Visuelle Perzeption für Mensch-Maschine Schnittstellen

Vorlesung, WS 2009

Prof. Dr. Rainer Stiefelhagen
Dr. Edgar Seemann

Institut für Anthropomatik
Universität Karlsruhe (TH)

http://cvhci.ira.uka.de
rainer.stiefelhagen@kit.edu
seemann@pedestrian-detection.com
Computer Vision:

People Detection III

WS 2009/10

Dr. Edgar Seemann

seemann@pedestrian-detection.com
### Termine (1)

<table>
<thead>
<tr>
<th>Termine</th>
<th>Thema</th>
</tr>
</thead>
<tbody>
<tr>
<td>19.10.2009</td>
<td>Introduction, Applications</td>
</tr>
<tr>
<td>23.10.2009</td>
<td>Basics: Cameras, Transformations, Color</td>
</tr>
<tr>
<td>26.10.2009</td>
<td>Basics: Image Processing</td>
</tr>
<tr>
<td>30.10.2009</td>
<td>Basics: Pattern recognition</td>
</tr>
<tr>
<td>02.11.2009</td>
<td>Computer Vision: Tasks, Challenges, Learning, Performance measures</td>
</tr>
<tr>
<td>06.11.2009</td>
<td>Face Detection I: Color, Edges (Birchfield)</td>
</tr>
<tr>
<td>09.11.2009</td>
<td><strong>Project 1: Intro + Programming tips</strong></td>
</tr>
<tr>
<td>13.11.2009</td>
<td>Face Detection II: ANNs, SVM, Viola &amp; Jones</td>
</tr>
<tr>
<td>16.11.2009</td>
<td>Project 1: Questions</td>
</tr>
<tr>
<td>20.11.2009</td>
<td>Face Recognition I: Traditional Approaches, Eigenfaces, Fisherfaces, EBGM</td>
</tr>
<tr>
<td>23.12.2009</td>
<td>Face Recognition II</td>
</tr>
<tr>
<td>27.11.2009</td>
<td>Head Pose Estimation: Model-based, NN, Texture Mapping, Focus of Attention</td>
</tr>
<tr>
<td>30.11.2009</td>
<td>People Detection I</td>
</tr>
<tr>
<td>03.12.2009</td>
<td>People Detection II</td>
</tr>
<tr>
<td>07.12.2009</td>
<td><strong>Project 1: Student Presentations, Project 2: Intro</strong></td>
</tr>
<tr>
<td>11.12.2009</td>
<td>People Detection III (Part-Based Models)</td>
</tr>
<tr>
<td>14.12.2009</td>
<td>People Detection IV</td>
</tr>
<tr>
<td>18.12.2009</td>
<td>Scene Context, Geometry, Stereo and Optical Flow</td>
</tr>
<tr>
<td>21.12.2009</td>
<td>TBA</td>
</tr>
</tbody>
</table>
Today

- Part-Based Models
  - Flexible Spatial Layout
  - Pictorial Structures

- Local Features [credits: K. Mikolajczyk, B. Leibe, B. Schiele]
  - Interest Point Detectors
  - Scale Selection
Flexible Spatial Layout: Pictorial Structures
Example: Ronfard et al.

- Body is decomposed into a tree of 14 parts
- Each part is described by
  - Relative position to parent (dx, dy, dz)
  - Angular orientation (α, β, γ)
Ronfard et al.: Part recognition

- Recognition done on
  - 8 scales
  - 36 orientations (= 10° steps)
- 14x24 pixels for each part are used
- Used features include:
  - $G, G_x, G_y, G_{xx}, G_{xy}, G_{yy}$
- Classification based on SVMs

We obtain 14x24x6 features, which are NOT scale and orientation invariant.
Part Feature Examples

Figure 3. The $\nabla_x G$ and $\nabla_y G$ feature images for the example in Figure 2.
Efficient Body tree matching
[Felzenszwalb & Huttenlocher CVPR '2000]

- Parsing body trees $E$ with configuration $L$
- Optimal solution:

$$L^* = \arg \min_L \left( \sum_{(v_i,v_j) \in E} d_{ij}(l_i,l_j) + \sum_{v_i \in V} m_i(I,l_i) \right)$$

- $v_i$: body part candidate
- $l_i$: location of body part candidate $v_i$
- $d(l_i,l_j)$: distance function (considering orientation)
- $m(l_i)$: matching score (e.g. SVM score of part detector)
- $I$: the image

Complexity for minimization in its general form: $O(m^n)$ where $n$ #body parts, $m$ #candidates.
Efficient Body tree matching
[Felzenszwalb & Huttenlocher CVPR ‘2000]

