Similarity-based cross-layered hierarchical representation for object categorization
by Sanja Fidler, Ales Leonardis, et al.

Slides by Edward Wertz

March 24, 2014
Introduction

Brief Introduction To Gabor Filters

Hierarchical Object Structure

Scaling On Hierarchical Object Structure

Similarity and Shapinals in Hierarchical Object Structure

References
Introduction

Brief Introduction To Gabor Filters

Hierarchical Object Structure

Scaling On Hierarchical Object Structure

Similarity and Shapinals in Hierarchical Object Structure

References
Problem Description

Problem
We want a tractable system which is able to learn to recognize and categorize objects within images through unsupervised online learning.

Solution
In these slides we will be examining the solution put forward by Sanja Fidler, Ales Leonardis, et al. which describes objects as a hierarchy of composed parts
Fidler, et al. have refined their solution over the following papers:

Each paper builds on the previous paper, and these slides present the primary concepts and mathematics in the same order as the papers. But first we take a brief look at Gabor Filters since they are used at the lowest level of the object hierarchies.
Introduction

Brief Introduction To Gabor Filters

Hierarchical Object Structure

Scaling On Hierarchical Object Structure

Similarity and Shapinals in Hierarchical Object Structure

References
Figure 4: Gabor filter composition: (a) 2D sinusoid oriented at 30° with the x-axis, (b) a Gaussian kernel, (c) the corresponding Gabor filter. Notice how the sinusoid becomes spatially localized.

Figure 5: Example of Gabor filters with different frequencies and orientations. First column shows their 3D plots and the second one, the intensity plots of their amplitude along the image plane.

Figure: Image From [5]
Gabor Filter Energy

Figure 1: Top: An input signal. Second: Output of Gabor filter (cosine carrier). Third: Output of Gabor Filter in quadrature (sine carrier); Fourth: Output of Gabor Energy Filter

Figure: Image From [4]
we obtain a Gabor filter - Figure 4(c). Let $g(x, y, \theta, \phi)$ be the function defining a Gabor filter centered at the origin with $\theta$ as the spatial frequency and $\phi$ as the orientation. We can view Gabor filters as:

$$g(x, y, \theta, \phi) = \exp\left(-\frac{x^2 + y^2}{\sigma^2}\right) \exp\left(2\pi\theta i(x \cos \phi + y \sin \phi)\right)$$  

(1)

It has been shown that $\sigma$, the standard deviation of the Gaussian kernel depends upon the spatial frequency to measured, i.e. $\theta$. In our case, $\sigma = 0.65\theta$. Figure 5 shows 3D...

From [5]
Image analysis typically occurs by performing multiple convolutions of the Gabor Filter at different angles.
Convolution

http://fourier.eng.hmc.edu/e161/lectures/convolution/convolution.html
Figure 6: (a) An image, (b) The response for Gabor filter oriented horizontally - white indicates high amplitude of response, black indicates low response. Notice how regions of vertical stripes are highlighted.

**Figure:** Image From [5]
Gabor Filter Example

Chinese Character OCR.

http://en.wikipedia.org/wiki/Gabor_filter
Introduction

Brief Introduction To Gabor Filters

Hierarchical Object Structure

Scaling On Hierarchical Object Structure

Similarity and Shapinals in Hierarchical Object Structure

References
Hierarchical Organization

- A large number of parts is computationally infeasible.
- A small number of parts means having too many hypotheses to verify.
- The compromise is to have a layered system which manages the complexity.
Properties of The System

Top-Down Verification

- In order to enable robust verification, parts on higher layers should easily indicate and verify the presence of its underlying components on a lower layer.

Bottom-Up Learning

- Parts and their higher level combinations should be learned in an unsupervised manner.
Properties of Parts

Locality

- The receptive field in which parts are formed should not be too large to avoid complex computation on a large number of feature combinations.
- Features that appear close together are likely to belong to the same entity.

Shift and Rotation Invariance

- Recognition should proceed independently of position and orientation, but not lose information about local structure.

Encode Geometrical Structure

- The presence of features is not enough to create robust identity, the relative locations of features with respect to each other is necessary to capture the structure and shape of the objects depicted.
Figure 1. The hierarchical top-down and bottom-up architecture
Definition Of Parts

$P^n_i$ is the $i_{th}$ part in layer $n$ and has the following components:

1. Center of mass located at $(x, y) = (0, 0)$ with respect to itself.
2. Orientation of $\theta = 0$ degrees with respect to itself.
3. A list of sub-parts of the form $(P^{n-1}_j, \theta_j, (x_j, y_j)^T, \sigma_j)$ where:
   - $\theta_j$ and $(x_j, y_j)$ indicate the relative orientation and position of $P^{n-1}_j$ with respect to $P^n_i$.
   - $\sigma_j$ indicates the allowed variance in the relative pose.

