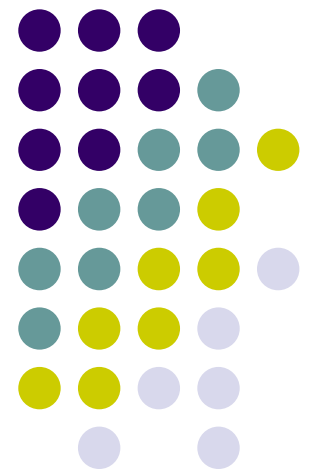


Texton and Histogram I

Hao Tang

Based on

T. Leung and J. Malik, "Representing and
Recognizing the Visual Appearance of
Materials using Three-dimensional Textons,"
IJCV 01





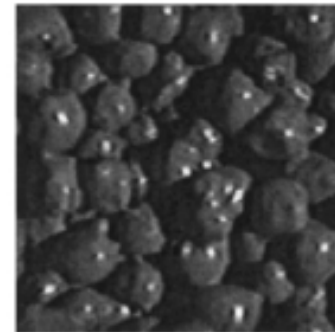
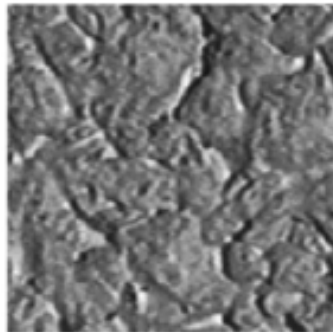
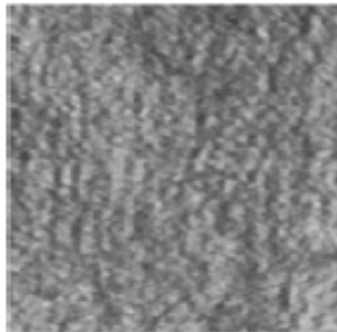
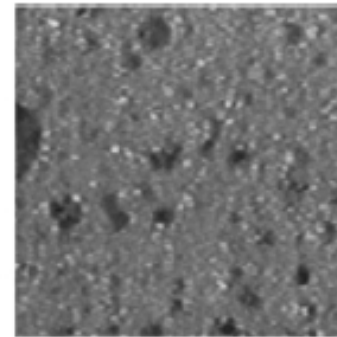
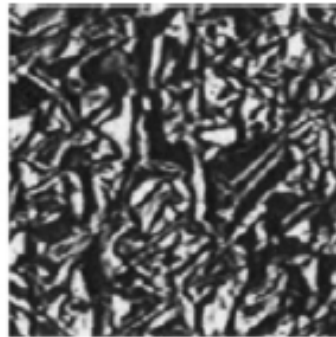
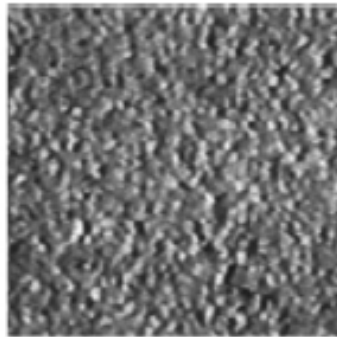
Contributions

- Presents a framework for representing natural textures with 3D textons
- Presents an algorithm for recognizing 3D textures from a single image under ANY lighting and viewing direction
- Shows how 3D texton representation can be used to predict the appearance of natural materials under novel illumination condition and viewing geometry



What is texture?

- Entities with some spatially repeating properties
- Different materials show different texture appearance





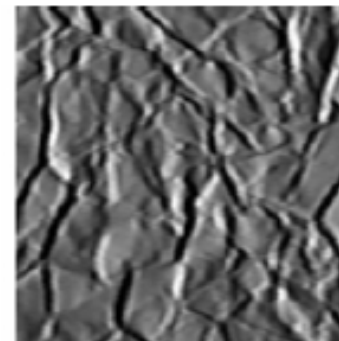
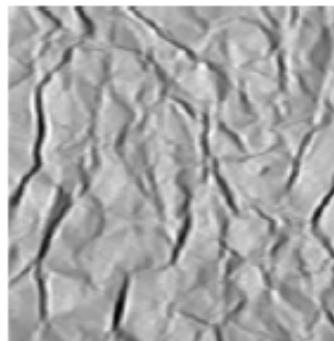
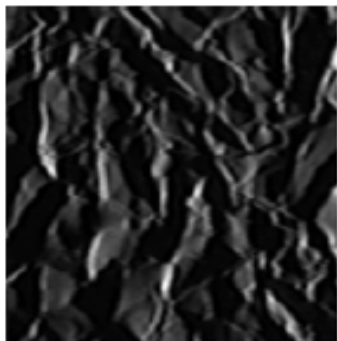
2D Texture

- “Flat” texture
- Viewpoint and illumination are assumed constant
- Surface normal variations are ignored
- Representative techniques:
 - Markov Random Field
 - Filter responses



3D Texture

- Nature shows an abundance of “relief” textures
- Particularly due to surface normal variations (specularities, shadows, and occlusions)
- Appearance changes dramatically due to different viewpoint/lighting settings





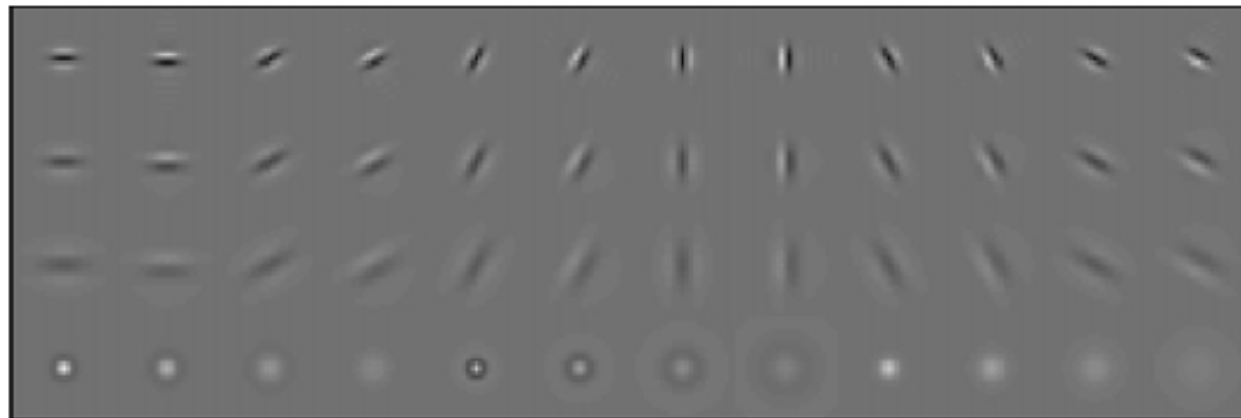
The Theme

- We want to recognize surfaces made from different materials on the basis of their texture appearance
- Main idea: represent a surface of any material with a spatial arrangement of a small number of perceptually distinguishable micro structures – 3D textons
- The goal is to build a small, finite vocabulary of 3D textons

