Computer Vision Patch-based Object Recognition (2)

Contents

- Papers on patch-based object recognition
- Previous class: basic idea
- Bayes Theorem: probability background
- Papers in this class
  - Hierarchy recognition
  - Application for contour extraction

Previous class

- What is object recognition?
- Basic idea of object recognition
- Recent research

What is "Object Recognition"?

- Traditional definition

  For an given object *A*, to determine automatically if *A* exists in an input image *X* and where *A* is located if *A* exists.
- Ultimate issue (unsolved)

  For an given input image *X*, to determine automatically what *X* is.

An example of traditional issue

- What is this car?
  - Is this car any of given cars in advance?

Input image

Training images

An example of ultimate issue

- What does this picture show?
  - Street, 4 lanes for each direction, divided road, keeping left, signalized intersection, daytime, in Tokyo,…

Basic idea

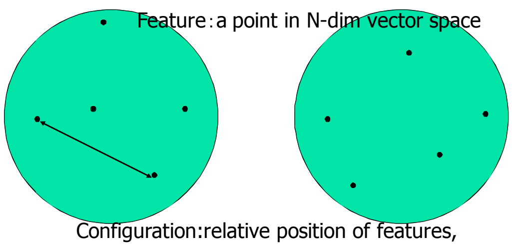
- Make models from training images
- Find closest model for each input image
- You need “good” model
  - Objects are similar, so are models
  - Objects are different, so are models
  - Estimation of similarity is important
  - (More compact models are, better)

Recent models

- Extract a lot of feature patches
- Configuration of the patches makes model
- Why patches?
  - Object might be occluded
  - Location of object is unknown
  - No complete match in class recognition:
    - Similarity among patches is easier
- Number is power

Patch-based models

- Local features and its configuration

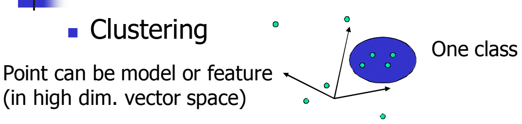


Class and Specified object

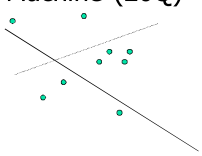
- Recognition of specified object(s)
  - Model is different from any other objects
- Class recognition
  - Model is “similar” among objects in the same class
  - All objects in a class are not given

Model in class recognition

- Clustering



- Support Vector Machine (20Q)



Similarity Estimation

- Easy to estimate
  - Images of the same dimension
  - Points in the same vector space
- Hard to estimate
  - Patch-based models
  - Parts of images

How to Estimate Similarity

- Distance (or correlation)
  - Points in a vector (metric) space
  - Distance is not always euclidian
- Probability
  - Clustering can be parameterized with pdf
  - SVM, answer for H>0 can be probability

Recognition with probability?

- Assume an input image is given
- Does a car exist in the image?
  - For human: easy to answer: Yes or No.
  - For computer: might be hard to answer, but the answer should be yes or no!
- Why you can apply probability for yes-no question?

Posterior probability

- Situation
  - You have just rushed on Chuo line train at Ochanomizu stn for Shinjuku direction.
  - It is not crowded.
  - Is it special rapid train?
- Discussion
  - There is the timetable, the answer is known.
  - If you don't know it, what will your answer?

Background

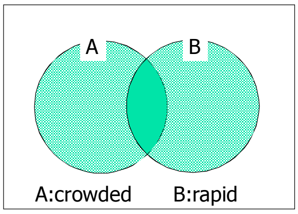
- Any Chuo line train is rapid or special rapid
- You have no idea on which train you get on
- Special rapid train is more crowded than rapid train
- So you can say, "If I bet, I prefer rapid train"
- If odds is 1-2, do you bet?

Estimation

- Assume the followings are known
  - Pr(train is special rapid)
  - Pr(special rapid is not crowded)
  - Pr(rapid is not crowded)
- You can calculate the probability that your train is actually rapid.