- Parsing algorithm based on the following intuition:
  - for the tree leaves (hands, feet, head):
    - best location depends on SVM score + location of parent
  - for internal nodes
    - best location depends on SVM score + parent + children
  - for the root (torso)
    - best location depends on SVM score + children

Calculate best body part locations by recursion
dynamic programming.
Algorithm

- **Leaf nodes**
  - Best location for a leaf node $l_j$ depends only on score and the parent
  - The quality of the best location can be expressed by
    \[
    B_j(l_i) = \min_{l_j} (d_{i,j}(l_i, l_j) + m_j(I, l_j))
    \]

- **We continue bottom up**
  - The quality of the best location of a node at location $l_j$ can be expressed by
    \[
    B_j(l_i) = \min_{l_j} \left( d_{i,j}(l_i, l_j) + m_j(I, l_j) + \sum_{v_c \in C_j} B_c(l_j) \right)
    \]
Algorithm continued

Finally, for the root $v_r$, if $B_c(l_r)$ is known for each child $v_c \in C_r$ then the best location of the root is

$$l^*_r = \min_{l_r} \left( m_r(I, l_r) + \sum_{v_c \in C_r} B_c(l_j) \right)$$

- Note, that knowing the best root location, we can also infer the complete body tree again

Complexity $O(m^2n)$ where $n$ #body parts, $m$ #candidates.
Extension

- Matching is still quite costly as it’s quadratic in the number of part candidates.
- In practice this can be prohibitive as part detectors typically have high false positive rates.
- We can do even better:
  - By restricting the pairwise cost function $d$, we can achieve a complexity of $O(mn)$.
  - See original paper of Fellzenszwalb & Huttenlocher CVPR’2000.
Ronfard et al: Results

- Training done on 100 hand-labelled sample images
- Part detection:
  - false positive rates of approximately 14:15 (for 100% recall)
  - Low precision
- Overall detection:
  - recognition rates of 85% if only 55% of the body parts were positioned correctly
Ronfard et al.

Edgar Seemann, 11.12.09
Felzenszwalb & Huttenlocher

- Color-based part detectors
- Tuned to a single instance
- Different sizes for parts
- Indoor
Summary

- Relating detected parts helps disambiguating hypotheses
- Pose can be recovered from final body tree
- Relative position of nodes make model suitable for non-rigid objects
Flexible Spatial Layout: Probabilistic Model
Example: Mikolajczyk et al [ECCV’04]

- Sophisticated part models
- Relatively few parts
- Probabilistic assembly of parts
  - Joint-Likelihood model of appearance and geometric relations
  - Modeled in a Bayesian framework

\[
p(B|\mathcal{R}, \mathcal{F}) \frac{p(\mathcal{R}, \mathcal{F})}{p(\text{non } B|\mathcal{R}, \mathcal{F})} = \frac{p(\mathcal{R}|\mathcal{F}, B)}{p(\mathcal{R}|\mathcal{F}, \text{non } B)} \cdot \frac{p(\mathcal{F}|B)}{p(\mathcal{F}|\text{non } B)} \cdot \frac{p(B)}{p(\text{non } B)}
\]

- With B Body, R geometric relations and F feature values
Body Parts

- Only few body parts
- Different parts for front/side-views

Fig. 1. Body parts. (a) Frontal head and face (inner frame). (b) Profile head and face (inner frame). (c) Frontal upper body. (d) Profile upper body. (e) Legs.
Geometric Relationships

- Modeled by Gaussian distributions

Fig. 2. Gaussian geometric relations between body parts. (a) Frontal face location. (b) Frontal head location with respect to the face location. (c) Profile location. (d) Profile head location with respect to the profile location. (e) Profile upper body location with respect to the head. (f) Frontal upper body location with respect to the head location, and legs with respect to the upper body.
Feature Representation

- **Orientation features**
  - Gradients and Laplacian responses
  - Dominant orientation in a neighborhood
  - Computed on 5 scales

Fig. 3. Orientation features. (a) Head image. (b) Gradient image. (c) Dominant gradient orientations. (d) Positive Laplacian responses. (e) Dominant orientations of the second derivatives.

- **Feature Groups**
  - 3 vertical
  - 3 horizontal
  - Location quantized in a 5x5 grid

Fig. 4. Local groups of features. (a) Two groups of local orientations. (b) Location of the feature on the object. (c) Grid of quantized locations.
Part Classifier

- Based on AdaBoost
- Weak classifiers
  - Single occurrence
    \[ h_{f_a} = \ln \left( \frac{p(f_a|\text{object})}{p(f_a|\text{non object})} \right) \]
  - Joint occurrence
    \[ h_{f_{ab}} = \ln \left( \frac{p(f_a, f_b|\text{object})}{p(f_a, f_b|\text{non object})} \right) \]
- Cascade of classifier similar to Viola&Jones
Full body model

- Start from a single confident part detection
- Search for other body parts, which comply with the detection
- Compute the combined probabilistic score

Insights:
- Leg detector unreliable
- Face detection quite reliable
Example Results
Local Features
So far

- Parts were defined manually
- Parts represented the semantic structure
  - i.e. face, leg etc.

