Local Feature Extraction and Description for
Wide-Baseline Matching, Object Recognition and Image Retrieval Methods, Stitching and more ...

Jiří Matas and Ondra Chum
Center for Machine Perception, Czech Technical University Prague

Includes slides by:
- Darya Frolova, Denis Simakov, The Weizmann Institute of Science
- Martin Urban, Stepan Obdrzalek, Ondra Chum, Center for Machine Perception Prague
- Matthew Brown, David Lowe, University of British Columbia
Outline

- Local features: introduction, terminology
- Motivation: generalisation of local stereo to wide-baseline stereo
- Examples: retrieval, panorama, recognition
- Local invariant features:
  - Harris
  - Scale invariant: SIFT, MSER, LAF
  - FAST, BRIEF-multi-scale FAST with orientation, ORB
- Comparison of properties
Local Features

- Methods based on “Local Features” are the state-of-the-art for number of computer vision problems (often those, that require local correspondences).

- E.g.: Wide-baseline stereo, object recognition and image retrieval.

- Terminology is a mess:
  Local Feature = Interest “Point” = The “Patch” =
  = Feature “Point”
  = Distinguished Region
  = (Transformation) Covariant Region
Motivation: Generalization of Local Stereo to WBS

1. Local Feature (Region) = a rectangular “window”
   - robust to occlusion, translation invariant
   - windows matched by correlation, assuming small displacement
   - successful in stereo matching

2. Local Feature (Region) = a circle around an “interest point”
   - robust to occlusion, translation and rotation invariant
   - matching based on correlation or rotation invariants (*note that the set of circles of a fixed radius is closed under translation and rotation*).
   - successful in tracking and stereo matching

Hard Impossible for a Local feature based method?
Motivation: Generalization of Local Stereo to WBS

3. Widening of baseline or zooming in/out
   • local deformation is well modelled by affine or similarity transformations
   • how can the “local feature” concept be generalised? *The set of ellipses is closed under affine tr., but it’s too big to be tested*
   • window scanning approach becomes computationally difficult.
Local Features in Action (1): Image Retrieval
Local Features in Action (1): Image Retrieval
Local Features in Action (1): Image Retrieval
Local Features in Action (1): Image Retrieval
Local Features in Action (1): Image Retrieval
Local Features in Action (1): Image Retrieval
Local Features in Action (2): Building a Panorama

Local Features in Action (2): Building a Panorama

- We need to match (align) images = find (dense) correspondence
- (technically, this can be done only if both images taken from the same viewpoint)
Local Features in Action (2): Building a Panorama

Possible approach:

1. Detect features in both images
2. Find corresponding pairs
3. Estimate transformations (Geometry and Photometry)
4. Put all images into one frame, blend.
Local Features in Action (2): Building a Panorama

Problem 1:
- Detect the \textit{same} feature \textit{independently} in both images*
- Note that the set of “features” is rather sparse

\begin{itemize}
  \item \hspace{2cm} no chance to match!
\end{itemize}

A \textit{repeatable} detector needed.

* Other methods exist that do not need independency
Problem 2:

• how to correctly recognize the corresponding features?

Solution:

1. Find a discriminative and stable descriptor
2. Solve the matching problem
Local Features in Action (2): Building a Panorama

- Detect feature points in both images
- Find corresponding pairs
- Use these pairs to align images

Any alternatives?
Local Features in Action (4): “Recognition”

(as a Sequence of Wide-Baseline Matching Problems)

Properties: robust to occlusion, clutter, handles pose change, illumination but becomes unrealistic even for moderate number of objects.

Recognition requires indexing

(as a Sequence of Wide-Baseline Matching Problems)
Local Features in Action (3): Object Retrieval

- **Visual Words**
  - $\text{word}_1$, $\text{word}_2$, $\text{word}_8$, ...
  - $\text{word}_{948534}$, $\text{word}_{998125}$

Image representation:

- **Set of words**
- **Bag of words**
- **Visual words**
Local Invariant Features
“Local Features” are **regions**, i.e. in principle arbitrary sets of pixels (not necessarily contiguous) with

- High **repeatability**, (invariance in theory) under
  - Illumination changes
  - Changes of viewpoint ⇒ geometric transformations
  i.e. are **distinguishable** in an image regardless of viewpoint/illumination ⇒ are **distinguished regions**

- Are **robust to occlusion** ⇒ must be **local**

- Must have discriminative neighborhood ⇒ they are “**features**”

Methods based on local features/distinguished regions (DRs) formulate computer vision problems as matching of some representation derived from DR (as opposed to matching of images)
Two core ideas (in “modern terminology”):

1. To be a distinguished region, a region must be at least distinguishable from all its neighbours.

2. Approximation of Property 1. can be tested very efficiently, without explicitly testing.

Note: both properties were proposed before Harris paper, (1) by Moravec, (1)+(2) by Foerstner.
Harris Detector: Basic Idea

“flat” region: no change in all directions

“edge”: no change along the edge direction

“corner”: significant change in all directions

• We should easily recognize the point by looking through a small window

• Shifting a window in any direction should give a large change
Harris Detector: Basic Idea

