Multimedia Content Analysis and Indexing

Bag of Visual Words

Outline

- Content based image retrieval
- Bag of Words
- Bag of Visual Words
- Classification Pipeline
  - Feature Extraction
  - Codebook Generation
  - Classification
- Selected Use Cases
- Bag of Visual Words Outro ???

Content based image retrieval revisited

- Use content based analysis and indexing to describe and store meta data of multimedia content
- Use visual content of videos or images for retrieval
- Extract global features (color, edge information, etc.)
- Store visual fingerprint of multimedia data in a vector (e.g. color histogram)
- Compute distances between vectors in order to measure their similarity
- Use a so called BoW approach instead of using global features, to retrieve more satisfying results

Bag of Words for image retrieval

- Derived from text retrieval
- Documents are represented by word frequencies
- Example
  - Document1: The bag of words approach in text retrieval is also used for content based image retrieval.
  - Document2: Content based image retrieval is cool.
  - Dictionary = {the, bag, of, words, approach, in, text, retrieval, is, also, used, for, content, based, image, cool}
  - Doc1 = [1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,0]
  - Doc2 = [0,0,0,0,0,0,0,1,0,0,0,1,1,1,1,1]

Bag of Visual Words

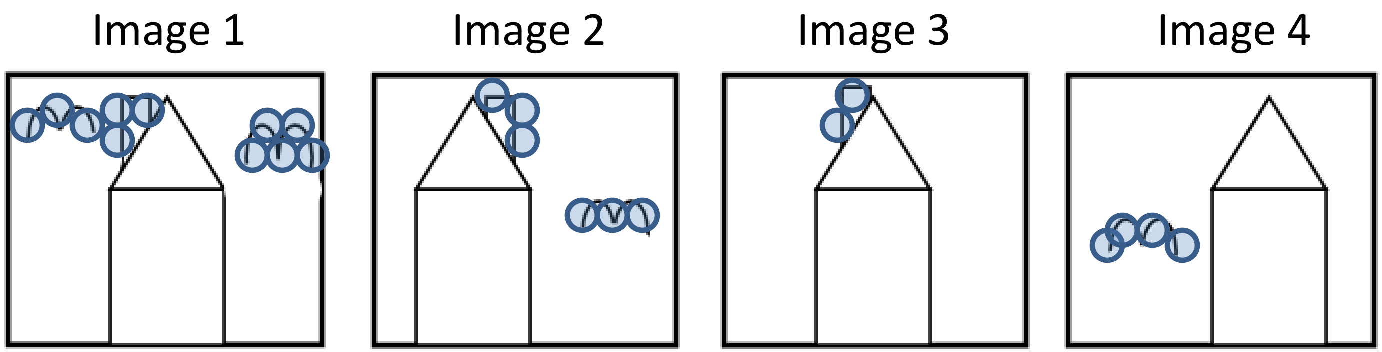
- Gives superior classification results over a global feature approach
- Computationally expensive
  - TrecVid Video Retrieval Task
  - 40,000 frames (1 frame per shot of 180 hours of video)
- Sample local regions from an image – convert them to visual words

