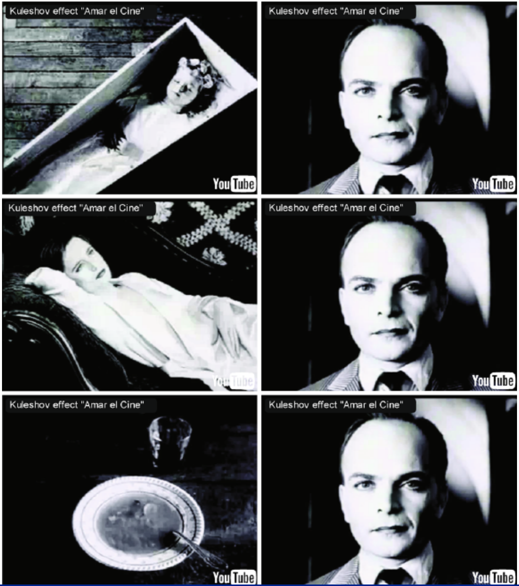
Context, Computation, and Optimal ROC Performance in Hierarchical Models

Chang, Jin, Zhang, Borenstein, Gemen

Introduction

- human vision relies on contextual information
- example: don't get meaning of a single word, but understand it in a full sentence.
- example: the kuleshov effect with an expressionless face between different situation.
- idea: start from local features like edges and continuing to high-level knowledge.

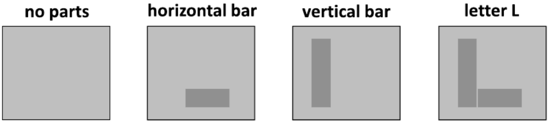

Kuleshov effect "Amar el Cine"
Kuleshov effect "Amar el Cine"
Kuleshov effect "Amar el Cine"
Kuleshov effect "Amar el Cine"
Kuleshov effect "Amar el Cine"
Kuleshov effect "Amar el Cine"

Introduction

- backgrounds are difficult to seperate
- most of false detection of an object occure at background locations that share pieces.
- only modeling the objects of interest!
- matching templates needs description of every pose (6 degrees of freedom)
- how to check for an object? when it is present?
- to avoid false positives, a model for **object is absent** is also needed
- better: test for parts of an object

Example

- describe objects with parts
- latent variables are binary, representing the absence (0) or presence (1) of a part



Latent variables ("interpretations")	$Z_1 = 0$	$Z_1 = 1$	$Z_1 = 0$	$Z_1 = 1$
	$Z_2 = 0$	$Z_2 = 0$	$Z_2 = 1$	$Z_2 = 1$
	$p_{Z_1,Z_2}(0,0)$	$p_{Z_1,Z_2}(1,0)$	$p_{Z_1,Z_2}(0,1)$	$p_{Z_1,Z_2}(1,1)$

Example

- the joint probability is $P(Z_1 = z_1, Z_2 = z_2), z_1, z_2 \in \{0, 1\}$.
- set of all pixels is S, S^1 and S^2 are subsets of S, denote the pixels belonging to parts of the object
- the random variable X_s denotes the intensity of pixel $s \in S$

Optimal Decision Rule

- L is present, if the horizontal and vertical bars are present, i.e. $\{Z_1 = 1\} \cap \{Z_2 = 1\}$.
- thresholding the posterior prabability $S_G(x_s) = P(Z_1 = 1, Z_2 = 1 | X_s = x_s)$
- letter L was found if $S_G(x_s) > t$ and was not found if $S_G(x_s) \leq t$
- following the Neyman-Pearson-Lemma, $S_G(x_s)$ is a monotone increasing function of the likelihood ratio $\frac{P(x_s|\text{L present})}{P(x_s|\text{L not present})}$
- problem: can not be computed in general, since the full likelihood mean a mixture over **every possible explanation** of the data

Template Matching

- approximate by pretending that the world has only two states **object** or **nothing**.
- $S_T(x_s)$

Testing for Parts

- performing tests sequentially.
- test for first part and **ignore** informations about the second part
- the second part is only computed, if the first part was tested
- $S_P(x_s)$

Foveal Limit

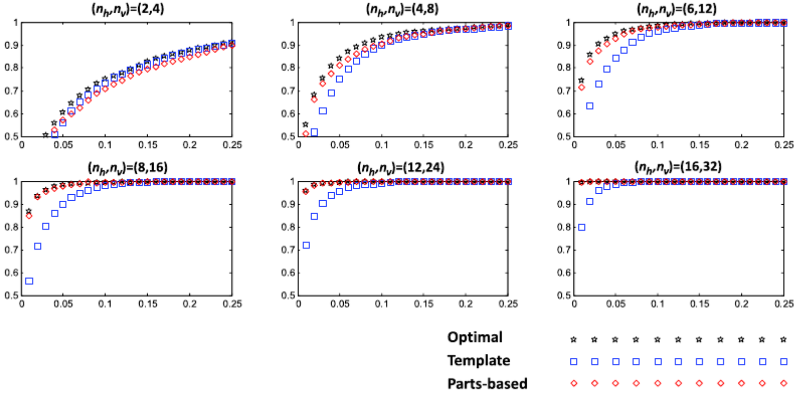
- compare previously described methods
- simple 'L'-example can be computed and analysed mathematicaly
- computing ROC curves in the limit as the density of pixels goes to infinty
- $\rightarrow$ the **foveal limit** (= high resolution)