Questions:
- Do these parts decompose the variability in an optimal way?
  - Must the parts have a semantic meaning
  - Should we use smaller/larger parts?
- Can we find parts automatically?
Requirements for part decomposition

- **Repeatable**
  - i.e. we should be able to find the part despite articulation or image transformations (e.g. rotation, perspective, lighting)

- **Distinctive**
  - Part should not be confounded with other parts
  - The regions should contain an “interesting” structure

- **Compact**
  - Typically no lengthy or strangely shaped parts

- **Efficient**
  - It should be computationally inexpensive to detect or represent part

- **Cover**
  - parts need to sufficiently cover the object
Going local

- Local Feature Approaches
  - Use a large number of parts (typically 100-10000 parts)
  - Parts have mostly no direct semantic meaning
  - Parts are generated automatically

- Let algorithm find its own parts
- Typically smaller parts
Keypoints and descriptors

- We distinguish
  - Key or interest points
  - Local (key point) descriptors

- Interest Points
  - Specify repeatable points on the object
  - x-, y-position and scale

- Local Descriptors
  - Define the feature representation around an interest point
Approach

1. Find a set of distinctive keypoints
2. Define a region around each keypoint
3. Extract and normalize the region content
4. Compute a local descriptor from the normalized region
5. Match local descriptors

\[ d(f_A, f_B) < T \]
Key Point Detectors
Key Point Detectors

- Many Existing Detectors Available
  - Hessian & Harris [Beaudet ‘78], [Harris ‘88]
  - Laplacian, DoG [Lindeberg ‘98], [Lowe 1999]
  - Harris-/Hessian-Laplace [Mikolajczyk & Schmid ‘01]
  - Harris-/Hessian-Affine [Mikolajczyk & Schmid ‘04]
  - EBR and IBR [Tuytelaars & Van Gool ‘04]
  - MSER [Matas ‘02]
  - Salient Regions [Kadir & Brady ‘01]
  - Others…

- Reference site:
  - http://www.robots.ox.ac.uk/~vgg/research/affine/index.html
Keypoint Localization

- Goals:
  - Repeatable detection
  - Precise localization
  - Interesting content

⇒ Look for two-dimensional signal changes
Hessian Detector \([\text{Beaudet78}]\)

- **Hessian determinant**

\[
\text{Hessian} (I) = \begin{bmatrix}
I_{xx} & I_{xy} \\
I_{xy} & I_{yy}
\end{bmatrix}
\]

**Intuition:** Search for strong derivatives in two orthogonal directions
Hessian Detector [Beaudet78]

- Hessian determinant

\[ \text{Hessian}(I) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix} \]

\[ \text{det}(\text{Hessian}(I)) = I_{xx}I_{yy} - I_{xy}^2 \]

In Matlab:

\[ I_{xx} \cdot I_{yy} - (I_{xy})^2 \]

Edgar Seemann, 11.12.09
Effect: Responses mainly on corners and strongly textured areas.

[Beaudet78]
Hessian Detector – Responses [Beaudet78]
Harris Detector [Harris88]

- Second moment matrix
  (autocorrelation matrix)

\[
\mu(\sigma_i, \sigma_D) = g(\sigma_i) \ast \begin{bmatrix}
I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\
I_x I_y(\sigma_D) & I_y^2(\sigma_D)
\end{bmatrix}
\]

*Intuition:* Search for local neighborhoods where the image content has two main directions (eigenvectors).
Harris Detector [Harris88]

- Second moment matrix (autocorrelation matrix)

\[
\mu(\sigma_I, \sigma_D) = g(\sigma_I) \ast \begin{bmatrix}
I_x^2(\sigma_D) & I_xI_y(\sigma_D) \\
I_xI_y(\sigma_D) & I_y^2(\sigma_D)
\end{bmatrix}
\]

1. Image derivatives
   \(g_x(\sigma_D), \ g_y(\sigma_D),\)
Harris Detector [Harris88]

- Second moment matrix (autocorrelation matrix)
  \[ \mu(\sigma_I, \sigma_D) = g(\sigma_I) \ast \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix} \]