BoVW Classification Pipeline Overview

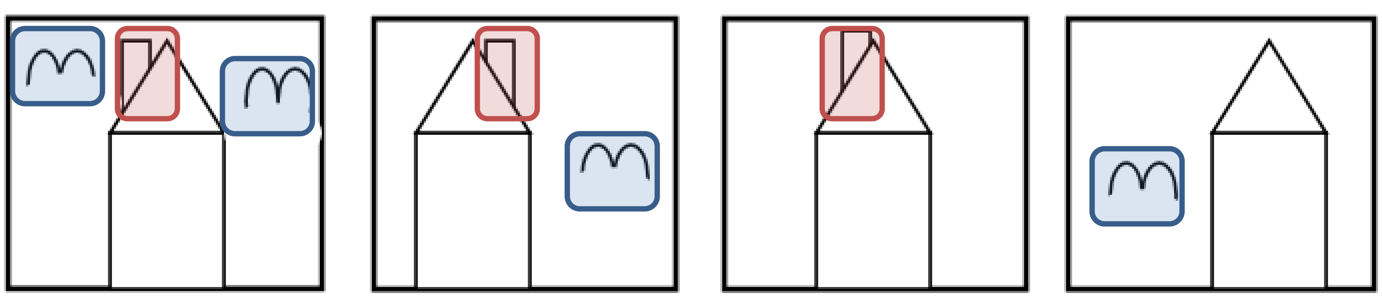
- Descriptor Extraction
  - Extract local regions from an image
  - Description of regions by feature descriptors (vectors)
- Codebook Generation / Word Assignment
  - A visual vocabulary (codebook) consisting of visual words must be generated before assignment step (use clustering techniques like K-Means)
  - map descriptor to the most similar visual word (using K-nearest neighbor search)
- Classification
  - Classify images to retrieve similar ones during retrieval process

BoVW
Codebook Generation Example

- Feature Extraction (salient image patches)



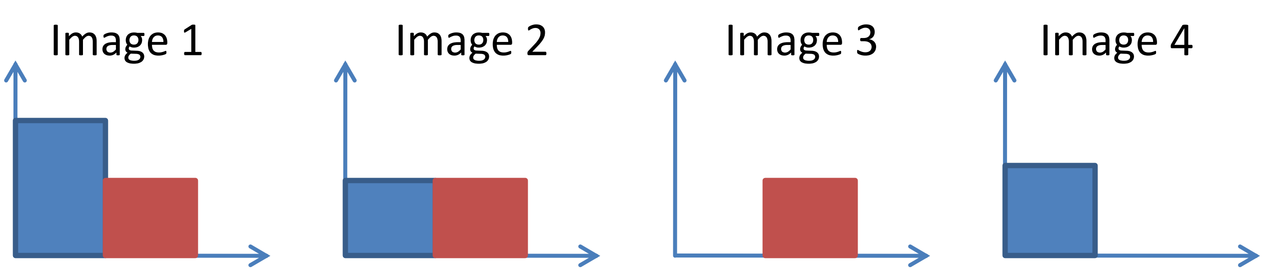
- Cluster features and generate codebook