The Harris detector is a method for detecting corners in an image. It works by computing a corner response function for each pixel in the image. The response function is defined as:

\[ R(x, y) = \sum_{m,n} \left( I_{x+m,y+n}^2 I_{x,y} - \alpha \left( I_{x+m,y+n} I_{x,y} + I_{x,y} I_{x+m,y+n} \right) \right) \]

where \( I_{x,y} \) is the intensity of the image at location \( (x,y) \), and \( \alpha \) is a parameter that balances the responses to changes in brightness and contrast. The detector then identifies corners as locations where the response function is maximized.
Tests how similar is the image function $I(x_0, y_0)$ at point $(x_0, y_0)$ to itself when shifted by $(u, v)$:

- given by autocorrelation function

$$E(x_0, y_0; u, v) = \sum_{(x,y)\in W(x_0,y_0)} w(x,y)(I(x,y) - I(x+u, y+v))^2$$

- $W(x_0, y_0)$ is a window centered at point $(x_0, y_0)$
- $w(x, y)$ can be constant or (better) Gaussian
Approximate intensity function in shifted position by the first-order Taylor expansion:

\[ I(x + u, y + v) \approx I(x, y) + [I_x(x, y), I_y(x, y)] \begin{bmatrix} u \\ v \end{bmatrix} \]

where \( I_x, I_y \) are partial derivatives of \( I(x, y) \).

\[
E(x_0, y_0; u, v) \approx \sum_{(x, y) \in W(x_0,y_0)} w(x,y) \begin{bmatrix} u \\ v \end{bmatrix}^2\begin{bmatrix} I_x(x_0,y_0) \\ I_x(x_0,y_0)I_y(x_0,y_0) \\ I_y(x_0,y_0) \\ I_y(x_0,y_0) \end{bmatrix} \]

\[
= [u, v] \sum_{W} w(x,y) \begin{bmatrix} I_x(x_0,y_0)^2 & I_x(x_0,y_0)I_y(x_0,y_0) \\ I_x(x_0,y_0)I_y(x_0,y_0) & I_y(x_0,y_0)^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \]
Harris Detector: Mathematics

\[ E(x_0, y_0; u, v) \approx [u, v] M(x_0, y_0) \begin{bmatrix} u \\ v \end{bmatrix} \]

Intensity change in shifting window: eigenvalue analysis of \( M \)

- \( \lambda_1, \lambda_2 \) – eigenvalues of \( M \)
- \( M \) symmetric, positive definite

Ellipse:

\[ E(x_0, y_0; u, v) = \text{const} \]

Direction of the fastest change: \((\lambda_{\text{max}})^{-1/2}\)

Direction of the slowest change: \((\lambda_{\text{min}})^{-1/2}\)
Classification of image points using eigenvalues of $M$:

- "Edge": $\lambda_2 >> \lambda_1$
- "Corner": $\lambda_1$ and $\lambda_2$ are large, $\lambda_1 \sim \lambda_2$; $E$ increases in all directions
- "Flat" region: $\lambda_1$ and $\lambda_2$ are small; $E$ is almost constant in all directions

$\lambda_1$ and $\lambda_2$ are the eigenvalues of the matrix $M$. The classification is based on the relationship between these eigenvalues.
Measure of corner response ("cornerness"): 

\[ R = \det M - k(\text{trace } M) \]

\[ M = \begin{bmatrix} A & B \\ B & C \end{bmatrix} \]

\[ \det M = \lambda_1 \lambda_2 = AC - B^2 \]

\[ \text{trace } M = \lambda_1 + \lambda_2 = A + C \]

\[ k \ldots \text{empirical constant, } k \in (0.04, 0.06) \]

Find corner points as \textbf{local maxima} of corner response \( R \):

- points greater than its neighbours in given neighbourhood (3 × 3, or 5 × 5)
Harris Detector: Mathematics

- $R$ depends only on eigenvalues of $M$
- $R$ is large for a corner
- $R$ is negative with large magnitude for an edge
- $|R|$ is small for a flat region

![Diagram showing regions of $R$ for different eigenvalues $\lambda_1$ and $\lambda_2$.]

- "Corner" region with $R > 0$
- "Edge" region with $R < 0$ and $|R|$ large
- "Flat" region with $|R|$ small
Harris Detector

- The Algorithm:
  - Compute partial derivatives $I_x, I_y$
  - Compute: $A = \sum W I_x^2$, $B = \sum W I_x I_y$, $C = \sum W I_y^2$
  - Compute corner response $R$
  - Find local maxima in $R$

- Parameters:
  - Threshold on $R$
  - Scale of the derivative operator (standard setting: very small, just enough to filter anisotropy of the image grid)
  - Size of window $W$ (“integration scale”)
  - Non-maximum suppression algorithm
Harris Detector: Workflow
Harris Detector: Workflow

Compute corner response $R$
Harris Detector: Workflow

Find points with large corner response: $R > \text{threshold}$
Harris Detector: Workflow

Take only the points of local maxima of $R$
Harris Detector: Workflow
Harris Detector: Properties

- Rotation invariance

Ellipse rotates but its shape (i.e. eigenvalues) remains the same

*Corner response $R$ is invariant to image rotation*
Rotation Invariance of Harris Detector

Harris Detector: Intensity change

- Partial invariance to additive and multiplicative intensity changes
  - Only derivatives are used =>
    - invariance to intensity shift $I \rightarrow I + b$
  - Intensity scale: $I \rightarrow aI$
Harris Detector: Scale Change

- Not invariant to *image scale*!