1. Image derivatives
   \( I_x(\sigma_D), \ I_y(\sigma_D), \)

2. Square of derivatives
   \( I_x I_y \)
Harris Detector [Harris88]

- Second moment matrix (autocorrelation matrix)

\[ \mu(\sigma_I, \sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix} \]

1. Image derivatives
2. Square of derivatives
3. Gaussian filter \( g(\sigma_I) \)
Harris Detector [Harris88]

- Second moment matrix (autocorrelation matrix)
  \[
  \mu(\sigma_1, \sigma_D) = g(\sigma_I) \ast \begin{bmatrix}
  I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\
  I_x I_y(\sigma_D) & I_y^2(\sigma_D)
  \end{bmatrix}
  \]

1. Image derivatives
2. Square of derivatives
3. Gaussian filter \( g(\sigma_I) \)

\[
har = \det[\mu(\sigma_1, \sigma_D)] - \alpha[\text{trace}(\mu(\sigma_1, \sigma_D))] = \\
g(I_x^2)g(I_y^2) - [g(I_x I_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2
\]

5. Non-maxima suppression

Edgar Seemann, 11.12.09
Harris Detector – Responses [Harris88]

**Effect:** A very precise corner detector.
Harris Detector – Responses [Harris88]
Scale Space

- So far, we can detect repeatable points in the image
- Now what about the image scale?
- Can we not only detect a distinctive position, but also a characteristic scale around an interest point?
Automatic Scale Selection

\[ f(I_{i_1...i_m}(x, \sigma)) = f(I_{i_1...i_m}(x', \sigma')) \]

Same operator responses if the patch contains the same image up to scale factor
Automatic Scale Selection

- Function responses for increasing scale (scale signature)

Edgar Seemann, 11.12.09
Automatic Scale Selection

- Function responses for increasing scale (scale signature)
Automatic Scale Selection

- Function responses for increasing scale (scale signature)

\[ f(I_{x \ldots x_m}(x, \sigma)) \]

\[ f(I_{x \ldots x_m}(x', \sigma)) \]
Automatic Scale Selection

- Function responses for increasing scale (scale signature)
Automatic Scale Selection

- Function responses for increasing scale (scale signature)

\[ f(I_{h\ldots j\_m}(x, \sigma)) \]

\[ f(I_{l_1\ldots j\_m}(x', \sigma)) \]
Automatic Scale Selection

- Function responses for increasing scale (scale signature)

\[ f(I_{i\ldots i_m}(x, \sigma)) \]

\[ f(I_{i\ldots i_m}(x', \sigma')) \]
What Is A Useful Signature Function?

- Laplacian-of-Gaussian = “blob” detector
Laplacian-of-Gaussian (LoG)

- Local maxima in scale space of Laplacian-of-Gaussian

\[ L_{xx}(\sigma) + L_{yy}(\sigma) \Rightarrow (x, y, s) \]
Results: Laplacian-of-Gaussian
Maximally Stable Extremal Regions [Matas ‘02]

- Based on Watershed segmentation algorithm
- Select regions that stay stable over a large parameter range
Example Results: MSER
Local Descriptors
Local Descriptors

- Most available descriptors focus on edge/gradient information
  - Capture boundary and texture information
  - Color still used relatively seldom
    (more suitable for homogenous regions)
Local Descriptors: SIFT Descriptor

Histogram of oriented gradients
  • Captures important texture information
  • Robust to small translations / affine deformations

[Lowe, ICCV 1999]
Orientation Normalization

- Compute orientation histogram
- Select dominant orientation
- Normalize: rotate to fixed orientation

[Lowe, SIFT, 1999]
Local Descriptors: SURF

Fast approximation of SIFT idea
Efficient computation by 2D box filters & integral images
⇒ 6 times faster than SIFT
Equivalent quality for object identification

GPU implementation available
Feature extraction @ 100Hz
(detector + descriptor, 640×480 img)
http://www.vision.ee.ethz.ch/~surf

[Bay, ECCV’06], [Cornelis, CVGPU’08]
Local Descriptors: Shape Context

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

Count = 4

\[ \vdots \]

Count = 10

Log-polar binning: more precision for nearby points, more flexibility for farther points.

Belongie & Malik, ICCV 2001
Local Descriptors: Geometric Blur

Compute edges at four orientations
Extract a patch in each channel

Apply spatially varying blur and sub-sample

Example descriptor

Berg & Malik, CVPR 2001
So, What Local Features Should I Use?

- There have been extensive evaluations/comparisons
  - [Mikolajczyk et al., IJCV’05, PAMI’05]
  - All detectors/descriptors shown there work well

- Best choice often application dependent
  - MSER works well for buildings and printed things
  - Harris-/Hessian-Laplace/DoG work well for many natural categories

- More features are better
  - Combining several detectors often helps