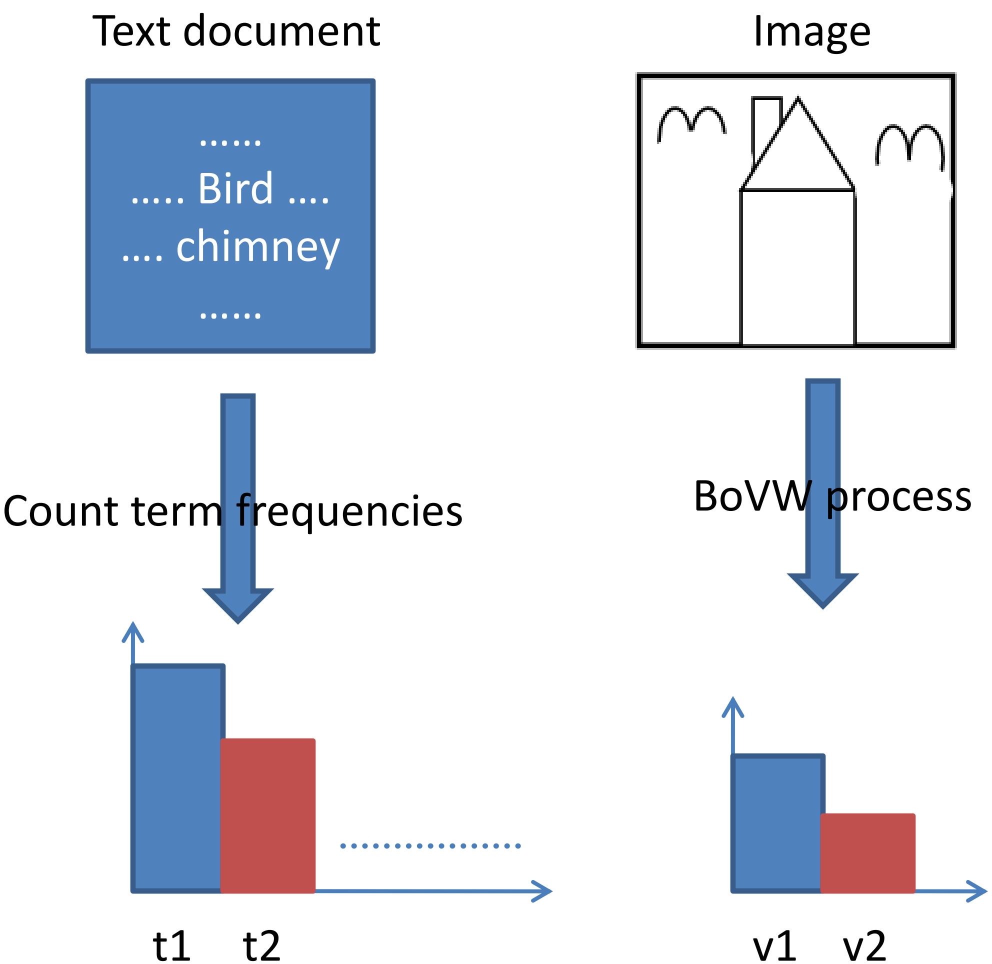
BoVW
LFH Computation Example

- Generated Codebook ( ) … consisting of visual words
- Ordinarily a codebook would consist of many visual words depicting the bird. In this case the bird is represented as one visual word to simplify the example.
- Compute Local Feature Histograms by measuring the distances between the extracted features and the cluster centers. Count the amount of image features belonging to a certain cluster.