All points will be classified as *edges*.
Harris Detector: Scale Change

- Quality of Harris detector for different scale changes

Repeatability rate:

\[
\text{# correspondences} \quad \text{# possible correspondences}
\]

FAST Feature Detector

- Considers a circle of 16 pixels around the corner candidate $p$
- $\geq 12$ contiguous pixels brighter/darker than $I_p \pm t$, $t$... threshold
- Rapid rejection by testing 1,9,5 then 13
  - Only if at least 3 of those are brighter/darker than $I_p \pm t$, the full segment test is applied
FAST: Weaknesses

- Corners are clustered together:
  - Use non-maximal suppression:

\[ V = \max \left( \sum_{q \in S_b} |I_q - I_p| - t, \sum_{q \in S_d} |I_p - I_q| - t \right) \]

where \( S_b = \{ q | I_q \geq I_p + t \} \), \( S_d = \{ q | I_q \leq I_p - t \} \)

- High speed test does not generalize well for \( n < 12 \)
- Choice of high speed test is not optimal
- Knowledge from the first 4 tests is discarded
- Multiple features are detected adjacent to one another
FAST: running times

<table>
<thead>
<tr>
<th>Detector</th>
<th>Opteron 2.6GHz</th>
<th>Pentium III 850MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ms</td>
<td>%</td>
</tr>
<tr>
<td>Fast $n = 9$ (non-max suppression)</td>
<td>1.33</td>
<td>6.65</td>
</tr>
<tr>
<td>Fast $n = 9$ (raw)</td>
<td>1.08</td>
<td>5.40</td>
</tr>
<tr>
<td>Fast $n = 12$ (non-max suppression)</td>
<td>1.34</td>
<td>6.70</td>
</tr>
<tr>
<td>Fast $n = 12$ (raw)</td>
<td>1.17</td>
<td>5.85</td>
</tr>
<tr>
<td>Original FAST $n = 12$ (non-max suppression)</td>
<td>1.59</td>
<td>7.95</td>
</tr>
<tr>
<td>Original FAST $n = 12$ (raw)</td>
<td>1.49</td>
<td>7.45</td>
</tr>
<tr>
<td>Harris</td>
<td>24.0</td>
<td>120</td>
</tr>
<tr>
<td>DoG</td>
<td>60.1</td>
<td>301</td>
</tr>
<tr>
<td>SUSAN</td>
<td>7.58</td>
<td>37.9</td>
</tr>
</tbody>
</table>

Table 1. Timing results for a selection of feature detectors run on fields ($768 \times 288$) of a PAL video sequence in milliseconds, and as a percentage of the processing budget per frame. Note that since PAL and NTSC, DV and 30Hz VGA (common for webcams) have approximately the same pixel rate, the percentages are widely applicable. Approximately 500 features per field are detected.
Scale Invariant Detection

- Consider regions (e.g. circles) of different sizes around a point
- Regions of corresponding sizes will look the same in both images
The problem: how do we choose corresponding circles *independently* in each image?
Solution:

- Design a function on the region (circle), which is “scale covariant” (the same for corresponding regions, even if they are at different scales).

  - For a point in one image, we can consider it as a function of region size (circle radius).
Common approach:

- Take a local maximum of some function
- *Observation*: region size, for which the maximum is achieved, should be *invariant* to image scale.

Important: this scale invariant region size is found in each image *independently*!
A “good” function for scale detection: has one stable sharp peak

- For usual images: a good function would be a one which responds to contrast (sharp local intensity change)
Scale Invariant Detection

- Functions for determining scale

\[ f = \text{Kernel} \ast \text{Image} \]

Kernels:

\[
L = \sigma^2 \left( G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma) \right)
\]

(Laplacian)

\[
DoG = G(x, y, k\sigma) - G(x, y, \sigma)
\]

(Difference of Gaussians)

where Gaussian

\[
G(x, y, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2+y^2}{2\sigma^2}}
\]
Scale Invariant Detectors

Harris-Laplacian\(^1\)

*Find local maximum of:*

- Harris corner detector in space (image coordinates)
- Laplacian in scale

Laplacian-Laplacian = “SIFT” (Lowe)\(^2\)

*Find local maximum of:*

- Difference of Gaussians in space and scale

Other options: Hessian, ...

Harris does not work well for scale selection

---


Scale Invariant Detectors

- Experimental evaluation of detectors w.r.t. scale change

Repeatability rate:
- \# correspondences
- \# possible correspondences

Affine Invariant Detection

• Above we considered:
  Similarity transform (rotation + uniform scale)

• Now we go on to:
  Affine transform (rotation + non-uniform scale)
Affine Invariant Detection

- Take a local intensity extremum as initial point
- Go along every ray starting from this point and stop when extremum of function $f$ is reached

$$f(t) = \frac{|I(t) - I_0|}{\frac{1}{t} \int_0^t |I(t) - I_0| \, dt}$$

- We will obtain approximately corresponding regions

Remark: we search for scale in every direction

The regions found may not exactly correspond, so we approximate them with ellipses.