Transformation of $P^n_i$ by translation $(x_t, y_t)$ and rotation $\beta$ results in the following transformation on a sub-part:

- $(P^{n-1}_j, \theta_j + \beta, (x_t, y_t)^T + \text{ROT}(\beta) \ast (x_j, y_j)^T, \sigma_j)$ where $\text{ROT}(\beta)$ denotes the rotation matrix of $\beta$ degrees.
Algorithm 1: A hierarchical learning architecture

1: Top-down projection of parts defining Layer I (oriented filters)
2: Bottom-up statistical learning of local configurations of parts of Layer I
   → Result: Parts for Layer II
3: Top-down projection of Layer II parts
4: Bottom-up learning using parts of Layer II
   → Result: Parts for Layer III
   .
   .
Algorithm 2: Top-down

1: for each subpart $\mathcal{P}^{n-1}$ found in the image do
2: for each part $\mathcal{P}^n$ containing $\mathcal{P}^{n-1}$ do
3: Vote for the center and orientation of $\mathcal{P}^n$
4: According to the hypothesized center check for all subparts comprising $\mathcal{P}^n$ (variance in the position is allowed)
5: end for
6: All fully verified parts $\mathcal{P}^n$ are passed to step 8
7: end for
8: Part selection with the MDL principle
Figure 2. Voting of subparts for part $P^n_i$
Algorithm 3: Bottom-up learning

1: for each image scale do
2:   for each part $\mathcal{P}^{n-1}$ activated in an image do
3:     Local neighborhood is defined and normalized with respect to $\mathcal{P}^{n-1}$
     Complexity is decoupled and separate statistics are sought:
4:     what parts (only most frequent cases are passed on to the following stage)
5:     orientation parts (only most frequent cases are passed on to the following stage)
6:     where parts
7:   end for
8: end for
Figure 3. Bottom-up learning procedure of local configurations (left) by decoupling the complexity in separate statistics.
Layer Parts Example

Figure 7. The first row depicts the final parts comprising Layer II obtained for (a) Cliparts and (b) Airplanes. The variances of position distributions of parts, relative to the central part, are depicted in the middle. The feature probabilities are listed in the last row.

Figure 8. (a) Examples of Layer 3 parts, (b) variances of positions of the surrounding subparts
Introduction

Brief Introduction To Gabor Filters

Hierarchical Object Structure

Scaling On Hierarchical Object Structure

Similarity and Shapinals in Hierarchical Object Structure

References
Recognizing Scale

In order to recognize objects at different scales, the learned images were scaled and training occurred at each scale.
New Notation

Additional Notation is introduced.

1. \( \text{Links}(P^n_i) \) is a list of all identities of layer \( L_{n+1} \) parts that \( P^n_i \) indexes to.

2. \( \pi^n_i = \{P^n, \theta_i, x_i, y_i\} \) is the realization of part \( P^n \) from layer \( L_n \) (where it occurred in the image).

3. \( \Lambda_n(\pi^n_i) \) represents a list of all image location points that contributed to part \( \pi^n_i \).

4. \( \Lambda_1 \) indicates the location of the image pixels that caused a Gabor Filter to fire, indicating a primitive part was detected.

5. \( C^n \) is a set of sets of parts from \( L_n \). Each \( C^n_i \in C^n \) is a candidate for becoming a part on layer \( L_{n+1} \).
Algorithm 1: Indexing and matching