	Image 1	Image 2	Image 3	Image 4
Cluster 1	8	3	0	4
Cluster 2	3	3	2	0

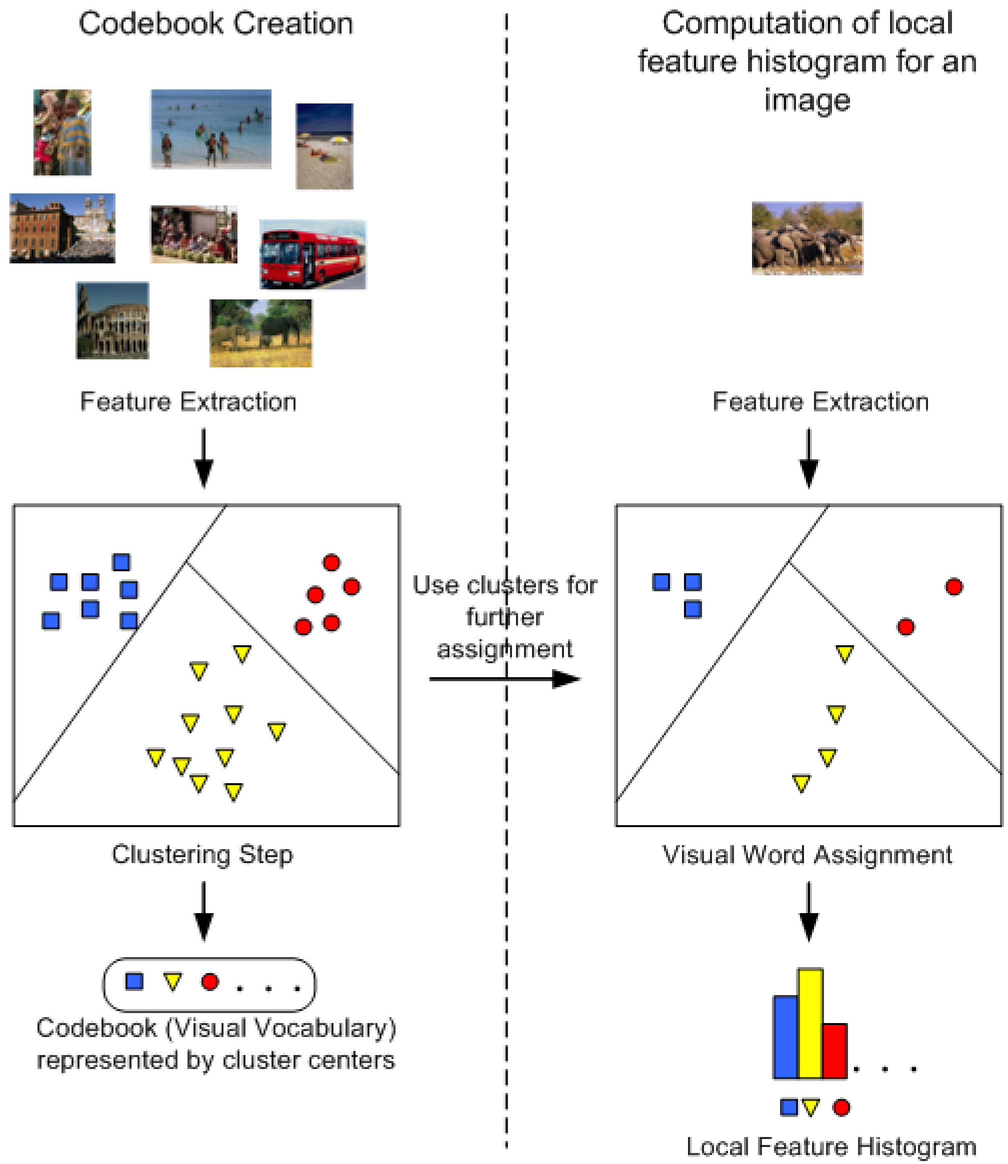


BoVW
Words and Visual Words Comparison



- Terms in documents are counted and summed up (depicted in the left histogram)
- Visual features are extracted from an image an assigned to the nearest cluster center (one visual word). Cluster assignments are counted (depicted in the right histogram)

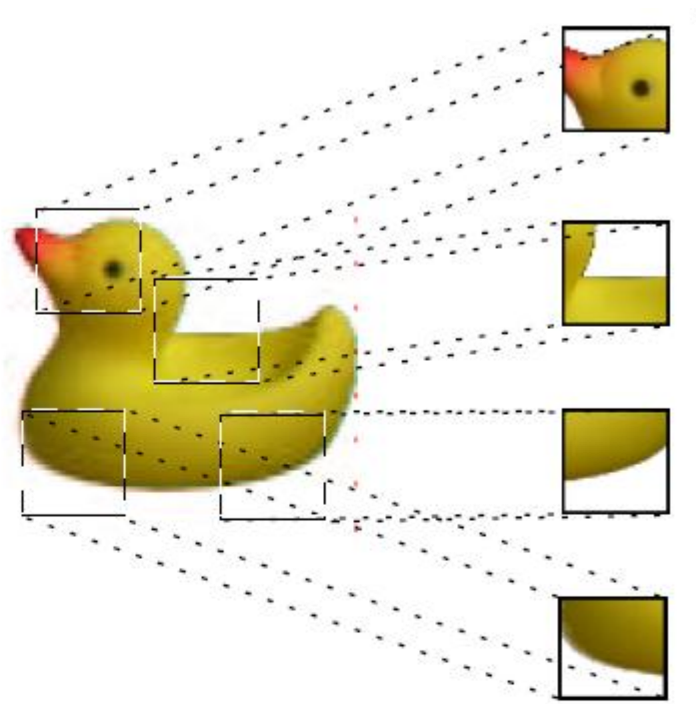
BoVW
Codbook Generation / Visual Words assignment



BoVW Classification Pipeline
Feature Extraction

- Global features describe an image in an holistic way
- Local features describe salient image patches within images
- Scale-Invariant Feature Transform (SIFT)
  - Based on grayscale images
  - Robust against image scale, rotation, viewpoint change, illumination
  - Used for object recognition, image stitching, etc.