- Geometric Moments:

\[ m_{pq} = \int \int x^p y^q f(x, y) dx dy \]

Fact: moments \( m_{pq} \) uniquely determine the function \( f \).

Taking \( f \) to be the characteristic function of a region (1 inside, 0 outside), moments of orders up to 2 allow to approximate the region by an ellipse.

This ellipse will have the same moments of orders up to 2 as the original region.
Affine Invariant Detection

- **Covariance matrix** of region points defines an ellipse:

\[ [x_2, y_2] = A[x_1, y_1] \]

\[ [x_1, y_1]^T \sum_1^{-1} [x_1, y_1] = 1 \]

\[ \sum_1 = \langle [x_1, y_1][x_1, y_1]^T \rangle_{\text{region}_1} \]

\[ [x_2, y_2]^T \sum_2^{-1} [x_2, y_2] = 1 \]

\[ \sum_2 = \langle [x_2, y_2][x_2, y_2]^T \rangle_{\text{region}_2} \]

\[ \sum_2 = A \sum_1 A^T \]

Ellipses, computed for corresponding regions, also correspond!
Algorithm summary (detection of affine invariant region):

- Start from a local intensity extremum point
- Go in every direction until the point of extremum of some function $f$
- Curve connecting the points is the region boundary
- Compute geometric moments of orders up to 2 for this region
- Replace the region with ellipse

1. Detect initial region with Harris or Hessian detector and select the scale

2. Estimate the shape with the second moment matrix

3. Normalize the affine region to the circular one

4. Go to step 2 if the eigenvalues of the second moment matrix for the new point are not equal

\[ [x_1, y_1] \rightarrow M_1^{-1/2} [x'_1, y'_1] \]

\[ [x'_1, y'_1] \rightarrow R [x'_2, y'_2] \]

\[ [x_2, y_2] \rightarrow M_2^{-1/2} [x'_2, y'_2] \]
The Maximally Stable Extremal Regions

- Consecutive image thresholding by all thresholds
- Maintain list of Connected Components
- Regions = Connected Components with stable area (or some other property) over multiple thresholds selected

The Maximally Stable Extremal Regions

video
MSER Stability

Properties:
- Covariant with continuous deformations of images
- Invariant to affine transformation of pixel intensities
- Enumerated in $O(n \log \log n)$, real-time computation

MSER regions (in green). The regions ‘follow’ the object (video1, video2).

Matas, Chum, Urban, Pajdla: “Robust wide baseline stereo from maximally stable extremal regions”. BMVC2002
Descriptors of Local Invariant Features
Descriptors Invariant to Rotation

- **Image moments in polar coordinates**

  \[ m_{kl} = \int \int r^k e^{-i\theta l} I(r, \theta) dr d\theta \]

  Rotation in polar coordinates is translation of the angle:
  \[ \theta \rightarrow \theta + \theta_0 \]

  This transformation changes only the phase of the moments, but not their magnitude

  Rotation invariant descriptor consists of magnitudes of moments:

  \[ |m_{kl}| \]

  Matching is done by comparing vectors \[ [|m_{kl}|]_{k,l} \]

Descriptors Invariant to Rotation

• Find local orientation

Dominant direction of gradient

• Compute image derivatives relative to this orientation

---

Descriptors Invariant to Scale

- Use the scale determined by detector to compute descriptor in a normalized frame

For example:
  - moments integrated over an adapted window
  - derivatives adapted to scale: $s_l^x$
Affine Invariant Descriptors

- Affine invariant color moments

$$m_{pq}^{abc} = \int_{\text{region}} x^p y^q R^a(x, y) G^b(x, y) B^c(x, y) dx dy$$

- Different combinations of these moments are fully affine invariant

- Also invariant to affine transformation of intensity $I \rightarrow aI + b$

F. Mindru et al. “Recognizing Color Patterns Irrespective of Viewpoint and Illumination”. CVPR99
Affine Invariant Descriptors

• Find affine normalized frame

\[ \Sigma_1 = \langle [x_1, y_1], [x_1, y_1]^T \rangle \]

\[ \Sigma_2 = \langle [x_2, y_2], [x_2, y_2]^T \rangle \]

\[ \Sigma_1 = A_1^T A_1 \]

\[ \Sigma_2 = A_2^T A_2 \]

rotation

• Compute rotational invariant descriptor in this normalized frame

Local Affine Frames

Step 1: Find MSERs (maximally stable extremal regions)
Step 2: Construct **Local Affine Frames (LAFs)** (local coordinate frames)
Step 3: **Geometrically normalize** some measurement region (MR) expressed in LAF coordinates

All measurements in the normalized frame are Invariants!