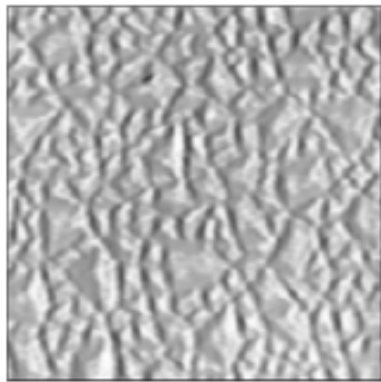
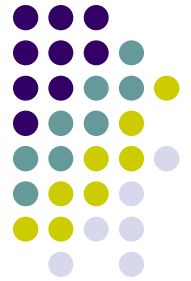


2D Textons

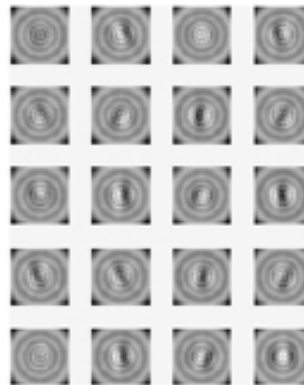
- Apply a set of orientation and spatial-frequency selective linear filters (a 48-filter filterbank) to each pixel on the surface
- Cluster the filter responses into a small set of 48-dimensional prototype vectors using the K-means algorithm
 - K-mean centers: 2D textons
 - Associated filter responses: appearance vectors



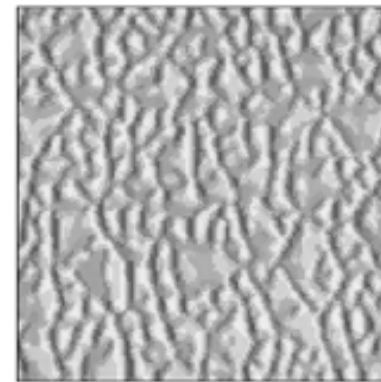
Reconstruction from Textons



(a)



(b)



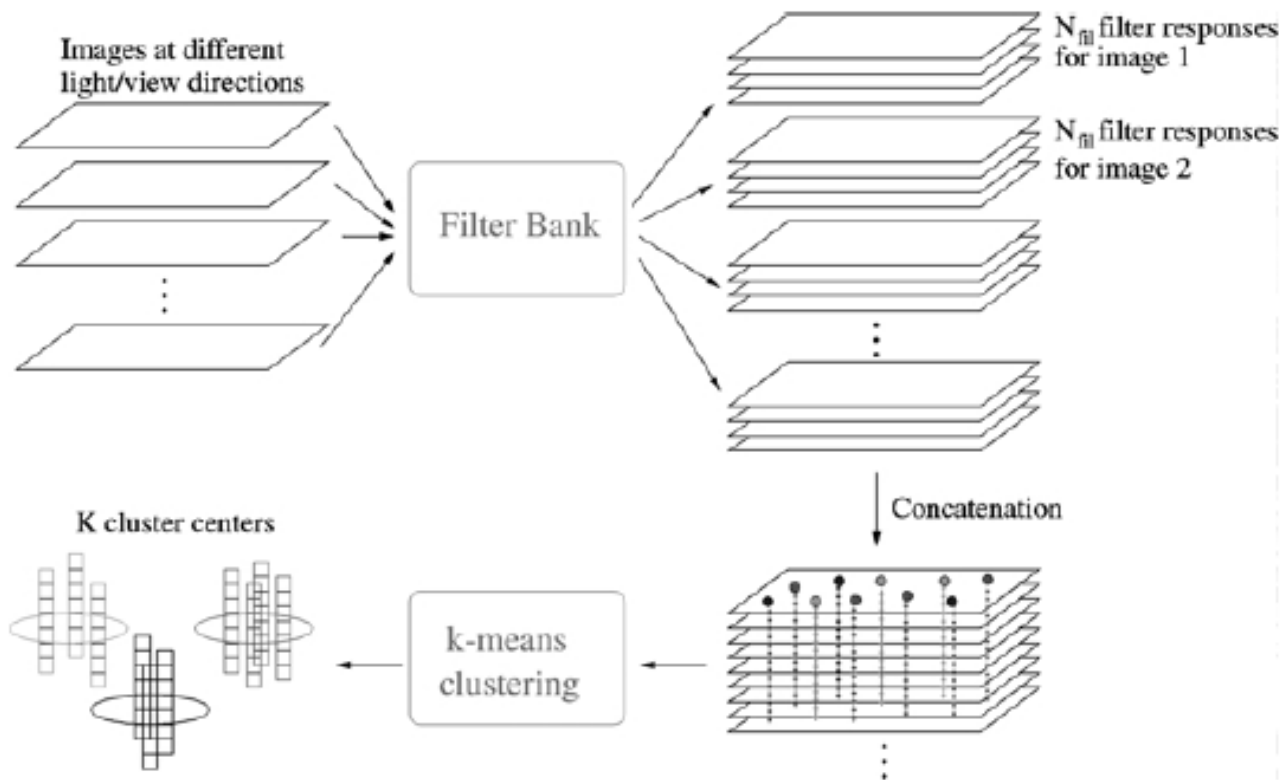
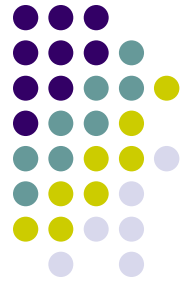
(c)



3D Textons

- Many images at different lighting and viewing directions are needed
- Apply the filterbank to each pixel on each of the many images under different lighting and viewing directions and concatenate the filter responses to a long vector
- It is believed that this long vector will encode the appearances of dominant features in the image under all lighting and viewing conditions