1: \textbf{INPUT:} \{\{\pi_i^{n-1}\}_i, \Lambda_{n-1}\}_{s=1}^{n_{\text{sc}ales}}
2: \textbf{for each scale do}
3: \quad \Pi_{\text{scale}} = \{\}
4: \quad \textbf{for each } \pi_i^{n-1} = \{P_{i_k}^{n-1}, \alpha_i, x_i, y_i\} \textbf{ do}
5: \quad \quad \text{Rotate the neighborhood of } \pi_i^{n-1} \text{ by angle } -\alpha_i
6: \quad \quad \textbf{for each part } P^n \in \text{Links}(P_{i_k}^{n-1}) \textbf{ do}
7: \quad \quad \quad \text{Check for subparts of } P^n \text{ according to their relative positions and spatial variance}
8: \quad \quad \quad \textbf{if subparts found then}
9: \quad \quad \quad \quad \text{add } \pi^n = \{P^n, \alpha_i, x_i, y_i\} \text{ to } \Pi_{\text{scale}}, \set\Lambda_n(\pi^n) = \bigcup \Lambda_{n-1}(\pi_j^{n-1}), \text{ where } \pi_j^{n-1} \text{ are the found subparts of } P^n.$$
10: \quad \quad \quad \textbf{end if}$$
11: \quad \quad \textbf{end for}$$
12: \quad \textbf{end for}$$
13: \textbf{end for}$$
14: \text{Perform local inhibition over } \{\pi_i^n\}$
15: \textbf{return } \{\{\pi_i^n\}_i, \Lambda_n\}_{s=1}^{n_{\text{sc}ales}}
Equivalence Of Parts

We define the overlap of two parts $\pi^m_i$ and $\pi^m_j$ found in an image as: $overlap(\pi^m_i, \pi^m_j) = \min \left( \frac{|\Lambda(\pi^m_i) \cap \Lambda(\pi^m_j)|}{|\Lambda(\pi^m_i)|}, \frac{|\Lambda(\pi^m_j) \cap \Lambda(\pi^m_i)|}{|\Lambda(\pi^m_j)|} \right)$. The overlap of parts $\mathcal{P}^n_k$ and $\mathcal{P}^n_l$ is calculated as the average overlap of all image parts $\pi^m_i$ and $\pi^m_j$ that encode the identities $\mathcal{P}^n_k$ and $\mathcal{P}^n_l$, respectively. This measures the average number of overlapping image layer points that the two parts describe.

According to the acquired statistics, parts $\mathcal{P}^n_k$ and $\mathcal{P}^n_l$ are pronounced equal (their identities are set equal) if their average overlap is high.
Algorithm 2: Learning of $s-$subcompositions

1: INPUT: Collection of images
2: for each image and each $scale$ do
3:   Preprocessing:
4:   process image with $L_1$ parts to produce $\left\{ \{\pi^1_i\}_i, \Lambda_1 \right\}$
5:   for $k = 2$ to $n - 1$ do
6:     $\left\{ \{\pi^k_i\}_i, \Lambda_k \right\} = \text{Algorithm } 1(\left\{ \{\pi^{k-1}_i\}_i, \Lambda_{k-1} \right\})$
7:   end for

Learning:
8: for each $\pi^{n-1}_i = \{\mathcal{P}^{n-1}, x_i, y_i\}$ do
9:   for each $C^n_s \in \text{Links}(\mathcal{P}^{n-1})$ do
10:      Find all parts $\pi^{n-1}$ within the neighborhood
11:      Match the first $(s - 1)$-subparts contained within the subcomposition relative to the central part
12:      Perform local inhibition: $\Lambda(\text{neigh.parts}) := \Lambda(\text{neigh.parts}) \cup \Lambda(\text{found subparts})$. Keep parts that have $|\Lambda(\pi^{n-1})| \geq \text{thresh} \cdot |\Lambda(\pi^{n-1})|$. We use $\text{thresh} = 0.5$.
13:     If all $s - 1$ subparts are found and $s$-th subpart appears anywhere in the neighborhood, update the spatial map for the $s$-th subpart.
14:   end for
15: end for
16: end for
Figure 2. Learning of compositions by sequentially increasing the number of subparts
Figure 4. Mean reconstructions of the learned parts (spatial flexibility also modeled by the parts is omitted due to lack of space). **1st row**: $L_2$, $L_3$ (the first 186 of all 499 parts are shown), **2nd row**: $L_4$ parts for faces, cars, and mugs, **3rd row**: $L_5$ parts for faces, cars (obtained on 3 different scales), and mugs.
Figure 8. Detection of $\mathcal{L}_4$ and $\mathcal{L}_5$ parts. Detection proceeds bottom-up as described in Subsection 2.2. Active parts in the top layer are traced down to the image through links $\Lambda$. 
Introduction

Brief Introduction To Gabor Filters

Hierarchical Object Structure

Scaling On Hierarchical Object Structure

Similarity and Shapinals in Hierarchical Object Structure

References
Paper’s Contributions

Problems

1. Which hierarchical features to extract and forward to higher level category-specific representations?
2. Scale normalization of objects and their respective features
3. The problem of generalization (Allowing greater non-Gaussian variance).