Stability of LAFs: **concavity, curvature max 1, curvature max 2**

Obdržálek and Matas: “Object recognition using local affine frames on distinguished regions”. BMVC02
Obdržálek and Matas: “Sub-linear Indexing for Large Scale Object Recognition”, BMVC 2005
Affine-Covariant Constructions: Taxonomy

Derived from *region outer boundary*

- Region area (1 constraint)
- Center of gravity (2 constraints)
- Matrix of second moments (symmetric 2x2 matrix: 3 constraints)
  - Points of extremal distance to the center of gravity (2 constraints)
  - Points of extremal curvature (2 constraints)

\[
|\Omega| = \int_{\Omega} 1 \, d\Omega \\
\mu = \frac{1}{|\Omega|} \int_{\Omega} x \, d\Omega \\
\Sigma = \frac{1}{|\Omega|} \int_{\Omega} (x - \mu)(x - \mu)^T \, d\Omega
\]

Shape normalisation by the covariance matrix. (a) a detected region, (b) the region shape-normalised to have unit covariance matrix, (c) local curvatures of the normalised shape, (d) distances to the center of gravity.
Derived from region outer boundary (continued)
- Concavities (4 constraints for 2 tangent points)
  - Farthest point on region contour/concavity (2 constraints)

Example region concavities. (a) A detected non-convex region with indicated concavities and their covariance matrices (b) One of the concavities - the bitangent line and region and concavity farthest points.
Derived from *image intensities* in a region (or its neighbourhood)

- From orientation of gradients
  - peaks of gradient orientation histograms [Low04] (1 constraint)
- Direction of dominant texture periodicity (1 constraint)
- Extrema or centers of gravity of R, G, B components, or of any scalar function of the RGB values (2 constraints)

- many other

Affine-Covariant Constructions: Taxonomy

- Derived from *topology* of regions
  - mutual configuration of regions (combined constraints)
    - nested regions
    - incident regions
    - neighbouring regions

- Region holes and concavities can be considered as regions of their own
  - all aforementioned constructions recursively applicable

- Convex hull of a region without loosing affine invariance
Combinations of constructions used to form the local affine frames

- center of gravity + covariance matrix + curvature minima
Constructions of Local Affine Frames

- Combinations of constructions used to form the local affine frames
  - center of gravity + covariance matrix + curvature maxima
Constructions of Local Affine Frames

- Combinations of constructions used to form the local affine frames
  - center of gravity + tangent points of a concavity
Constructions of Local Affine Frames

- Combinations of constructions used to form the local affine frames
  - tangent points + farthest point of the region
Constructions of Local Affine Frames

- Combinations of constructions used to form the local affine frames
  - tangent points + farthest point of the concavity
Constructions of Local Affine Frames

- Combinations of constructions used to form the local affine frames
  - tangent points + center of gravity of the concavity
Constructions of Local Affine Frames

- Combinations of constructions used to form the local affine frames
  - center of gravity + covariance matrix + center of gravity of a concavity
Constructions of Local Affine Frames

- Combinations of constructions used to form the local affine frames
  - center of gravity + covariance matrix + direction of a bitangent
Combinations of constructions used to form the local affine frames

- center of gravity of a concavity + covariance matrix of the concavity + the direction of the bitangent
Constructions of Local Affine Frames

- Combinations of constructions used to form the local affine frames
  - center of gravity + covariance matrix + the direction of a linear segment of the contour
Constructions of Local Affine Frames

- Combinations of constructions used to form the local affine frames
  - center of gravity + covariance matrix + the direction to an inflection point
Combinations of constructions used to form the local affine frames

- center of gravity + covariance matrix + the direction given by the third-order moments of the region
Derived from *region outer boundary* (continued)

- Points of curvature inflection (2 constraints)
  - curvature changes from convex to concave or vice-versa
- Straight line segments (1 stable constraint for direction, or 4 for the end-points)
- Higher than 2\textsuperscript{nd} order moments
  - a complex number formed from 3\textsuperscript{rd} order moments

whose phase angle

\[ c = \mu_{x^3} + \mu_{x^2y} + i(\mu_{x^2y} + \mu_{y^3}) \]

changes covariantly with the region’s rotation [Hei04] (1 constraint)

\[ \alpha = \tan^{-1}\left(\frac{\mu_{x^2y} + \mu_{y^3}}{\mu_{x^3} + \mu_{xy^2}}\right) \]

 Canonical Frames are an old idea ...

Rothwell, Zisserman, Forsyth, Mundy:
*Canonical Frames for Planar Object Recognition*, 1992

- Multiple reference frames
- Grouping of distinguished points is based on ordering on the segment
Fig. 1. (a) Construction of the four points necessary to define the canonical frame for a concavity. The first two points \((A, D)\) are points of bitangency that mark the entrance to the concavity. Two further distinguished points, \((B, C)\), are obtained from rays cast from the bitangent contact points and tangent to the curve segment within the concavity. These four points are used to map the curve to the canonical frame. (b) Curve in canonical frame. A projection is constructed that transforms the four points in (a) to the corner of the unit square. The same projection transforms the curve into this frame.
Impressing the Reader: Robustness to occlusion, clutter, multiple objects

Fig. 6. (a) Spanner almost entirely occluded by keys. The keys are not the li in this scene. (b) Detected concavities, highlighted in white, which are used (c) The spanner which is the only model in the scene contained in the library the end slot concavity. The projected outline used for verification is highli

Fig. 7. (a) Image of various planar objects. (b) Concavities, highlighted in white, which are used to compute indexes (c) The pliers which are the only model in the scene contained in the library, is recognised and verified by projecting the edgels from an acquisition image, and checking overlap with edgels in this image.
Common Structure of “Local Feature” Algorithms

1. Detect affine- (or similarity-) covariant regions (=distinguished regions) = local features
   Yields regions (connected set of pixels) that are detectable with high repeatability over a large range of conditions.