3D Texton Vocabulary

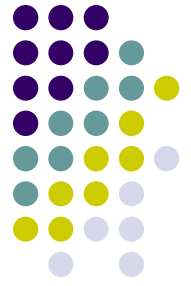




Building Procedures

1. For each of the 20 training materials, the filter bank is applied to each of the $N_{vl} = 20$ images under different viewing and lighting conditions. The response vectors at every pixel are concatenated together to form a $N_{fil}N_{vl}$ vector^T.
2. For each of the 20 materials individually, the K -means clustering algorithm is applied to the data vectors. The number of centers, denoted by K , is 400. The K -means algorithm finds a local minimum of the following sum-of-square distance error:

$$Err = \sum_{i=1}^N \sum_{k=1}^K q_{ik} \|\mathbf{x}_i - \mathbf{c}_k\|^2$$



Building Procedures

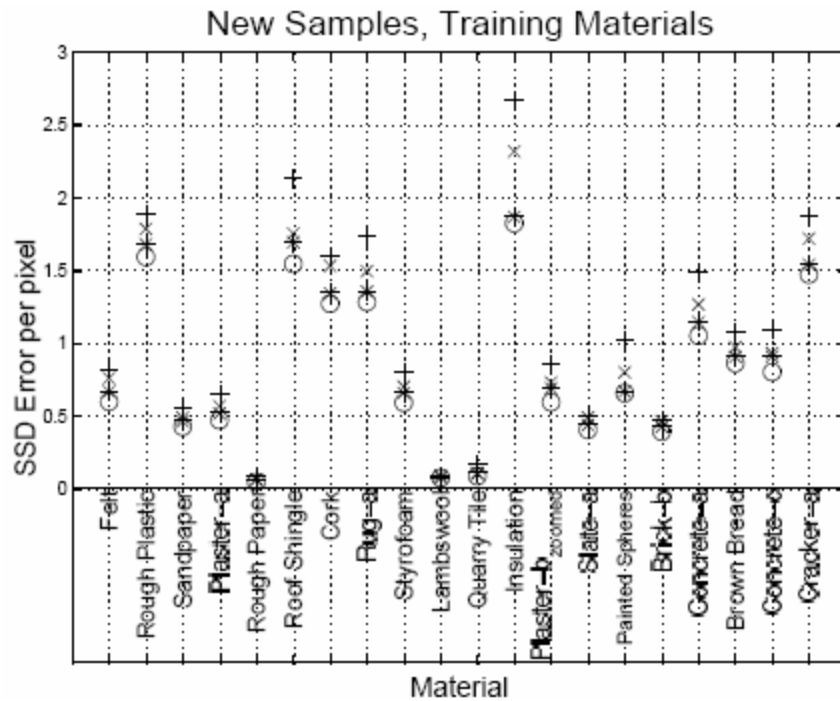
where

$$q_{ik} = 1 \quad \text{if } \|\mathbf{x}_i - \mathbf{c}_k\|^2 < \|\mathbf{x}_i - \mathbf{c}_j\|^2 \\ \forall j = 1, \dots, K \text{ and } j \neq k$$
$$q_{ik} = 0 \quad \text{otherwise}$$

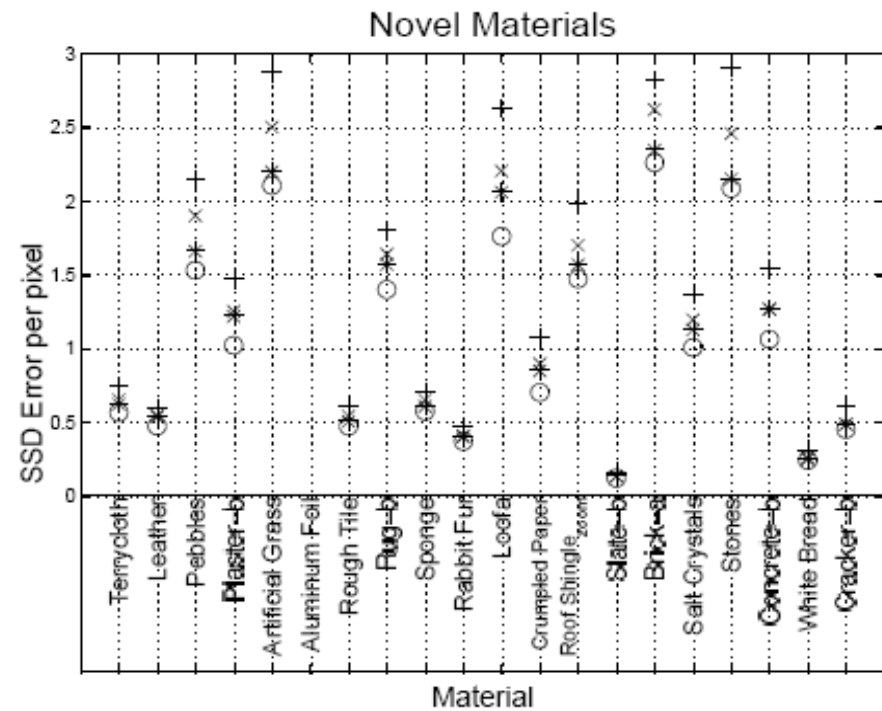
N denotes the number of pixels; \mathbf{x}_i is the concatenated filter response vector of the i^{th} pixel and \mathbf{c}_k is the appearance vector for the k^{th} center. The K -means algorithm is initialized by random samples from all the data vectors.

3. The centers for all the materials are merged together to produce a universal alphabet of size $K = 8000$.
4. The codebook is pruned down to $K = 100$ by merging centers too close together or getting rid of those centers with too few data assigned to them⁸.
5. The K -means algorithm is applied again on samples from all the images to achieve a local minimum.