Bayes Theorem

- $P(A \cap B)=P(B|A)P(A)=P(A|B)P(B)$
- $P(B|A)=P(A|B)P(B)/P(A)$



Even if B happens prior than A, P(B|A) can be calculated

Answer for the example

- A:train is rapid
- B:train is not crowded
- P(A|B): Prob. of no-crowded train is rapid
- P(B)
  =(prob. of rapid train is not crowded)
  +(prob. of special rapid is not crowded)
- P(B|A)=(prob. of rapid train is not crowded)
- P(A|B)=P(B|A)P(A)/P(B) … can be calculated

For example…

- Assume special rapid runs 0,20,40 and rapid runs 10, 30, 50; P(A)=0.5, $P(A^c)=0.5$
- P(rapid is not crowded)=P(B|A)=0.7
- P(special rapid is not crowded)=$P(B|A^c)=0.2$
- P(train is not crowded)=P(B)=$P(A\cap B)+P(A^c \cap B)$
  $P(B|A)\,P(A)+ P(B|A^c)\,P(A)$=0.7x0.5+0.2x0.5=0.45
- P(rushed train is rapid if it is not crowded)=P(A|B)

=P(B|A)P(B)/P(A)

=(0.7x0.45)/0.5=0.63

Essence

- What you can investigate in advance is:
  probability that train is not crowded when it is rapid or special rapid
  (general theory)
- What you like to know is:
  probability that your train is rapid or not when it is not crowded
  (special case estimation)

Apply for object recognition

- What you know in advance are:
  the models of objects *Xi* (might be class) will be like this if *Xi* appears in given images
- What you like to know is:
  The object X appears in this given image if models of the possible objects in it are like this

How to apply

- $X_1, X_2, \dots, X_n$: Objects to be recognized
- I: Input image

- Now you have I, are there any X_i in I?
  P(X_i exists |I is observed)

$\propto$P(I is observed | X_i exists)P(X_i exists)

$\propto$P (I is observed | X_i exists) (if P(X_i exists)can be considered to be constant for all i)

First paper

- **Semantic Hierarchies for Recognizing Objects and Parts**
  - Boris Epshtein Shimon Ullman
  - *Weizmann Institute of Science, ISRAEL*
  - *CVPR 2007*

Abstract

- Patch-based class recognition
- Hierarchy
- Automatic generation of hierarchy from images
- Experiment

Hierarchies (Face case)

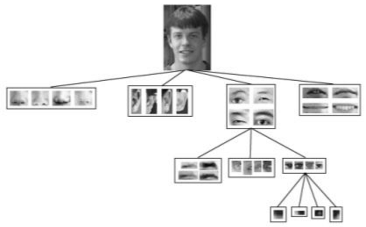
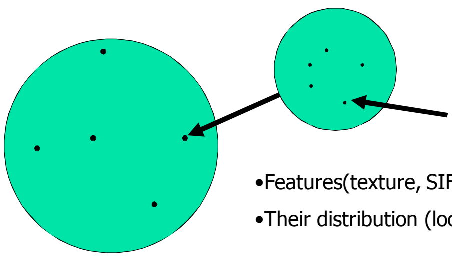


Figure 1. Schematic illustration of a semantic hierarchy. A face is represented as a combination of parts and sub-parts. Each part is represented as a semantic equivalence set of different possible appearances. The proposed scheme is the first to extract and use semantic parts in feature hierarchies.

Model



•Features(texture, SIFT…)
•Their distribution (location)

Hierarchies (Theory)

- Tree diagram
- Classification and parts (patches)
- How to construct hierarchies
- Training method

Tree diagram

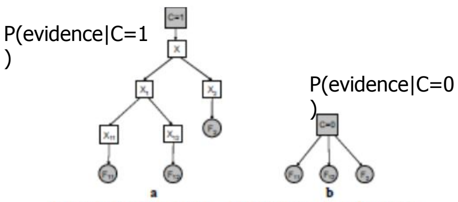


Figure 2. (a) Class model (b) Non-class model. F_i are the observable features, X is the entire object, X_i are object parts, and C is the class node. During recognition, the features F_i are observed in the images, and the computation infers the most likely values of X, X_i.

Class Model

- Class X consists of Xi, Xij, Xijk …
- Each X_I has $A(X_I)$, $L(X_I)$
  - $A(X_I)$: view of X_I
    ex) open mouth if 1, closed mouth if 2,….
  - If X_I is an end, $A(X_I)$ corresponds to some image feature F_I
  - $L(X_I)$: location of X_I
    $L(X_I)=0$ means X_I is occluded

End of tree diagram

- If X_I is an end, $A(X_I)$ corresponds to some image feature F_I
- X_I, F_I consists of NxK components (S[1,1],…,S[1,N],…,S[K,1],…,S[K,N]), where i in S[i,j] corresponds to view change of XI, j to its location
- For each i,j , give similarity of F and X