BoVW Classification Pipeline
Feature Extraction

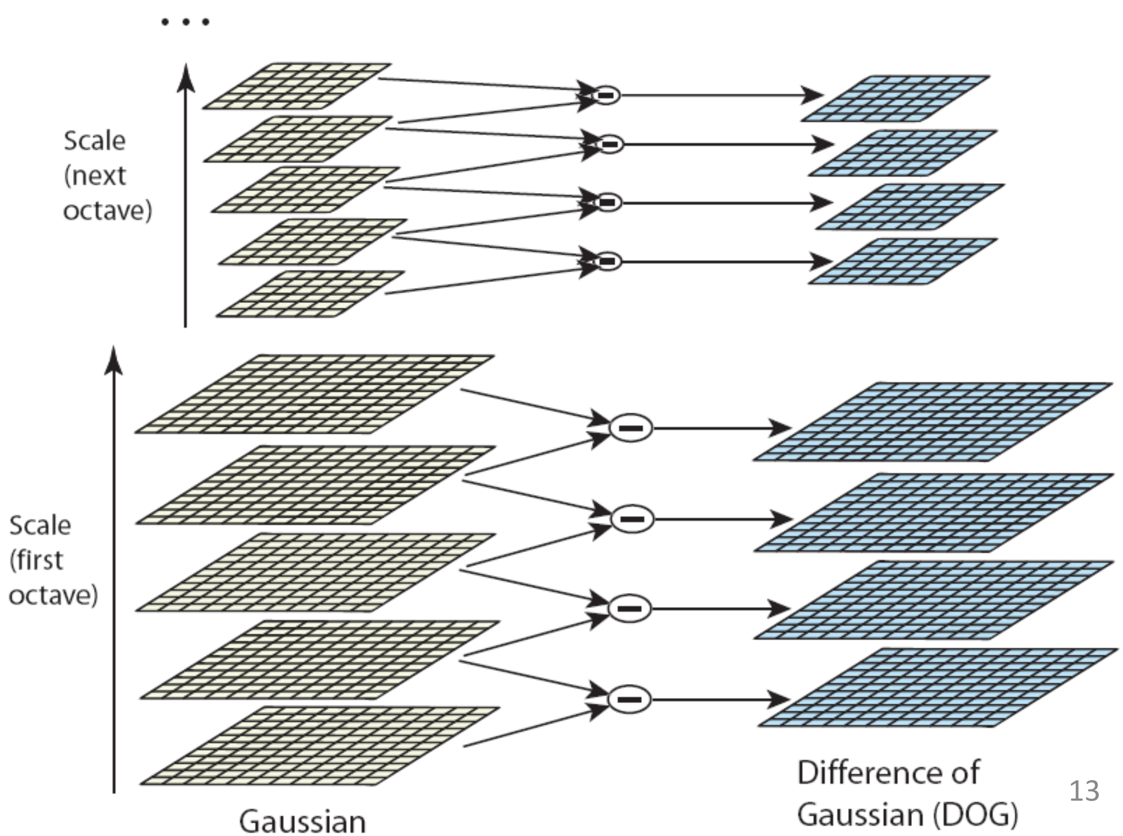


How do we extract local image features like the depicted ones?

Feature Extraction
Scale-space extrema detection

– identify interest points within an image by using Difference of Gaussians

-Use Gaussian blurred images at different octaves (resolutions)
-Compute differences of adjacent blurred images pixel wise
-Results in DoG images