Evaluation of 3D Textures



Expressiveness



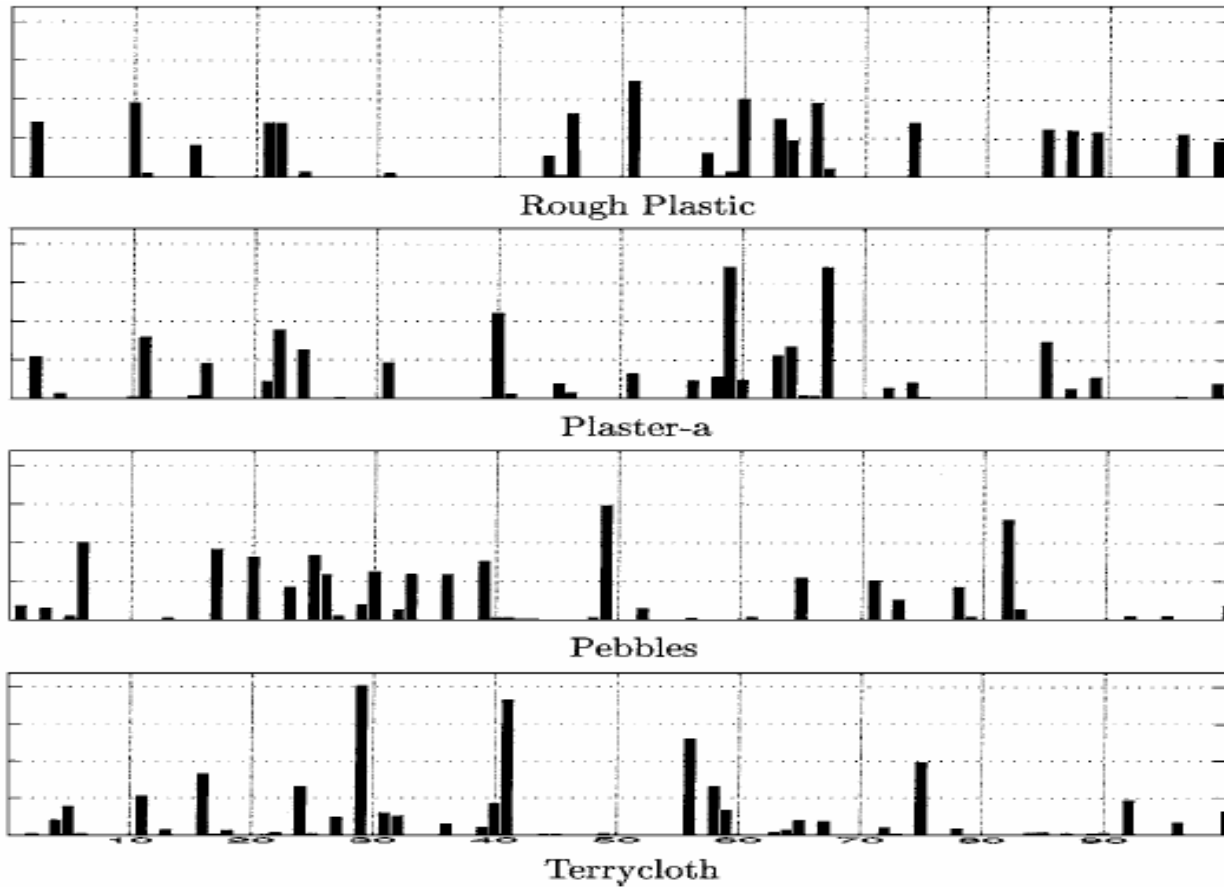
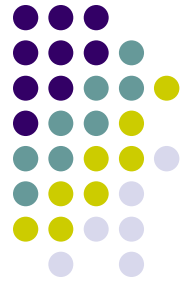
Generalization

Representation by 3D Textons



- The surface of any material is represented as a spatial arrangement of symbols from the 3D texton vocabulary
- A model of each material can be acquired
- Material recognition can be performed based on the histogram of the 3D textons

Histogram-based Recognition





Histogram-based Recognition

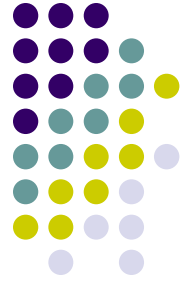
- Chi-square significance test

$$\chi^2(h_1, h_2) = \frac{1}{2} \sum_{n=1}^{\#bins} \frac{(h_1(n) - h_2(n))^2}{h_1(n) + h_2(n)} \quad (1)$$

- Chi-square probability function

$$P(\chi^2|\nu) = Q(\nu/2, \chi^2/2) \quad (2)$$

$$Q(a, x) = \frac{1}{\Gamma(a)} \int_0^x e^{-t} t^{a-1} dt$$



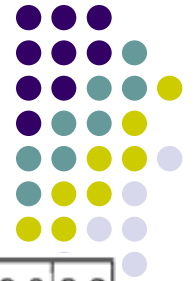
Texture Recognition

- 3D texture recognition from multiple viewpoint/lighting images
 - Straightforward...
- 3D texture recognition from a single image