ROC curves

Receiver **O**perating **C**haracteristic

- graphical plot of **true positive** vs. **false positive** rate
- often used to select optimal models
- a complete random guess would lead to a diagonal line
- perfect models starts in the upper left corner
- goal: minimize area above the ROC curve

Comparsion Theorem

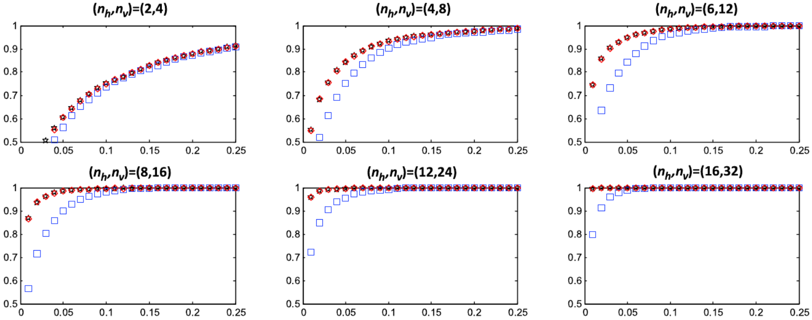


Comparsion Theorem

- different densities for horizontal and vertical direction
- always at the foveal limit (pxiel density goes to infinity)
- obviously, optimal method performs best
- template matching performs not far from optimal with small patch density
- with bigger patch densities, parts-based testing performs better
- nearly optimal with higher densities

Saliency

- look first for parts with strongest evidence, then continue with conditionally most salient part and so on
- extend theorem to an arbitrary number of parts
- **most salient part** $\rightarrow$ the most probable part when only local evidence is taken into account
- if first test succeeds, then compute the most salient part of the **remaining** parts with context that the first part was found
- visiting parts in the order of saliency is better than using a fixed order



- parts-based testing is nearly equivalent to optimal testing

Another example

- create a JPEG image from a books page and search for **at** and **the**
- assuming pixels for each character in a word are independent and identically distributed
- every page is partitioned into blocks containing a character, a symbol or a blank
- each letter has two conditions, **absent** or **present**

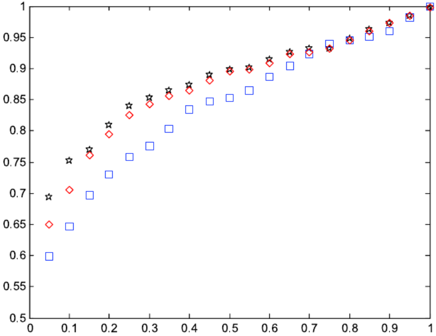
Generalizations

- so far: rigid objects, no or single conjunction
- There can be variations in structure, pose or appearance
- and: presence of the parts does not imply the presence of an object!
- solution: expand latent variable Z from binary ($Z_k \in \{0, 1\}$, absent or present) to n-ary ($Z_k \in \{0, 1, \ldots, n\}$)
- encoding part k is absent or present **and** in a given state.
- states represent a partitioning of pose space and/or a selection of styles or rendering models

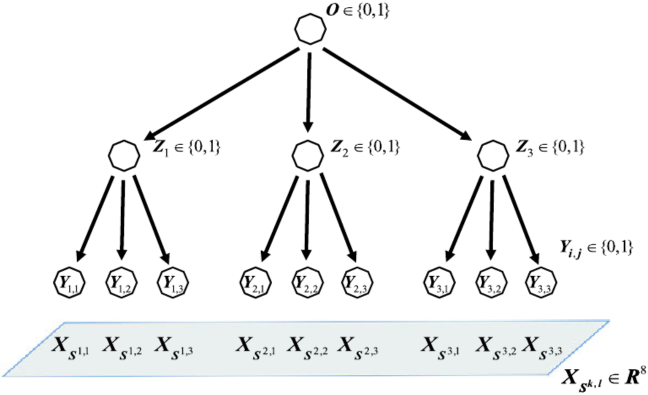
Illustration

- primitive car detector
- collection of 327 car images and 200 natural images without car object
- cars were modeled two have two tires, one in the front, one the back.
- each tire could be at one of five scales and at any position in the image
- a present (sideways) car is detected, when two tires with the same scale were found and depend on the scale-corrected position of one tire with respect to the other

The Problem is small enough and the model simple enough to compute the probabilities for each of the three approaches and the resulting ROC curves:



Instantiation of Object



Conclusion

- nearly optimal detection rate ...
  - for a sequence of local tests for the parts of an object in a high-resolution limit.
  - if tests are ordered by local cond. probabilities of the parts
- dilemma:
  - **when** is a object present and **when** it is absent?
  - absent, if there is nothing which resembles the object.
  - false detection.
- solution: objects are made of componentens of other objects
- hierarchical model for objects in terms of its parts and subparts
- extrem efficient with sequential testing of components (coarse-to-fine search)

Conclusion

- two ways to explore posterior distribution:
  - integration with focus for present objects.
  - identify specific instantiations of objects which are sufficiently likely.
- best strategy: mixture of both. integrate low-level variables but look for specific instantiations of high-level variables.
- but: right computation is much architecture and application specific!