median and the box filters for a greyscale image corrupted by the salt-and-pepper noise. The images are grey-scale normalised.

Frequency domain: low-, high- and band-pass filters



Block diagrams of low-, high- and band-pass filtering.

- Low-pass filter (LPF): Spatial averaging.
- High-pass filter: Weighted difference of the original image and an LPF.
- Band-pass filter: Weighted difference of two low-pass filters, LPF1 and LPF2.



Low-, high- and band-pass filtering:
Typical shapes of filters in frequency domain E and spatial domain W.

- Low-pass
  - Fourier kernel E: Low frequencies (gradual variations) preserved or amplified, high frequencies (edges) suppressed.
  - Spatial kernel W: Non-negative, monotonically decreasing with radius $r = |d|$.
- High-pass
  - Fourier kernel E: High frequencies preserved or amplified, low frequencies suppressed .
  - Spatial kernel W: Positive center (original image), negative wings (averaged image subtracted).
- Band-pass
  - Fourier kernel E: A band of frequencies preserved or amplified, other frequencies suppressed.
  - Spatial kernel W: Weighted difference of two low-pass filters.

Matching and correspondence in computer vision

Image matching: Finding correspondences between two or more images.

Basic tasks of computer vision related to matching:

1. Given images of a scene taken by different sensors, bring them into registration.
   - This is called image registration.
   - Typical example: Medical imaging
     - Images obtained by sensors of different types are called modalities.

2. Given images of a scene taken at different times, find correspondences, displacements, or changes.
   - This is motion analysis.
   - Typical example: Motion tracking.

3. Given images of a scene taken from different positions, find correspondent points to obtain 3D information about the scene.

   - This is stereopsis, or simply stereo.
     - Matching provides disparity: the shift of a point between the two views
     - By triangulation, disparity and baseline (distance between eyes) provide depth: the 3D distance to the point.
   - Generalised stereo is called 3D scene reconstruction from multiple views.

4. Find places in an image or on a contour where it matches a given pattern.

   - Template matching: Pattern specified by template.
   - Feature detection: Feature specified by description.

5. Match two contours for object recognition, measurement, or positioning.

   - This is contour matching.

Key issues of matching:

- Invariance under imaging transformations
  - spatial
  - photometric (intensity)
- Sensitivity to noise and distortions

Considered in this course are:

- Tasks
  - Task 4: Template matching and feature detection
  - Task 5: Contour matching
- Transformations
  - Spatial: 2D shift and rotation
  - Photometric: Shift and scaling of intensity (linear)

Template matching

Compare a subimage (template) $w(r', c')$ with an image $f(r', c')$ for all possible displacements (r, c).
In other words: Match $w(r', c')$ against $f(r + r', c + c')$ for all (r, c).

Measures of dissimilarity between image f and template w in position (r, c):

D1 Sum of Square Differences (SSD):

$$D(r,c) = \sum_{\substack{(r',c') \in W \\ (r+r',c+c') \in F}} \left\{ f(r + r', c + c') - w(r', c') \right\}^2$$

- W: set of pixel positions in template w (template coordinates)
- F: set of pixel positions in image f (image coordinates)

$D(r, c)$ is not invariant under the following transformations

- 2D rotation $\Rightarrow$ Cannot find significantly rotated pattern
- Shift or scaling of intensity $\Rightarrow$ Cannot cope with any varying illumination

D2 Intensity shift-corrected SSD:

$$\delta(r,c) = \sum_{\substack{(r',c') \in W \\ (r+r',c+c') \in F}} \left\{ \left[f(r+r', c+c') - \overline{f}(r,c) \right] - \left[w(r',c') - \overline{w} \right] \right\}^2$$

- $\overline{f}(r,c)$: Average value of image in region covered by template
  - computed in each position (r,c)

- $\overline{w}$: Average value of template
  - computed only once

$\delta(r,c)$ is a bit more sophisticated measure used to compensate for intensity shift due to uneven illumination.

- Handles changes in average level of signal
- Cannot handle changes in amplitude of signal

Measures of similarity between image f and template w in position (r, c):

S1 Unnormalised cross-correlation (CC) of image f with template w:

$$C_{un}(r, c) = \sum_{\substack{(r',c') \in W \\ (r+r',c+c') \in F}} f(r + r', c + c') \cdot w(r', c')$$

- We have already studied the properties of cross-correlation and convolution.
- $C_{un}(r, c)$ is formally the same as filtering image f with mask w.
  $\Rightarrow$ Our knowledge of filters is applicable, including normalisation, separability, fast implementation, etc.
- $C_{un}(r, c)$ is not invariant under intensity shift and scaling. When $w > 0$ and f is large, $C_{un}(r, c)$ is large, independently from (dis)similarity between w and f.
  $\Rightarrow$ To compensate for this, a normalised version is used.