Results: Multiple Images

- Recognition rate: 95.6%

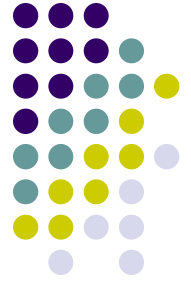




Results: Multiple Images

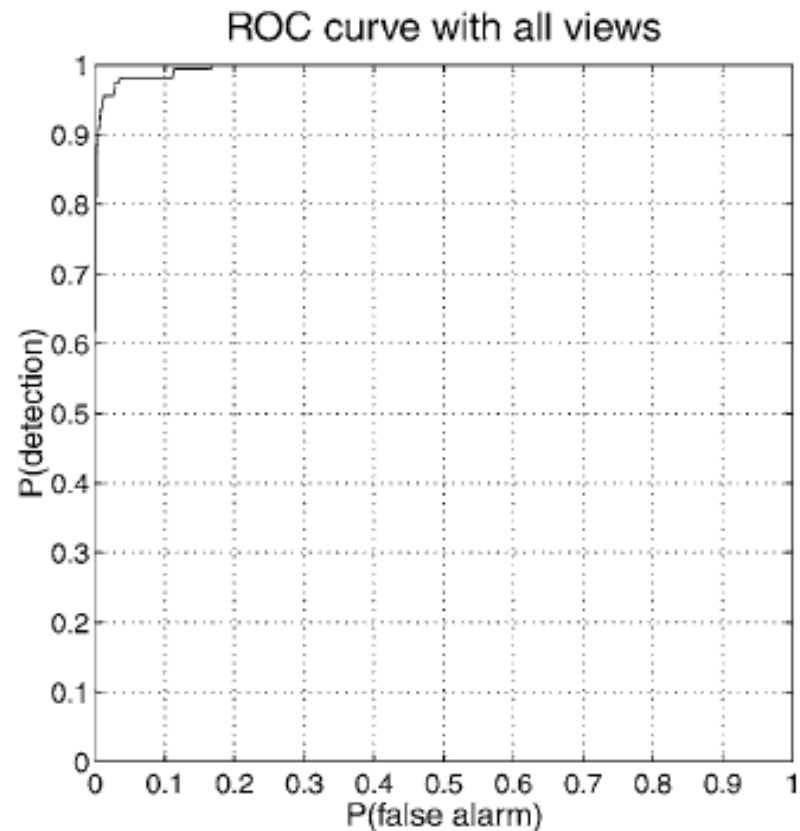
Similarity matrix

Felt	0.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Terrycloth	0.0	1.0	0.0	0.0	0.3	0.0	0.1	0.2	0.0	0.0	0.0	0.0	0.0	0.0
Rough Plastic	0.0	0.0	0.9	0.0	0.0	0.0	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0
Leather	0.2	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Sandpaper	0.0	0.1	0.0	0.0	1.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Pebbles	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0
Plaster-a	0.0	0.1	0.2	0.0	0.1	0.0	1.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0
Plaster-b	0.0	0.2	0.1	0.0	0.0	0.0	0.8	1.0	0.0	0.0	0.0	0.0	0.0	0.0
Rough Paper	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.0	0.0	0.0
Artificial Grass	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.1	0.1	0.0	0.0
Roof Shingle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	1.0	0.1	0.0	0.0
Aluminum Foil	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	1.0	0.0	0.0
Cork	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.2
Rough Tile	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.9
	Felt	Terrycloth	Rough Plastic	Leather	Sandpaper	Pebbles	Plaster-a	Plaster-b	Rough Paper	Artificial Grass	Rough Shingle	Aluminum Foil	Cork	Rough Tile

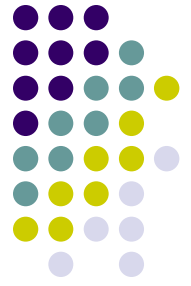


Results: Multiple Images

- Receiver Operation Characteristics (ROC) Curve



Recognition from Single Image



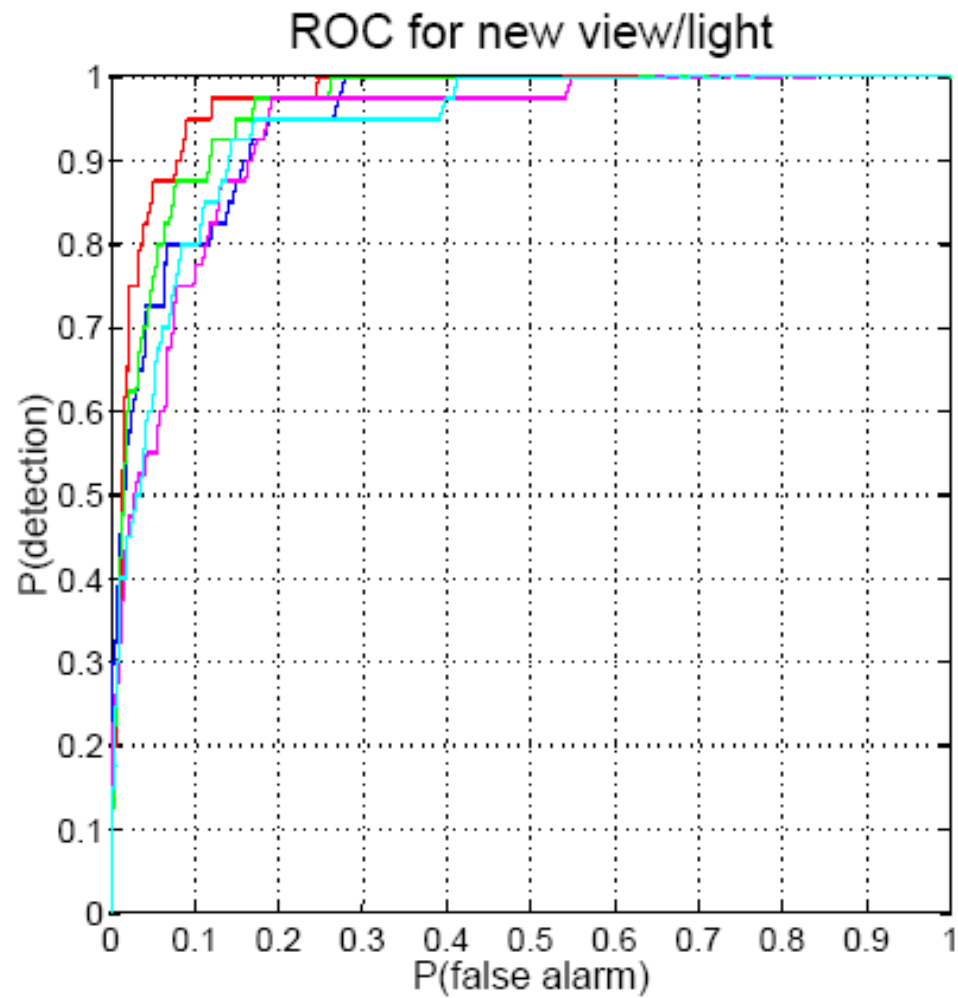
- Texture recognition based on a single image is generally very hard, because finding the texton label for each pixel is difficult
- If we know the texton labels for each pixel, the material can be reconized; if we know the material identity, the texton labels can be assigned
- “Chicken-and-egg” problem