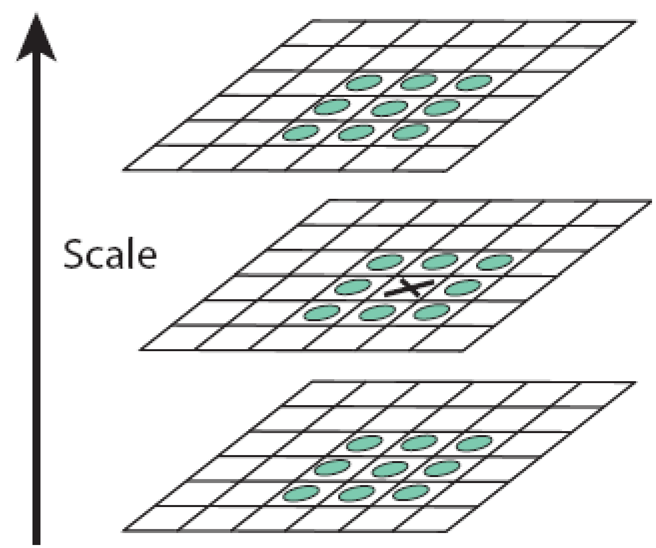


Feature Extraction

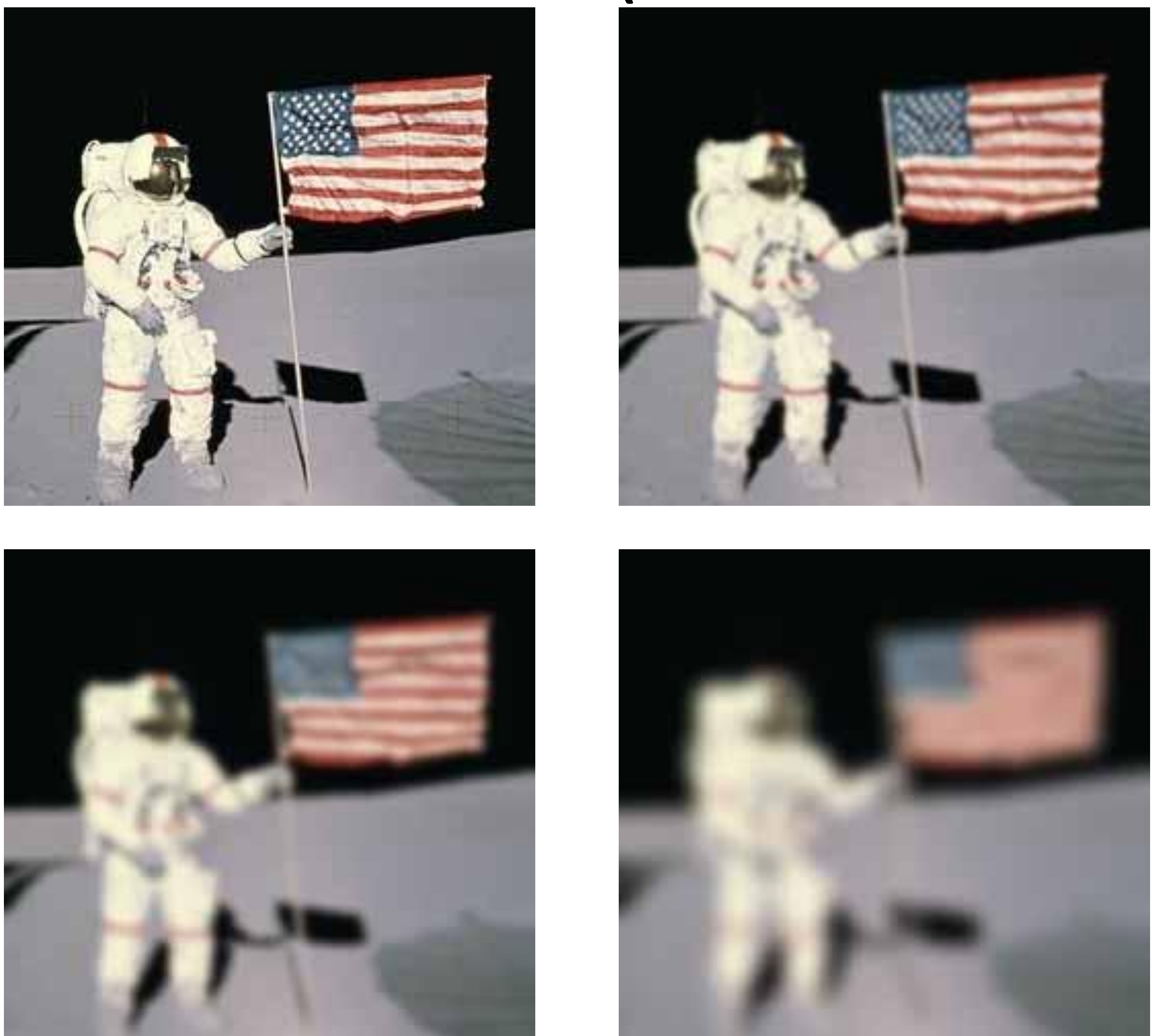
Scale-space extrema detection

– Difference of Gaussians images are computed from adjacent Gaussian blurred images per octave

-Compare each pixel in a DoG Image with it‘s eight neighboring pixels and with the nine neighboring pixels of the adjacent scales
-Find local minima and maxima of pixel values
-These are considered as interest points