S2 Normalised cross-correlation (NCC), or correlation coefficient:

$$C_{nr}(r,c) = \frac{1}{\sqrt{S_f(r,c)\cdot S_w}} \sum \Big[f(r+r',c+c') - \overline{f}(r,c) \Big] \cdot \Big[w(r',c') - \overline{w} \Big]$$

where

$$S_f(r,c) = \sum \Big[f(r+r',c+c') - \overline{f}(r,c) \Big]^2, \qquad S_w = \sum \Big[w(r',c') - \overline{w} \Big]^2$$

and for simplicity

$$\sum \text{ denotes } \sum_{\substack{(r',c')\in W \\ (r+r',c+c')\in F}}$$

- $S_f(r,c)$ is computed in each position (r,c), S_W is computed in only once.
- $C_{nr}(r,c)$ is invariant to any linear intensity transformation.
- If the average values are not subtracted, $C_{nr}(r,c)$ is only intensity scale-invariant (scale-corrected).

S3 Modified normalised cross-correlation (MNCC):

$$C_{mnr}(r,c) = \frac{1}{S_f(r,c) + S_w} \sum \Big[f(r+r', c+c') - \overline{f}(r,c) \Big] \cdot \Big[w(r',c') - \overline{w} \Big]$$

where as before

$$S_f(r,c) = \sum \Big[f(r+r', c+c') - \overline{f}(r,c) \Big]^2, \qquad S_w = \sum \Big[w(r',c') - \overline{w} \Big]^2$$

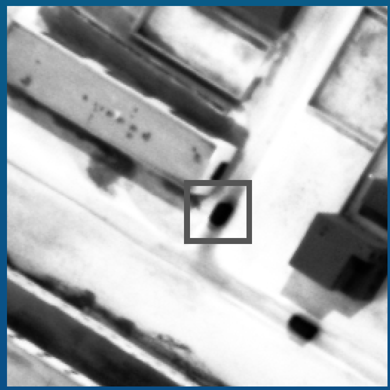
- C_{mnr} is another normalisation:

C_{nr}	is divided by	$\sqrt{S_f(r,c) \cdot S_w}$
C_{mnr}	is divided by	$S_f(r,c) + S_w$

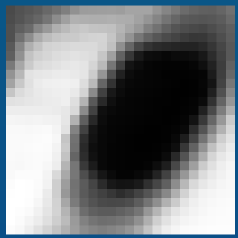
- C_{mnr} is used instead of the standard C_{nr} to avoid the numerically unstable division by a small number when $S_f(r,c)$ is small. (Small image variation.)

- Formally, C_{mnr} is only shift-corrected. In practice, C_{mnr} is reasonably insensitive to intensity scaling as well, since S_w is constant and $S_f(r,c) + S_w$ is roughly proportional to $S_f(r,c)$.

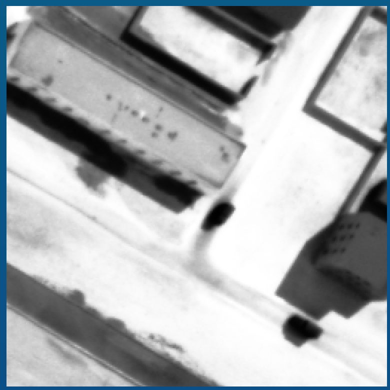
Template matching: Varying r and c, search for locations of high similarity $C_{un}(r,c)$, $C_{nr}(r,c)$, $C_{mnr}(r,c)$, or low dissimilarity $D(r,c)$, $\delta(r,c)$.



Left image

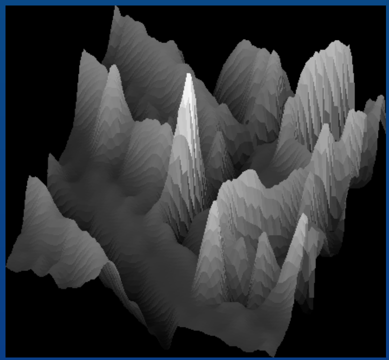


Template, zoomed



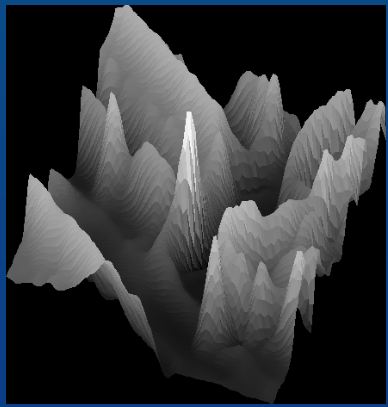
Right image

NCC image



NCC surface

SDD image



SDD surface

Examples of matching in stereo pair. Pattern from left image is searched in right image. NCC is Normalised Cross-Correlation, SSD is Sum of Square Differences.