2. Description: Invariants or Representation in Canonical Frames
   Representation of local appearance in a Measurement Region (MR). Size of MR has to be chosen as a compromise between discriminability vs. robustness to detector imprecision and image noise.

3. Indexing
   For fast (sub-linear) retrieval of potential matches

4. Verification of local matches

5. Verification of global geometric arrangement
   Confirms or rejects a candidate match
Local features meet Invariants: Schmid & Mohr, 1997

- Multi-scale differential gray value invariants computed at Harris points
- Scale and rotation invariant
- Feature vectors compared by Mahalanobis distance
- Similarity-based geometric constraint to reject mismatches
- Canonical Frame not used.

Detector:
- Scale-space peaks of Difference-of-Gaussians filter response (Lindeberg 1995)
- **Similarity frame** from modes of gradient histogram

SIFT Descriptor:
- Local histograms of gradient orientation
- Allows for small misalignments
  => robust to non-similarity transforms

Indexing:
- kD-tree structure

Matching:
- test on euclidean distance of 1\textsuperscript{st} and 2\textsuperscript{nd} match

Verification:
- Hough transform based clustering of correspondences with similar transformations

Fast, efficient implementation, **real-time recognition**

Scale space processed one octave at a time
Sub-pixel/ Sub-level Keypoint Localization

- Detect maxima and minima of difference-of-Gaussian in scale space
- Fit a quadratic to surrounding values for sub-pixel and sub-scale interpolation (Brown & Lowe, 2002)
- Taylor expansion around point:

\[ D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \]

- Offsets (finite differences for derivatives):

\[ \hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \]
Select canonical orientation (s)

- Compute a histogram of local gradient directions computed at the selected scale
- Assign canonical orientation(s) at peak(s) of smoothed histogram
- \((x, y, \text{scale}) + \text{orientation}\) defines a local similarity frame; equivalent to detecting 2 distinguished points

**Note:** if orientation of the object (image) is known, it may replace this construction
SIFT Descriptor

- A 4x4 histogram lattice of orientation histograms
- Orientations quantized (with interpolation) into 8 bins
- Each bin contains a weighted sum of the norms of the image gradients around its center, with complex normalization
SIFT Descriptor

SIFT descriptor can be viewed as a 3–D histogram in which two dimensions correspond to image spatial dimensions and the additional dimension to the image gradient direction (normally discretised into 8 bins).
SIFT – Scale Invariant Feature Transform\textsuperscript{1}

- Empirically found\textsuperscript{2} to show very good performance, invariant to *image rotation*, *scale*, *intensity change*, and to moderate *affine* transformations

Scale = 2.5
Rotation = 45\textdegree

\textsuperscript{1} D.Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. IJCV 2004
SIFT invariances

- Based on gradient orientations, which are robust to illumination changes
- Spatial binning gives tolerance to small shifts in location and scale, affine change.
- Explicit orientation normalization
- Photometric normalization by making all vectors unit norm
- Orientation histogram gives robustness to small local deformations
SIFT Descriptor

By far the most commonly used distinguished region descriptor:
• fast
• compact
• works for a broad class of scenes
• source code available

large number of ad hoc parameters ⇒ Enormous follow up literature on both “improvements” and improvements [HoG, Daisy, Cogain]
• GLOH, HoG: different grid, not 4x4, not necessarily a square
• Daisy: many parameters optimized
Learning Local Image Descriptors

The best result of all was obtained by combining steerable filters with the polar plan of S4 to give T3h-S4-25. At just under a 2% error rate, this is one third of the error rate produced by SIFT at 95% correct matches. The ROC curve for this descriptor is plotted on Figure 11. However the dimensionality is quite high at 400.
DAISY local image descriptor

I. Histograms at every pixel location are computed

\[ h_\Sigma(u, v) = \left[ G_1^\Sigma(u, v), \ldots, G_8^\Sigma(u, v) \right]^\top, \]

\[ h_\Sigma(u, v) \]: histogram at location \((u, v)\)

\[ G_i^\Sigma \]: Gaussian convolved orientation maps

II. Histograms are normalized to unit norm

III. Local image descriptor is computed as

\[ D(u_0, v_0) = \left[ \begin{array}{c} \tilde{h}_{\Sigma_1}^\top(u_0, v_0), \\ \tilde{h}_{\Sigma_1}^\top(1(u_0, v_0, R_1)), \ldots, \tilde{h}_{\Sigma_1}^\top(1_N(u_0, v_0, R_1)), \\ \tilde{h}_{\Sigma_2}^\top(1(u_0, v_0, R_2)), \ldots, \tilde{h}_{\Sigma_2}^\top(1_N(u_0, v_0, R_2)), \\ \tilde{h}_{\Sigma_3}^\top(1(u_0, v_0, R_3)), \ldots, \tilde{h}_{\Sigma_3}^\top(1_N(u_0, v_0, R_3)) \end{array} \right]^\top \]
**DAISY v. SIFT: computational complexity**

- Convolution is time-efficient for separable kernels like Gaussian.