What we have to do

- {F}: Features in an input image
- p(X|{F}) is what we like to know:
  - Larger it is, more assured object *X* is
- $P(X|\{F\})=P(\{F\}|X)P(X)/P(\{F\})$

  $\propto P(\{F\}|X)P(X)$

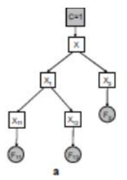
Calculate P(X), P({F}|X)

Basic relation

- From construction of tree diagram,

$P(X,\{F\})=p(X)\Pi p(X_i \mid X_{i\sim})p(F_k|X_k)$ …(1)

($X_{i\sim}$ is the parent of X_i)



Calculation of P(X)

- P(A(X)=a, L(X)=l)

Probability of Object *a* is located at *l*

- Assume this distribution is uniform
- In the case of ID photo, *l* is not uniform at all, but in this paper, assume this.

$P(F_i|A(X_i)=a,L(X_i)=l)$ part 1

- Prob. Of feature Fi is observed when Xi looks like *a* and located *l*
- F=(S[1,1],…,S[N,K])

$P(F_i|A(X_i)=a,L(X_i)=l)$

$=p(S[1,1],…,S[N,K] \mid A(X_i)=a,L(X_i)=l)$…(2)

$=\Pi p(S[k,n] \mid A(X_i)=a,L(X_i)=l)$

- Assume S[i,j] are independent

$P(F_i|A(X_i)=a,L(X_i)=l)$ part 2

- View and location are independent
  - Ph(S[a]): harmony with a
  - Pm(S[a]): missharmony with a

$p(S[1,1],…,S[N,K] \mid A(X_i)=a,L(X_i)=l)$

$=ph(S[a,l])\Pi Pm(S[k,n])$ $(k\neq a,n\neq l)$…(3)

$p(S[1,1],…,S[N,K] \mid L(X_i)=0)$;can't be seen

$= \Pi Pm(S[k,n])$ …(4) ; independent with a

$P(F_i|A(X_i)=a,L(X_i)=l)$ part 3

$P(F_i|A(X_i)=a,L(X_i)=l)$
$\propto P(F_i|A(X_i)=a,L(X_i)=l) / P(F_i|L(X_i)=0)$ …(5)
$=ph(S[a,l])/Pm(S[a,l])$

$p(A(X_i),L(X_i)|A(X_{i^\sim}),L(X_{i^\sim}))$

$p(X_i \mid X_{i^\sim})$ is still unknown in
$P(X,\{F\})=p(X)\Pi p(X_i \mid X_{i^\sim})p(F_k|X_k)$ …(1)

- View and location are independent

$p(A(X_i),L(X_i)|A(X_{i^\sim}),L(X_{i^\sim}))$
$= p(A(X_i)|A(X_{i^\sim}))p(L(X_i),L(X_{i^\sim}))$ …(6)

- Calculate 1st term and 2nd term

$p(A(X_i)|A(X_{i^\sim}))$

- Probability of what children can be if the parent is known
- No theoretical method; determine through training (explain later)
- Can be calculated in advance

$p(L(X_i),L(X_{i^\sim}))$

- Probability of child location when parent location is known
- When $L(X_{i^\sim})=0$ (The parent can't be seen)
  - Uniform: $P(L(X_i)=l, L(X_{i^\sim})=0) = \delta_0/K$
  - $P(L(X_i)=0, L(X_{i^\sim})=0) = 1-\delta_0$
- $L(X_{i^\sim}) \neq 0$
  - $P(L(X_i)=0, L(X_{i^\sim})=L) = 1-\delta_1$
  - Gaussian: $P((L(X_i)=l, L(X_{i^\sim})=L)$ is determined as normal distribution of l
- These parameters are determined throughout training

Classification and parts

- Estimating $p(C=1|F)$
  - $P(C=1|F)/p(C=0|F)$
    $=P(F|C=1)P(C=1)/(P(F|C=0)P(C=0))$
    $\propto P(F|C=1)/P(F|C=0)$
- Bottom up
- Top down

Bottom up

- $P(F|C=0)$ is constant.
- $P(F|C=1)$ can be calculated by bottom-up method
- F(Xi):evidence of subtree under node Xi

$$p(F(X_i)|X_i=k)= \\ =\prod_{X_{i'}}\left(\sum_t p(F(X_{ij})|X_{ij}=t)p(X_{ij}=t|X_i=k)\right) \quad (8)$$