-Get invariance to image scaling



Feature Extraction
Scale-space extrema detection
(Gaussian Blur)



Smooth image

Feature Extraction
Keypoint localization

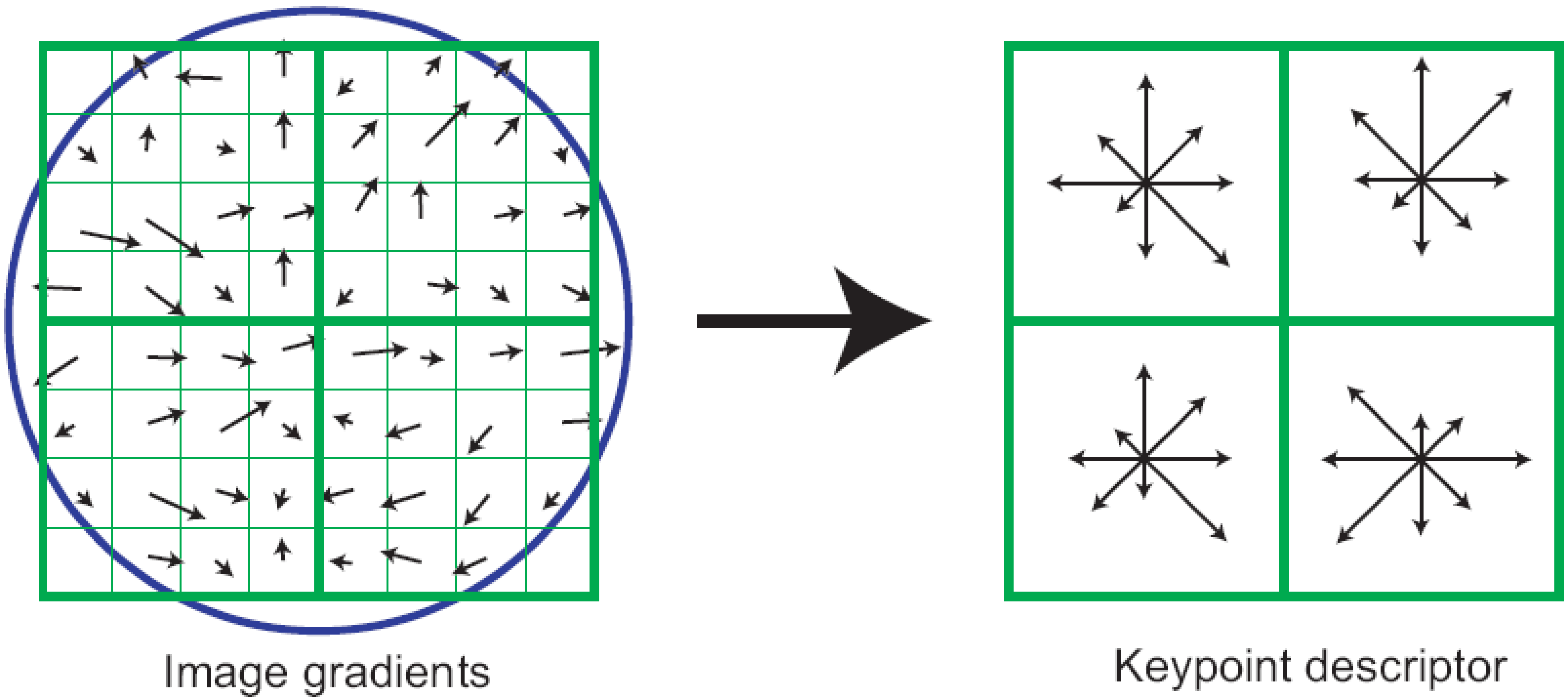
- Too many computed interest points
- Some of them are not adequate / stable
  - Reject points with low contrast
  - Reject points which are not localized along edges
- Interpolate nearby data to get stable keypoints