- Convolution maps with larger Gaussian kernel can be built upon convolution maps with smaller Gaussian kernel:

\[
G_o^{\Sigma_2} = G_{\Sigma_2} * \left( \frac{\partial I}{\partial o} \right)^+ = G_{\Sigma} * G_{\Sigma_1} * \left( \frac{\partial I}{\partial o} \right)^+ = G_{\Sigma} * G_o^{\Sigma_1},
\]

with \( \Sigma = \sqrt{\Sigma_2^2 - \Sigma_1^2} \).

<table>
<thead>
<tr>
<th>Image Size</th>
<th>DAISY</th>
<th>SIFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>800x600</td>
<td>5</td>
<td>252</td>
</tr>
<tr>
<td>1024x768</td>
<td>10</td>
<td>432</td>
</tr>
<tr>
<td>1290x960</td>
<td>13</td>
<td>651</td>
</tr>
</tbody>
</table>

Table 1. Computation Time Comparison (in seconds)
Shape Context

Count the number of points inside each bin, e.g.:

Count = 4

\[ \vdots \]

Count = 10

\[ \vdots \]

Compact representation of distribution of points relative to each point
Shape Context
Local Binary Pattern (LBP) Descriptor

The primitive LBP \((P,R)\) number that characterizes the spatial structure of the local image texture is defined as:

\[
LBP_{P,R} = \sum_{p=0}^{P-1} s(x)2^p, \quad x = g_p - g_c
\]

where,

\[
s(x) = \begin{cases} 
1 & \text{if } x \geq 0 \\
0 & \text{if } x < 0 
\end{cases}
\]

Circularly symmetric neighbor sets (\(P: \) angular resolution, \(R: \) spatial resolution)

LBP values in a 3 x 3 block

The LBP descriptor is invariant to any monotonic transformation of image
In order to remove the effect of rotation and assign a unique identifier to each, Rotation Invariant Local Binary Pattern is defined as:

\[ LBP^{ri}_{P,R} = \min \{ ROR(LBP_{P,R},i) \mid i = 0,1,..., P - 1 \} \]

where \( ROR(x,i) \) performs a circular bit-wise right shift on P-bit number \( x \), \( i \) time.

- 36 unique rotation invariant binary patterns can occur in the circularly symmetric neighbor set of \( LBP_{8,1} \).

slide credit: Sara Arasteh et al.
• This figure shows 36 unique rotation invariant binary patterns.

slide credit: Sara Arasteh et al.
Rotation Invariant LBP patterns include:

- **Uniform patterns**
  - At most two transitions from 0 to 1
- **Non-uniform patterns**
  - More than two transitions from 0 to 1

Samples of non-uniform patterns

Samples of uniform patterns

slide credit: Sara Arasteh et al.
It is observed that the uniform patterns are the majority, sometimes over 90 percent, of all $3 \times 3$ neighborhood pixels present in the observed textures.

They function as templates for microstructures such as:

- Bright spot (0)
- Flat area or dark spot (8)
- Edges of varying positive and negative curvature (1-7)

Uniform Local Binary Patterns

LBPs are popular, numerous modifications exist

slide credit: Sara Arasteh et al.
Detector:
- Scale-space peaks of Difference-of-Gaussians filter response (Lindeberg 1995)
- Similarity frame from modes of gradient histogram

SIFT Descriptor:
- Local histograms of gradient orientation
- Allows for small misalignments
  => robust to non-similarity transforms

Indexing:
- Modified kD-tree structure

Verification:
- Hough transform based clustering of correspondences with similar transformations

Fast, efficient implementation, real-time recognition

Nearest-neighbor matching

- Solve following problem for all feature vectors, $x$:

$$\forall j \quad NN(j) = \arg \min_i ||x_i - x_j||, \ i \neq j$$

- Nearest-neighbor matching is the major computational bottleneck
  - Linear search performs $dn^2$ operations for $n$ features and $d$ dimensions
  - No exact methods are faster than linear search for $d>10$ (?)
  - Approximate methods can be much faster, but at the cost of missing some correct matches. Failure rate gets worse for large datasets.
K-d tree construction

Simple 2D example

Slide credit: Anna Atramentov
K-d tree query

Slide credit: Anna Atramentov
Approximate k-d tree matching

Key idea:
- Search k-d tree bins in order of distance from query
- Requires use of a priority queue
- Copes better with high dimensionality
- Many different varieties
  - Ball tree, Spill tree etc.
Feature space outlier rejection

- How can we tell which putative matches are more reliable?
- Heuristic: compare distance of nearest neighbor to that of second nearest neighbor
  - Ratio will be high for features that are not distinctive
  - Threshold of 0.8 provides good separation

Randomized Forests

- Feature matching as a classification problem

Lepetit, Lagger and Fua. Randomized Trees for Real-Time Keypoint Matching, CVPR 2005
Synthesize training examples

- Planar object

3-D object

Lepetit, Lagger and Fua. Randomized Trees for Real-Time Keypoint Matching, CVPR 2005
Randomized Decision Tree

- Compare intensity of pairs of pixels
- In construction, pick pairs randomly

- Insert all training examples into tree

- Distribution at leaves is descriptor for the particular feature
Randomized Forests

- Use multiple trees (i.e. forest) to improve performance

- Very quick to compute in testing
  - Just comparison of pairs of pixels
  - Real-time performance

- ~10x faster than SIFT, but slightly inferior performance
1. Detect Distinguished Regions  Maximally Stable Extremal Regions (MSERs)
2. Construct Local Affine Frames (LAFs) (local coordinate frames)
3. Geometrically normalize some measurement region (MR) expressed in LAF coordinates
4. Photometrically normalize measurements inside MR, compute some derived description
5. Establish local (tentative) correspondences by the decision-measurement tree method
6. Verify global geometry (e.g. by RANSAC, geometric hashing, Hough transform.)