Top-down

- In bottom-up method, all probability of edges in tree diagram is calculated
- Now P(X,F)can be calculated, thus

$$D(\underline{X}) = \arg\max_{\underline{X}} p(\underline{X}, \underline{F} \mid C=1) \qquad (9)$$

can be calculated by top-down method

Hierarchic structure

- Simple hierarchy (from one image)
- semantic hierarchy (add images)
- Any node can be hierarchic if necessary

Example

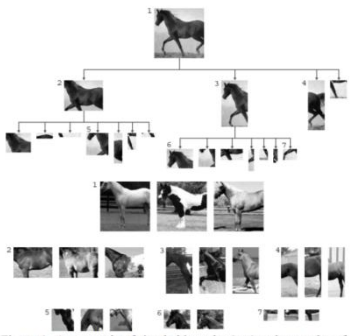


Figure 6. An example of simple hierarchy (top) and examples of additional semantic features at different levels of the semantic hierarchy.

Example of hierarchic structure

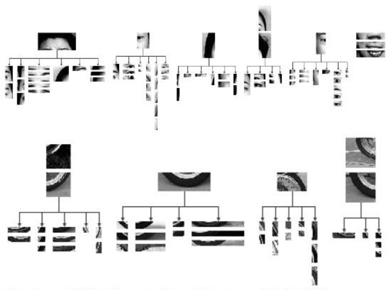


Figure 7. Additional examples of the semantic hierarchies.

Simple hierarchy

- Make node where a lot of features appear
- Use one image or a few images

Semantic nodes (1)

- T={Tn|n=1,2,…} Training images

  Make semantic nodes from training images
- For each Tn, calculate

  H(X)=D(X)=arg max p(X,F|C=1)
- L(Xi)=0 or probability is small but L(Xi~)≠0,

  L(Xi)=arg max p(L(Xi)|L(Xi~))

  A(Xi) is the one located at L(Xi)

Semantic nodes (2)

- Repeat previous step
- For each node, there become a list of "unseen views"
- Remove isolated unseen views (such that there are no similar views around it)
- For each node, find "effective" new views and add them as views

Semantic nodes (3)

- As adding new views, nodes can be hierarchies
- Even some views can be similar, hierarchies can distinguish each other

Training

- Determine the parameters
- Initialize
  - Location: distance between the parent and a child is in simple hierarchy, variance is half of the distance
  - δis 0.001
  - P(A(Xi)|A(Xi~)) is determined by counting
- For each training image, find H(X) and optimal {Xi}, and tune parameters
- Repeat this

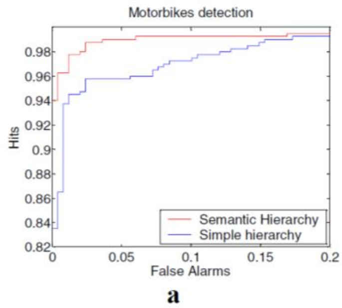
Experiment

- Class recognition
- Parts detection

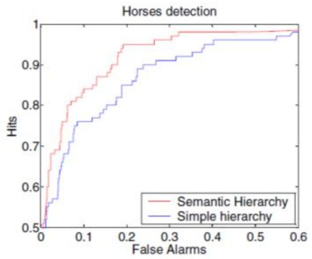
Class Recognition

Figure 3. Examples of class images. Rows, from top to bottom: Horses, motorbikes, cars, JAFFE dataset.

Result (motorbikes)

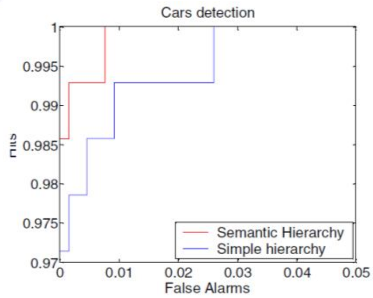


Result (Horses)



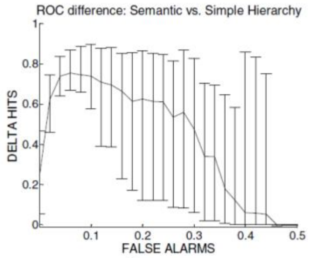
b

Result (Cars)



c

Result



d

Parts Detection

Figure 5. Variability of the object parts detected by the semantic hierarchy.