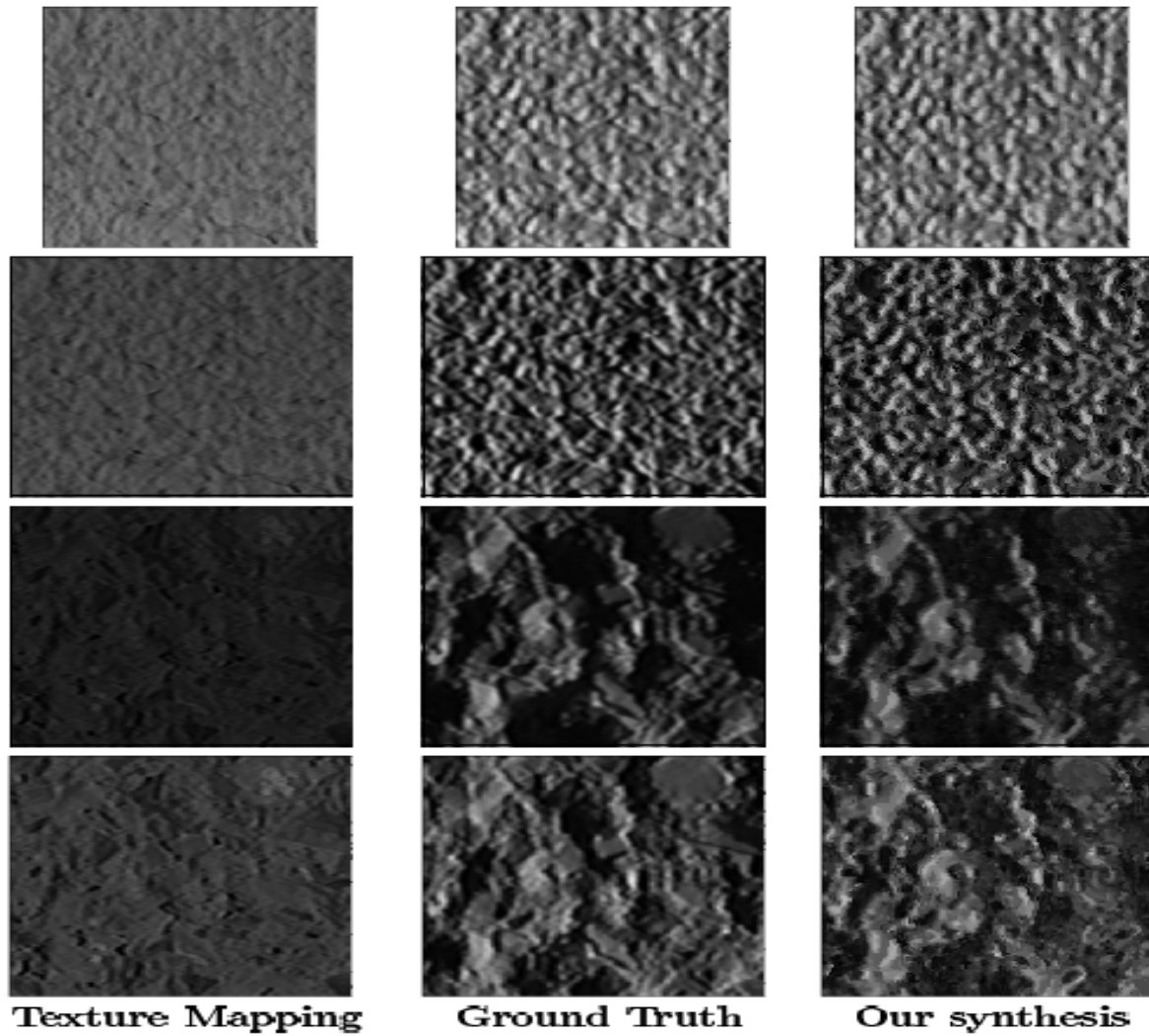
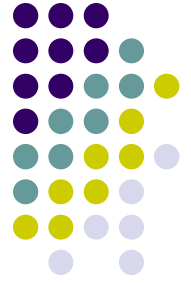
Recognition from Single Image

- Solved by Markov Chain Monte Carlo (MCMC) Algorithm
 1. Randomly assign a label to each pixel i among the N_i possibilities. Call this assignment the initial state $x^{(t)}$ with $t = 0$;
 2. Compute the probability of the current state $P(x^{(t)})$ using Equation 2 with h_{τ_i} as the model histogram;
 3. Obtain a tentative new state x' by randomly changing M labels of the current state;
 4. Compute $P(x')$ using Equation 2;
 5. Compute $\alpha = \frac{P(x')}{P(x^{(t)})}$;
 6. If $\alpha \geq 1$, the new state is accepted, otherwise, accept the new state with probability α ;
 7. Goto step 2 until the states converge to a stable distribution.

Results: Single Image



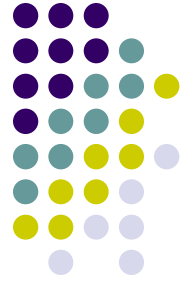
Novel View/Light Prediction



Texture Mapping

Ground Truth

Our synthesis



Conclusion

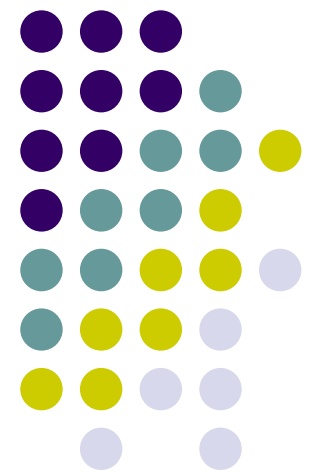
- Present a framework for representing natural textures by a universal 3D texton vocabulary
- Demonstrate excellent results for recognizing 3D textures from a single image under any viewing/lighting directions
- Showing how to predict the appearance of natural material under novel viewing/lighting conditions
- Can be combined with a texture synthesis algorithm to generate new samples of materials under all viewing and illumination conditions

Bag of Keypoints: A Analogous Method

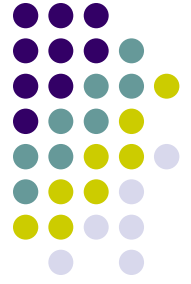
Hao Tang

Based on

G. Csurka et al, "Visual Categorization
with Bags of Keypoints," ECCV04

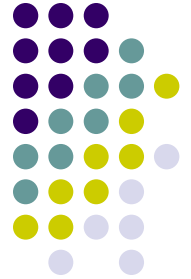


Visual Scene Categorization



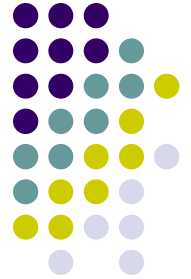
- The problem of identifying object contents of natural images
- Cope with many object types simultaneously
- Handle the variations in view, imaging, lighting, occlusion, etc.