Matas, Chum, Urban, Pajdla: “Robust wide baseline stereo from maximally stable extremal regions”. BMVC2002
Obdržálek and Matas: “Object recognition using local affine frames on distinguished regions”. BMVC02
Obdržálek and Matas: “Sub-linear Indexing for Large Scale Object Recognition”, BMVC 2005
4. **Photometrically normalize** measurements inside MR, compute some derived description

[video-1, video-2]
“Recognition” as a Sequence of Wide-Baseline Matching Problems??

Properties: robust to occlusion, clutter, handles pose change, illumination but becomes unrealistic even for moderate number of objects.

Database processing  Matching  Query processing
Simultaneous Recognition of Multiple Objects Using the Decision-Measurement Tree
Performance Evaluation 1.: Image Retrieval from ZuBuD[1]

- Publicly available dataset ZuBuD
- Database: 201 buildings, each represented by 5 images, more than 1000 images in the DB
- Queries: 115 new images
- Forced match

Recognition rates (rank 1 correct):
- Repeated LAF-MSER matching: 100% @ 27 seconds /retrieval
- Tree matching:
  - 93% @ 0.014 seconds
  - 99% @ 0.510 seconds

MSER detector, SIFT descriptor, K-means tree

Very carefully implemented

Evaluated on large databases
  • Indexing with up to 1M images

Online recognition for database of 50,000 CD covers
  • Retrieval in ~1s
However:

- Recognition of images, not objects
- Some of the object have no chance of being recognized via MSER+SIFT on different background
Correspondence Verification

From image to local invariant descriptors

- Image
- Distinguished regions
- LAFs
- Normalization
- Descriptors

Correspondence between two images

- $I_1$:
  - Descriptors 1
  - Matching
  - Tentative correspondences
  - Correspondence verification
  - RANSAC
  - Filtered correspondences
  - Final correspondences (+ model)

- $I_2$:
  - Descriptors 2
Correspondence Verification

- Difficult matching problems:
  - Rich 3D structure with many occlusions
  - Small overlap
  - Image quality and noise
  - (Repetitive patterns)

Measurement region too large

Measurement region too small
Correspondence Verification

- **Idea**: “Look at both images simultaneously”
  
  => *Sequential Correspondence Verification by Cosegmentaion*


- **Input**: fixed number of tentative correspondences

- **Output**: Statistical Correspondence quality

- A cosegmentaion process starts from LAF-correspondences to grow corresponding regions

- Various statistics are collected

- (Learned) Classifier to decide corresponding/non-correspond.
Correspondence Verification

- Learning a (sequential) classifier
  - Training set from WBS images
  - 16k LAF correspondences (40% correct)
Correspondence Verification: Experiments

[Graphs showing the performance of various verification methods.]
Correspondence Verification: Summary

- high discriminability
  - significantly outperforms a standard selection process based SIFT-ratio

- very fast (0.5 sec / 1000 correspondences)

- always applicable before RANSAC

- the process generating tentative correspondences can be much more permissive
  - 99% of outliers not a problem, correct correspondences recovered
  - higher number of correct correspondences
Local Features: Application Examples

- Detection of goods in tray at supermarket checkout
- Database: 500 objects, 6 images each
  - Queries: images captured from a camera at the checkout
  - Output: list of objects identified in the tray
Local Features: Application Examples

- Traffic sign recognition from a moving car
- Database: images of known signs

Output: identification of signs in images taken by an in-car camera (scene-interpretation is not part of the system)
Local Features: Application Examples

- Detection of product logos in scanned commercials

- Detection of advertising side-boards in TV coverage of sport events. “For how long was my commercial actually broadcasted?”

- Detection of company logos in automatic fax processing
Local Features : Application Examples

- 3D reconstruction
3D reconstruction pipeline

1. **matching distinguished regions**
   - ➞ tentative correspondences (verification)
   - ➞ two view geometry

2. **camera calibration**
   - ➞ camera positions
   - ➞ sparse reconstruction

3. **dense stereoscopic matching**
   - ➞ pixel/sub-pixel matching
   - ➞ depth maps, 3D point cloud

4. **surface reconstruction**
   - ➞ surface refinement
   - ➞ triangulated 3D model
Local Feature Methods: Analysis

1. Methods work well for a non-negligible class of objects, that are locally approximately planar, compact and have surface markings or where 3D effects are negligible (e.g. stitching photographs taken from a similar viewpoint)

2. They are correspondence based methods
   • insensitive to occlusion, background clutter
   • very fast
   • handles very large dataset
   • model-building is automatic

3. The space of problems and objects where it does not work is HUGE (examples are all around us).
Where Local Features Fail:

Challenge: Elongated, Wiry and Flexible Objects

In this case: “no recognition without segmentation”? 
Camouflage: No distinguished regions!
Very few animals can afford to be distinguishable....
Thank you for your attention.