Result (Parts detection)

Part	Simple hierarchy	Semantic hierarchy
	39.1%	25.7%
	41.9%	36.0%
	51.8%	19.2%
	4.87%	4.3%
	18%	3.4%
	28.85%	5.14%
	0.93%	0%
	22.5%	4.2%
	3.2%	0%

Table 1. Percentage of incorrectly detected or missed parts.

Summary

- Semantic hierarchies
  - Recognize a lot of parts
  - Parts can be hierarchical if it becomes too complicated
- Better than simple hierarchies
- Hierarchies are automatically generated even in complicated cases

Final paper

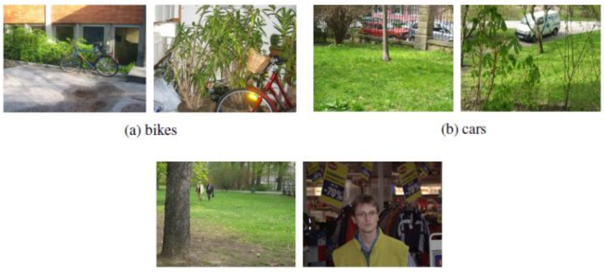
- **Accurate Object Localization with Shape Masks**
  - Marcin Marszaek Cordelia Schmid
  - *INRIA, LEAR - LJK*
  - *CVPR 2007*

Abstract

- Extract shape of an object class
  - "spin-off" method for class recognition
  - Robust against bad images
- Make mask image from an input image
  - Mask image consists of not 0, 1 but probability (0.0-1.0)

Aim

Examples of input images



(a) bikes (b) cars

(c) people

Contents

- Technique
  - Distance between masks
- Framework
  - Training method
  - Recognition method
- Experiment
- Conclusion

Technique

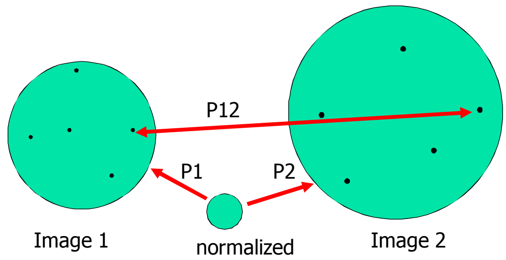
- Local feature and localization
  - Local feature
  - Localization with features
- Mask
  - Similarity of mask images
- Classification of masks using SVM

Local feature and localization

- Local features
  - Invariant against translation, rotation and/or scale
  - Scale invariant and normalization
- Localization using local features
  - Local feature θ in image 1 and 2 are similar
  - p1: normalized translation of feature θ in image 1
  - p2: normalized translation of feature θ in image 2
  - Localization between two images: $p12=p1^{-1}\ p2$

Localization

- P12: left to right (scale-up and translation)



Shape mask similarity

- Similarity between binary masks

$$o_b(Q_b, R_b) = \frac{|Q_b^1 \cap R_b^1|}{|Q_b^1 \cup R_b^1|} = \frac{\sum \min(Q_b, R_b)}{\sum \max(Q_b, R_b)} \quad (3)$$

- Similarity between probability masks

$$o_s(Q, R) = \frac{\int \min(Q, R)}{\int \max(Q, R)} \quad (4)$$

$$= \frac{C}{\int Q + \int R - C}, \; C = \int \min(Q, R) \quad (5)$$

- Localized similarity

$$o_f(i, j) = o_s(\zeta_i \circ P_{ij}, \zeta_j) = o_s(\zeta_i, \zeta_j \circ P_{ji}) \quad (6)$$

Mask classification using SVM

- Classify the view in the shape
- Inside→$H_i=\{H_{ij}\}$, H_{ij}=#of feature j
  - Any feature is one of v features
  - V-dim vector for each image
- Hi's can be classified with 20Q method
  - SVM(Support Vector Machine)
  - Automatically generate "good" questions

Mask classification using SVM

- Distance(similarity) between Hi and Hj is defined as follows

$$K(H_i, H_j) = e^{-\frac{1}{A} D(H_i, H_j)} \quad (7)$$

$$D(H_i, H_j) = \frac{1}{2} \sum_{n=1}^{V} \frac{(h_{in} - h_{jn})^2}{h_{in} + h_{jn}} \quad (8)$$

Where A is average of all D(Hi,Hj)

End of technique

- Similarity between two shape masks
- Similarity between two views in shape mask
- Make training and recognition

Framework

- Training
- Recognition

Training procedure

Find similar pairs

Merge similar features

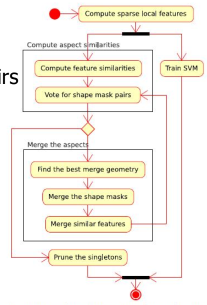


Figure 2. Overview of the training procedure. The main operation blocks are executed iteratively.