Ideas



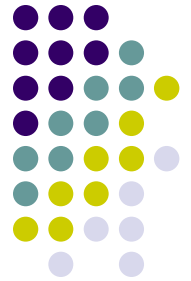
- Can we borrow ideas from texture recognition that we just talked about?

Analogous Ideas



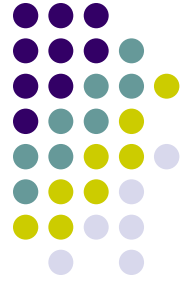
Texture Recognition	Scene Categorization
DoG filter responses	SIFT descriptors
Textons	Keypoints
Recognition based on texton histogram	Recognition based on “bag of keypoints”

The Method



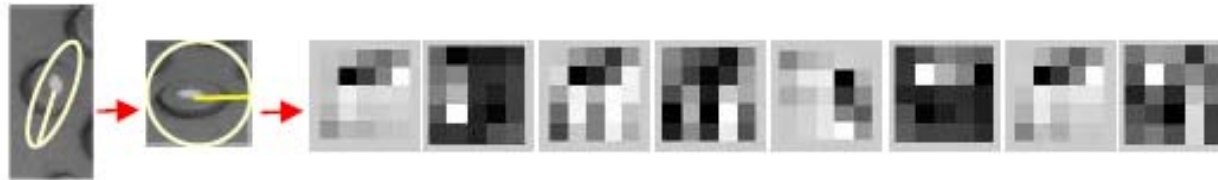
The main steps of our method are:

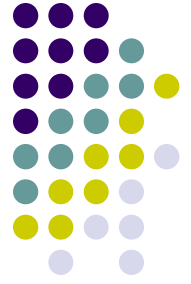
- Detection and description of image patches
- Assigning patch descriptors to a set of predetermined clusters (a *vocabulary*) with a vector quantization algorithm
- Constructing a *bag of keypoints*, which counts the number of patches assigned to each cluster
- Applying a multi-class classifier, treating the bag of keypoints as the feature vector, and thus determine which category or categories to assign to the image.



Feature Extraction

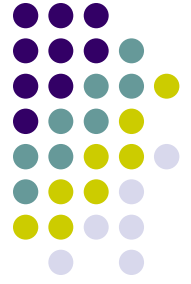
- Harris affine detector
- Scale Invariant Feature Transform (SIFT) descriptors





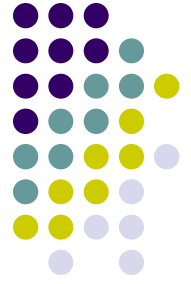
Vocabulary Construction

- Cluster feature vectors (SIFT descriptors) into a small set of prototype feature vectors using K-means
- Those prototype feature vectors are called “keypoints”
- Each image can be represented as a “bag of keypoints”



Categorization

- Given “keypoints”, the problem of visual scene categorization reduces to that of multi-class classification
- Classifiers
 - Naïve Bayes
 - Support Vector Machine (SVM)

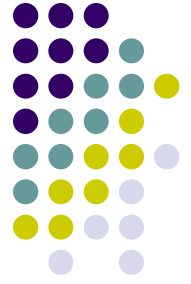


Naïve Bayes

- Maximum Posteriori Probability Classifier
- Simple but efficient
- Use Bayes' rule

$$P(C_j | I_i) \propto P(C_j)P(I_i | C_j) = P(C_j) \prod_{t=1}^{|V|} P(v_t | C_j)^{N(t,i)}.$$

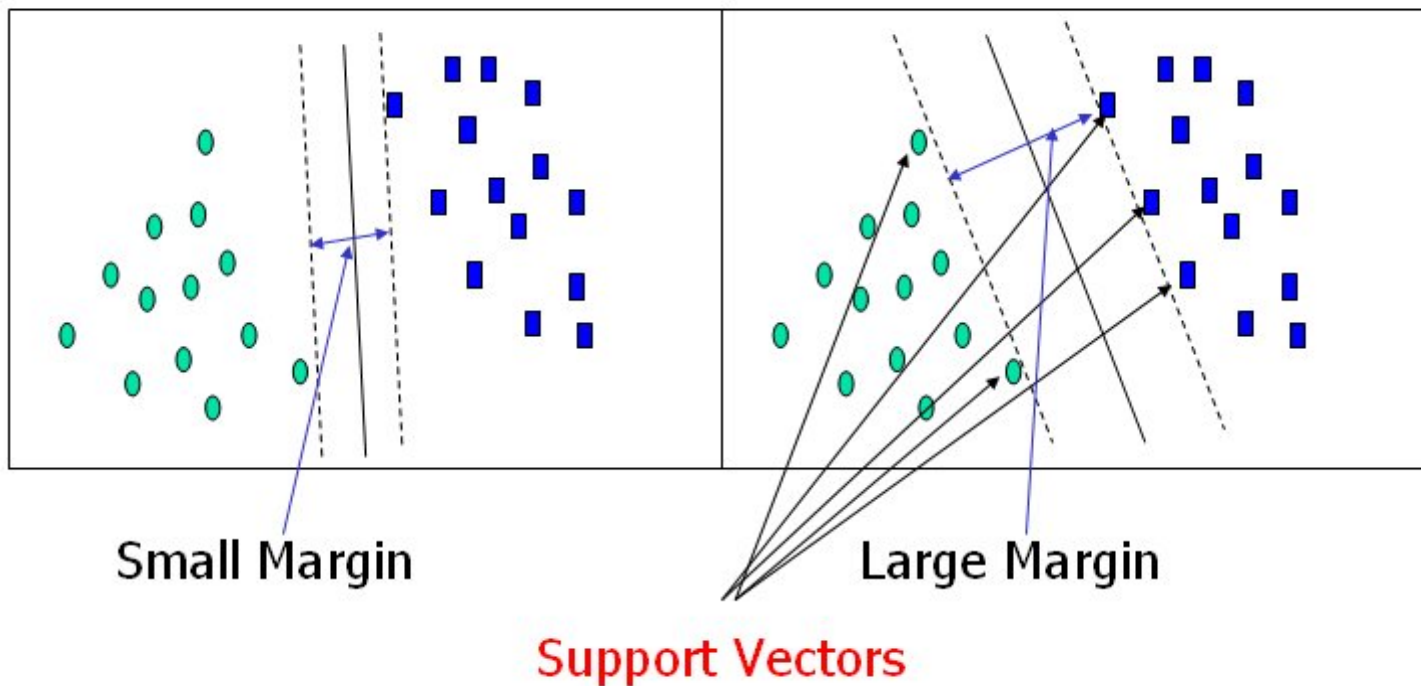
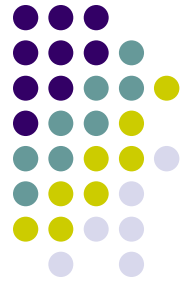
$$P(v_t | C_j) = \frac{1 + \sum_{\{I_i \in C_j\}} N(t, i)}{|V| + \sum_{s=1}^{|V|} \sum_{\{I_i \in C_j\}} N(s, i)}.$$