Solutions

1. Extracting “Shapinals” or Shape Terminals which inhabit the lower layers
2. Establishing similarity connections between features appearing across various levels of the hierarchy.
3. Establishing intra level similarity connections to generalize classes over similar but distinct patterns of parts.
Figure 2. For greater generalization we establish similarities between hierarchical nodes within layers.
Figure 1. Cross-layered, scale independent representation.
We thus define a similarity measure $\text{sim}_{n,k}(\mathcal{P}_i^k, \mathcal{P}_j^k)$ between parts $\mathcal{P}_i^k$ and $\mathcal{P}_j^k$, both from Layer $k$, with respect to Layer $n$ recursively as follows.

$$\text{sim}_{n,k}(\mathcal{P}_i^k, \mathcal{P}_j^k) = \min\{\text{sim}'_{n,k}(\mathcal{P}_i^k, \mathcal{P}_j^k), \text{sim}'_{n,k}(\mathcal{P}_j^k, \mathcal{P}_i^k)\}$$

for $1 < k \leq n$ and

$$\text{sim}_{n,1}(\mathcal{P}_i^1, \mathcal{P}_j^1) = \rho(\mathcal{R}_{f,n,1}(\text{filter}_i), \mathcal{R}_{f,n,1}(\text{filter}_j))$$
otherwise, where

\[
sim'_{n,k}(\mathcal{P}_i^k, \mathcal{P}_j^k) = M(\text{psim}_{n,k}(\mathcal{P}_i^{k-1}, \mathcal{P}_j^k), \ldots, \text{psim}_{n,k}(\mathcal{P}_i^{k-1}, \mathcal{P}_j^k)),
\]

and

\[
\text{psim}_{n,k}(\mathcal{P}_i^{k-1}, \mathcal{P}_j^k) = \max\{\text{sim}_{n,k-1}(\mathcal{P}_i^{k-1}, \mathcal{P}_j^{k-1}), \rho(\mathcal{R}_{f_n,k}(\text{map}_i), \mathcal{R}_{f_n,k}(\text{map}_j))\}.
\]

Here \( \rho(A, B) = \frac{A \cdot B}{\|A\| \cdot \|B\|} \) is a measure for the similarity between maps \( A \) and \( B \) (\( A \cdot B \) denotes a component-wise inner product of two matrices and \( \|A\| \) is a norm induced by this product). Note that \( \rho(A, A) = 1 \) and \( \rho(A, B) = 0 \) when the supports of \( A \) and \( B \) are disjoint.
Next, we observe that \( \text{psim}_{n,k}(P_{i_t}^{k-1}, P_j^k) \) gives a similarity between \( P_{i_t}^{k-1} \), a subpart of \( P_i^k \), and the best matching subpart of \( P_j^k \). For \( \text{sim}'_{n,k}(P_i^k, P_j^k) \) we would like to give an average similarity between the subparts of \( P_i^k \) and their best matched subparts of \( P_j^k \). To do this, we take:

\[
M_{\text{avg}}(x_1, x_2, \ldots, x_m) = \frac{1}{m} \sum_{i=1}^{m} x_i,
\]

\[
M(x_1, x_2, \ldots, x_m) = \begin{cases} 
0, & \text{if there exists } j \text{ s.t. } x_j < T, \\
M_{\text{avg}}(x_1, x_2, \ldots, x_m), & \text{otherwise}
\end{cases}
\]
lowing way. Let a similarity measure between the parts $\mathcal{P}^k_i$ and $\mathcal{P}^{k'}_j$, on Layers $k$ and $k'$ with respect to Layer $n$, be

$$
sim_{n,k,k'}(\mathcal{P}^k_i, \mathcal{P}^{k'}_j) = M(ps \text{sim}_{n,k,k'}(\mathcal{P}^{k-1}_{i1}, \mathcal{P}^{k'}_j), \ldots, ps \text{sim}_{n,k,k'}(\mathcal{P}^{k-1}_{im}, \mathcal{P}^{k'}_j))
$$

for $1 < k' < k \leq n$, and

$$
sim_{n,k,1}(\mathcal{P}^k_i, \mathcal{P}^1_j) = \rho(CR(\mathcal{P}^k_i), R_{f_{1,k}}(filter_j)),
$$