1.Feature extraction

- Any feature can be one of V features
- In training, object area is known
  - Features outside of shape is ignored
  - For each feature *i* in the shape is recorded along with normalized parameter *pi*

2. Similarity

- Two masks are similar if
  - Shape masks are similar
  - Local features with their location are similar
- More precisely,
  - If local feature *i* in image 1 and local feature *j* in image 2 is similar, localize two image with P*ij*
  - Similar if mask simlarity ≧0.85
  - Try all combination of similar local features

3.Voting shape masks

- Method 2 takes lots of time
- For any pair (x,y) of shape masks,
  - Vote 1 to point (x,y) if they are similar for some p*ij*
  - Vote will be large if local features with their location are similar
- Merge closest pair (x,y) (explain later)
- Repeat until no more merging

Key point of the vote

(a) Hypothesis evaluation

4.Location of merged mask

- New location of the mask merged with two masks
- For all pairs (i,j) of the same feature,
  - Localize two masks using Pij
  - Calculate similarity as follows

$$o_f(i,j) = o_s(\zeta_i \circ P_{ij}, \zeta_j) = o_s(\zeta_i, \zeta_j \circ P_{ji}) \qquad (6)$$

- Pij: (i,j)=arg max o_f(I,j) is determined

5.Merge shape masks

- Merge to "larger" mask
- Localized two images with Pij
- Merge weighted average
  - No detail is described, but probably depending on the number of masks merged before, merging will be executed.
- View of the new shape mask is changed, hence, shape mask distance from the new shape mask is re-calculated

6.Merging local features

- Local features are also merged
- Local features in the shape will be similar
- Local features are merged with the same way as local shape (weighted average)
- Repeat until merging can be

7. Remove singleton

- Singleton: after merging procedure, image *X* is not merged with any other images, then *X* is called a singleton
- This kind of image might be an outlier hence we remove all singletons

8. Training SVM

- SVM is also trained
- SVM is trained for each object class
  - Should be trained for each view
  - Number of each view was small

Recognition

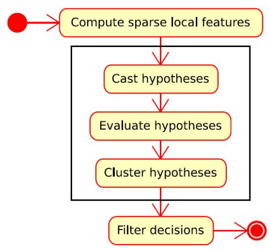
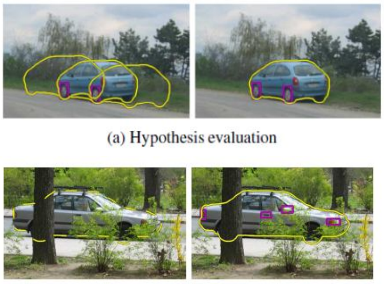


Figure 4. Overview of the recognition procedure. The main operation block is executed in a pipe to reduce memory requirements.

Recognition framework



(a) Hypothesis evaluation

(b) Evidence collection

1.Local features

- Extract local features from an input image
- Any feature is assumed as one of V-features

2. Hypothesis

- Local feature i in an input image
- Local feature j in an trained mask
  - Localize Pij
- Hypothesis appears that a mask is located at some location
- Too large number of hypothesis!

3.Hypothesis evaluation

- H can be calculated in the shape area
- H is also classified with SVM
- Confidence is calculated

Hypothesis evaluation

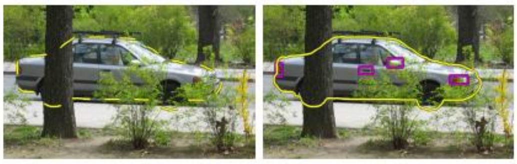
(a) Hypothesis evaluation

4. Cluster Hypothesis

- Occlusion decreases confidence
  - View and location of local feature is used
- Lots of shape mask hypothesis
  - Necessity of clustering
- Similar hypothesis should be clustered
  - New mask depending on confidence

$$S = \frac{w_Q}{w_Q + w_R}\cdot Q + \frac{w_R}{w_Q + w_R}\cdot R \qquad w_S = w_Q + w_R \quad (6)$$