Support Vector Machine

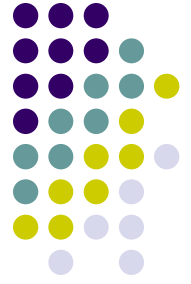
- “State-of-the-art” technique
- Maps data from original space to a higher (even infinite) dimensional feature space where data are linearly separable
- SVM finds a hyperplane in the feature space that separates the two classes with maximum margin
- Property: sparseness

Support Vector Machine

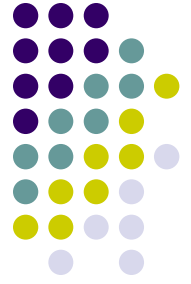


Reference: <http://www.dtreg.com/svm.htm>

Experiments



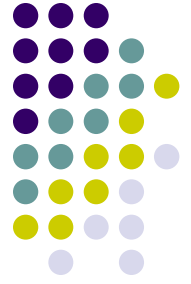
- The purpose of the experiments
 - Explore the impact of the number of “keypoints” on classification accuracy
 - Compare the performance of Naïve Bayes and SVM



Database

- 1776 images, 7 classes





Performance Measures

- Confusion matrix

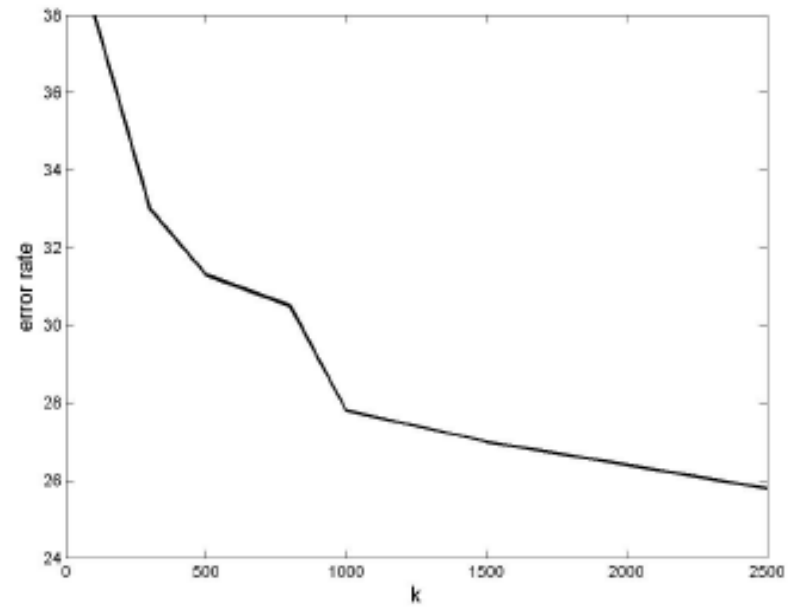
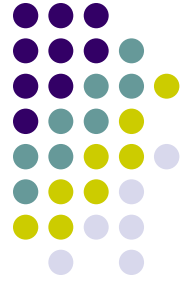
$$M_{ij} = \frac{|\{I_k \in C_j : h(I_k) = i\}|}{|C_j|}$$

- Overall error rate

$$R = 1 - \frac{\sum_{j=1}^{N_c} |C_j| M_{jj}}{\sum_{j=1}^{N_c} |C_j|}$$

- The mean ranks

Results



Results

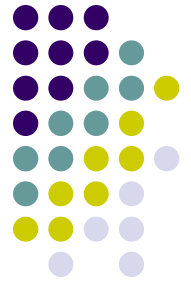


Table 1. Confusion matrix and the mean rank for the best vocabulary ($k=1000$).

True classes →	<i>faces</i>	<i>buildings</i>	<i>trees</i>	<i>cars</i>	<i>phones</i>	<i>bikes</i>	<i>books</i>
<i>faces</i>	76	4	2	3	4	4	13
<i>buildings</i>	2	44	5	0	5	1	3
<i>trees</i>	3	2	80	0	0	5	0
<i>cars</i>	4	1	0	75	3	1	4
<i>phones</i>	9	15	1	16	70	14	11
<i>bikes</i>	2	15	12	0	8	73	0
<i>books</i>	4	19	0	6	7	2	69
<i>Mean ranks</i>	1.49	1.88	1.33	1.33	1.63	1.57	1.57

Results

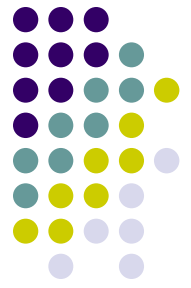


Table 2. Confusion matrix and mean rank for SVM ($k=1000$, linear kernel).

True classes →	<i>faces</i>	<i>buildings</i>	<i>trees</i>	<i>cars</i>	<i>phones</i>	<i>bikes</i>	<i>books</i>
<i>faces</i>	98	14	10	10	34	0	13
<i>buildings</i>	1	63	3	0	3	1	6
<i>trees</i>	1	10	81	1	0	6	0
<i>cars</i>	0	1	1	85	5	0	5
<i>phones</i>	0	5	4	3	55	2	3
<i>bikes</i>	0	4	1	0	1	91	0
<i>books</i>	0	3	0	1	2	0	73
<i>Mean ranks</i>	1.04	1.77	1.28	1.30	1.83	1.09	1.39

Results

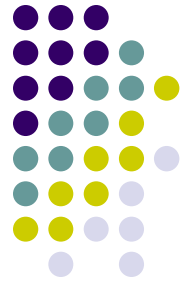


Fig. 5. Images correctly classified containing multiple objects of the same category.



Fig. 6. Profile face, partial view of a car, roof of a house correctly classified as face, cars, building.