for $1 < k \leq n$, where $CR(\mathcal{P}^k_i)$ denotes the reconstruction of the part $\mathcal{P}^k_i$ with filters on Layer 1 and

$$
ps \text{sim}_{n,k,k'}(\mathcal{P}^{k-1}_{i_t}, \mathcal{P}^{k'}_j) = \max_{j_l} \{ sim_{n,k-1,k'-1}(\mathcal{P}^{k-1}_{i_t}, \mathcal{P}^{k'-1}_{j_l}), \rho(R_{f_{n,k}}(map_{i_t}), R_{f_{n,k'}}(map_{j_l})) \}.
$$
**Algorithm 1**: Pooling of *shapinals* in images

1. **INPUT**: A list of parts found at each layer and scale:
   \[
   \Pi_{all} = \{\pi_{i, \text{scale}}, \Lambda_{i_{layer}}(\pi_i)\}_{i, \text{scale}, i_{layer}}
   \]

2. \[\Pi_{shapinals} = \emptyset\]

3. sort \(\Pi_{all}\) by decreasing value of \(|\Lambda_{i_{layer}}|\) (corresponding to the number of described image points)

4. **while** \(\Pi_{all} \neq \emptyset\) **do**

5. add the part that describes most image points to the *shapinal* list with its corresponding layer-independent label, \(\pi_{i, i_{layer}, \text{scale}} \in \Pi_{all}(1)\) and the value of *level-of-detail* (see text for explanation):

   \[\Pi_{shapinals} := \Pi_{shapinals} \cup \{\text{Cross-layer}(\pi_i), \text{lod}_i\},\]

   where \(\text{lod}_i = \text{round}(i_{layer} + 0.5 \cdot \text{scale})\)

6. perform local inhibition with the selected part:

   - find all parts \(\pi_j \in \Pi_{all}\) that have
     \[
     1 - \frac{|\Lambda_j(\pi_j) \cap \Lambda_j(\pi_i)|}{\max\{|\Lambda_j(\pi_i)|,|\Lambda_j(\pi_j)|\}} < \text{thresh}
     \]
     (we take \(\text{thresh} = 0.3\))

7. remove \(\{\pi_j\}\) from \(\Pi_{all}\): \(\Pi_{all} = \Pi_{all} \setminus \{\pi_j\}\)

8. **end while**

9. **return** A list of *shapinals*, \(\Pi_{shapinals}\)
Table 1. Average processing time for different steps per image

<table>
<thead>
<tr>
<th>Layer</th>
<th>Processing time per image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer 1</td>
<td>1.6 s</td>
</tr>
<tr>
<td>Layer 2</td>
<td>0.54 s</td>
</tr>
<tr>
<td>Layer 3</td>
<td>0.66 s</td>
</tr>
<tr>
<td>Shapinals</td>
<td>0.22 s</td>
</tr>
</tbody>
</table>
Table 2. Average classification rate (in percentage) on Caltech 101

<table>
<thead>
<tr>
<th>Method</th>
<th>$N_{train} = 15$</th>
<th>$N_{train} = 30$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer 2 only</td>
<td>55</td>
<td>/</td>
</tr>
<tr>
<td>Layer 3 only</td>
<td>52.9</td>
<td>/</td>
</tr>
<tr>
<td>shapinal</td>
<td>60.5</td>
<td>66.5</td>
</tr>
<tr>
<td>Serre et al. [18]</td>
<td>44</td>
<td>/</td>
</tr>
<tr>
<td>Mutch et al. [13]</td>
<td>51</td>
<td>56</td>
</tr>
<tr>
<td>Ranzato et al. [16]</td>
<td>/</td>
<td>54</td>
</tr>
<tr>
<td>Ommer et al. [14]</td>
<td>/</td>
<td>61.3</td>
</tr>
<tr>
<td>Wolf et al. [21]</td>
<td>51.18</td>
<td>/</td>
</tr>
</tbody>
</table>
S. Fidler, M. Boben, and A. Leonardis.  
Similarity-based cross-layered hierarchical representation for object categorization.  

S. Fidler and A. Leonardis.  
Towards scalable representations of object categories: Learning a hierarchy of parts.  

Sanja Fidler, Gregor Berginc, and Ales Leonardis.  
Hierarchical statistical learning of generic parts of object structure.  

Javier R. Movellan.  
Tutorial on gabor filters.  

V. Shiv Naga Prasad and Justin Domke.  
Gaber filter visualization.  