Evidence collection



(b) Evidence collection

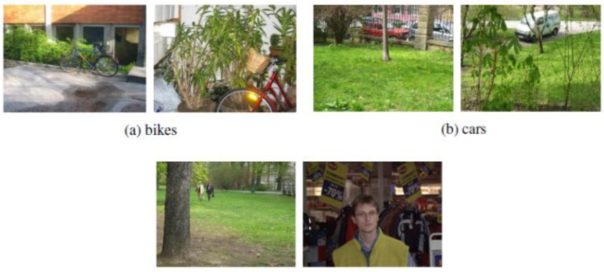
5.Decision

- To decrease false Positive
- Assume that there is only outside occlusion
  - No self-occlusion
- No detailed description
- Not only confidence, but also accept hypothesis whose confidence is spread into whole mask

Experiment

- Graz-02 dataset
- Effect of aspect clustering
- Comparison with Shotton's method

Examples of Graz-02 dataset



(a) bikes (b) cars (c) people

Recognition Result

object class	cars	people	bicycles
no hypothesis evaluation	40.4%	28.4%	46.6%
no evidence collection	50.3%	40.3%	48.9%
our full framework	**53.8%**	**44.1%**	**61.8%**

Table 1. Pixel-based RPC EER measuring the impact of hypothesis evaluation and evidence collection.

Extracted Shape Masks

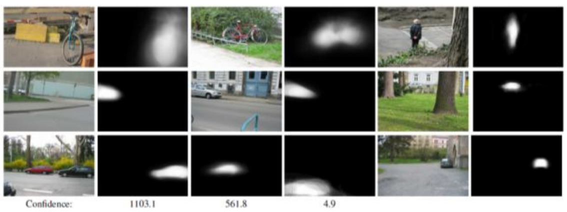


Figure 7. Results on Graz-02 dataset. Note the precise object shape estimations despite occlusions and background clutter. Multiple object instances are detected with subsequent hypotheses as is shown in the bottom row (4 left most columns).

Clustering sample

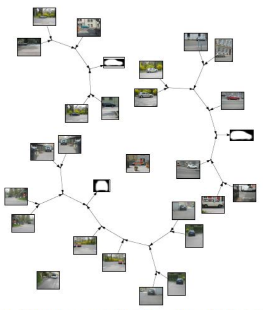
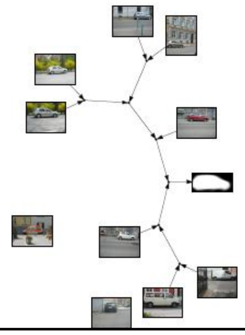


Figure 5. Several car aspects detected by agglomerative clustering.

Right-hand side



Effect of aspect clustering

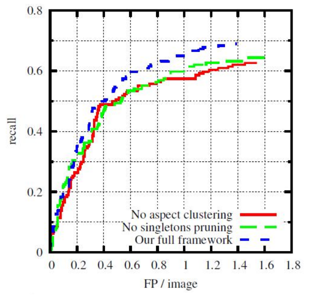


Figure 6. Recognition rate for cars given as recall in a function of FP per image. We can observe the impact of aspect clustering.

Comparison (Houses)

Shotton [21]	92.1%
Our framework ($T = 0.85$, with singletons)	**94.6%**
Our framework ($T = 0.7$, no singletons)	**94.6%**

Table 2. RPC EER for Weizmann horse dataset.

Extracted Shape (Houses)

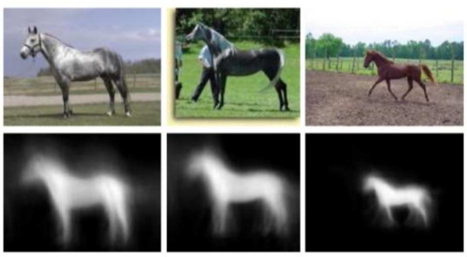


Figure 8. Results on Weizmann horses dataset. Note that the shape masks are very accurate: the horse articulations are visible.

Summary of this paper

- Global feature: Shape mask
- Local feature: view of features

- Generation of class mask
- Good result for clean images

Conclusion

- Class recognition from still image
- Model of view, location and similarity
  - View similarity, location similarity
  - View similarity can be clustered
- Bag of features
- Comparison with 20Q: Number is power
  - Intersection of many features is unique
  - Probability is used for similarity instead of yes, no