Local feature based supervised object detection: sampling, learning and detection strategies

J. Michel¹, M. Grizonnet¹, J. Inglada¹, J. Malik², A. Bricier², O. Lahlou²

¹CENTRE NATIONAL D’ÉTUDES SPATIALES
²COMMUNICATIONS & SYSTÈMES

IGARSS, July 25-29, 2011
Introduction

Sampling strategies

Learning architecture

Detection strategy

Experimental results

Conclusion
Introduction

Context

- In remote sensing: promising methods, but still early stage
- Object detection almost operational in natural images (facial recognition...)
- Keys to success:
  - Extensive (open) databases
  - Carefully designed learning architecture

This work

- Try to benefit from advances in natural images
- While addressing earth observation data constraints
- Into a generic (supervised) object detection framework
Overview of the proposed object detection framework

Introduction

Sampling

Learning

Detecting

Results

Conclusion

Statistics estimation

Bank: Images + ground truth + areas

Object detection

Object Vectorial Map

Detector training

Negative examples generation

Features computation

Normalization

Generalization

SVM Learning

Performances evaluation

Normalization

SVM classification

IGARSS, July 25-29, 2011
Outline

Introduction

Sampling strategies

Learning architecture

Detection strategy

Experimental results

Conclusion
What should examples databases look like?

In works with natural images

- Positive samples $\rightarrow$ bounding boxes (big objects)
- Negative samples drawn from “Empty” images

In our work on VHR earth observation images

- Point at the objects center instead of boxes
- No “empty” images, but “big” images:
  - Ask for exhaustivity $\rightarrow$ cumbersome!
  - Restrain exhaustivity to user-defined areas

Our training database
Images + positive instances points + areas of exhaustivity
Sampling negative examples

How to sample negative examples?

- Random sampling
- In areas of exhaustivity
- Away from positive examples (inhibition radius)
- Up to a target density
- Also densify positive examples
Outline

Introduction

Sampling strategies

Learning architecture

Detection strategy

Experimental results

Conclusion
From examples to measurements and training sets

- Measure features at a given location, on a given radius
- Measured on each channel or on intensity
  - Local histograms
  - Histogram of oriented gradients
  - Haralick textures
  - Flusser moments
  - Fourier-Mellin coefficients
  - Local statistics (up to 4th order)
- Center and reduce measures
- Simulate more data by random perturbation (optional)
- Split into training and validation set
Learning and validation

- Learning done with SVM (but other can be plugged)
- Parameters optimization with cross-validation
- Performance evaluation: precision, recall, f-score

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue Flusser moments</td>
<td>0.884782</td>
<td>0.692288</td>
<td>0.776787</td>
</tr>
<tr>
<td>Blue Fourier-Mellin</td>
<td>0.869089</td>
<td>0.847577</td>
<td>0.858198</td>
</tr>
<tr>
<td>Blue statistics mvsK</td>
<td>0.658339</td>
<td>0.549244</td>
<td>0.598864</td>
</tr>
<tr>
<td>Green Flusser moments</td>
<td>0.841265</td>
<td>0.658676</td>
<td>0.738857</td>
</tr>
<tr>
<td>Green Fourier-Mellin</td>
<td>0.853684</td>
<td>0.863471</td>
<td>0.858549</td>
</tr>
<tr>
<td>Green statistics mvsK</td>
<td>0.657171</td>
<td>0.522929</td>
<td>0.582414</td>
</tr>
<tr>
<td>Nir Flusser moments</td>
<td>0.764981</td>
<td>0.512246</td>
<td>0.613608</td>
</tr>
<tr>
<td>Nir Fourier-Mellin</td>
<td>0.818453</td>
<td>0.785826</td>
<td>0.801808</td>
</tr>
<tr>
<td>Nir statistics mvsK</td>
<td>0.664266</td>
<td>0.192288</td>
<td>0.298242</td>
</tr>
<tr>
<td>Red Flusser moments</td>
<td>0.817311</td>
<td>0.6889</td>
<td>0.747632</td>
</tr>
<tr>
<td>Red Fourier-Mellin</td>
<td>0.842715</td>
<td>0.815268</td>
<td>0.828764</td>
</tr>
<tr>
<td>Red statistics mvsK</td>
<td>0.651087</td>
<td>0.468213</td>
<td>0.544711</td>
</tr>
</tbody>
</table>
Outline

Introduction

Sampling strategies

Learning architecture

Detection strategy

Experimental results

Conclusion
Step 1: The coarse grid detection process

Inputs

- Parameters of the trained model
- List of features
- Statistics to center and reduce measurements

Strategy

- Define a regular grid (finer step → more computation time)
- Measure features at each location (center and reduce)
- Apply trained classifier
- Keep positive responses
Step 2 : Modes detection

Drawbacks of coarse detection

▶ Multiple detections for one object instance
▶ Isolated false alarms

Density of detections more informative than detections alone

Solution

Apply the Mean-Shift mode seeking algorithm on the coarse detection map

▶ Isolated false alarm filtered by cluster size
▶ Detections per instance reduced (1 in most cases)
▶ Finer Localization
Outline

Introduction
Sampling strategies
Learning architecture
Detection strategy
Experimental results
Conclusion
Coarse detection maps for planes

**Left**: Flusser, Statistics, Fourier-Mellin, **Right**: Hog, local histograms
Modes detection map for planes

**Left:** Flusser, Statistics, Fourier-Mellin,  **Right:** Hog, local histograms
Outline

Introduction

Sampling strategies

Learning architecture

Detection strategy

Experimental results

Conclusion
Conclusion & Perspectives

Improvements are still needed . . .

- Add more features
- One classifier per object or one for all?
- Test and validate on other objects
- Need for a reference database: crowd sourcing?

A pre-operational system

- Efficient way to perform Object Detection
- Complete framework from database to detector
- Openness and Reproducibility (source code, documentation and a test dataset are available)

All experiments have been done using the Orfeo ToolBox

(www.orfeo-toolbox.org)
Questions

Thank you for your attention!

The ORFEO Toolbox is not a black box.
http://www.orfeo-toolbox.org