Feature Extraction
Orientation Assignment

- Assign orientations to the identified keypoints
- Compute gradients (describe change between pixel intensity values in terms of magnitude and orientation) around the keypoints at a specific scale
- Achieve invariance to image rotation

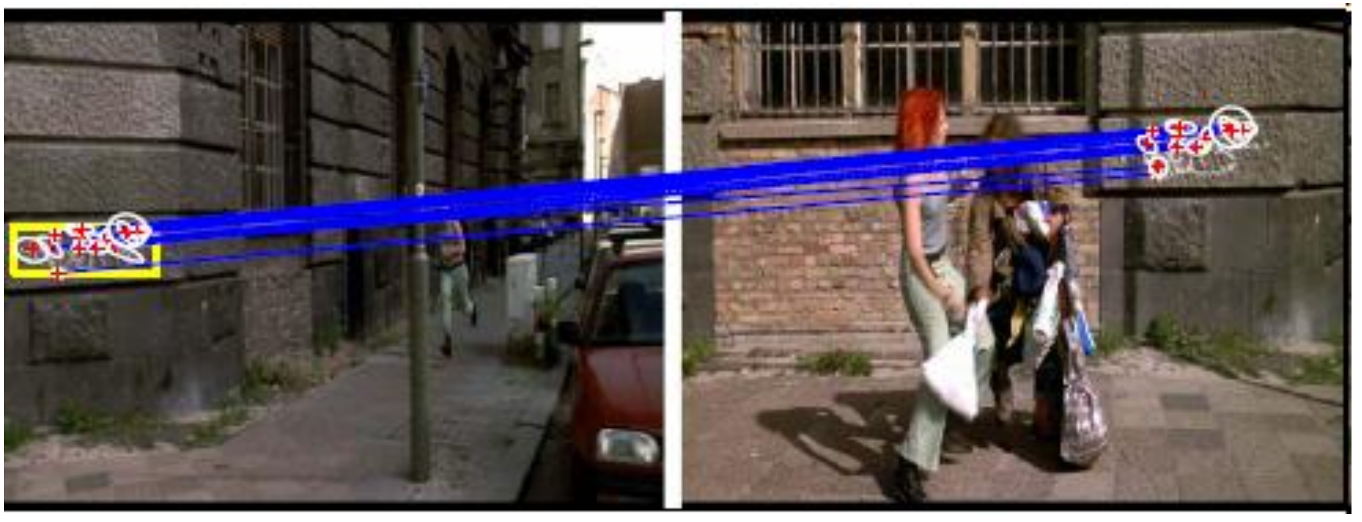
Feature Extraction
Keypoint Descriptor

- Gradient samples (depicted in the right picture) are accumulated into orientation histograms
- Length of each arrow correspond to the sum of the gradient magnitudes near a specific direction
- Figure shows only 2x2 descriptor array computed from 8x8 set of samples
- Usually a 4x4 descriptor array is used computed from 16x16 sample array
- 4x4 array x 8 bin hist => 128 dim vector

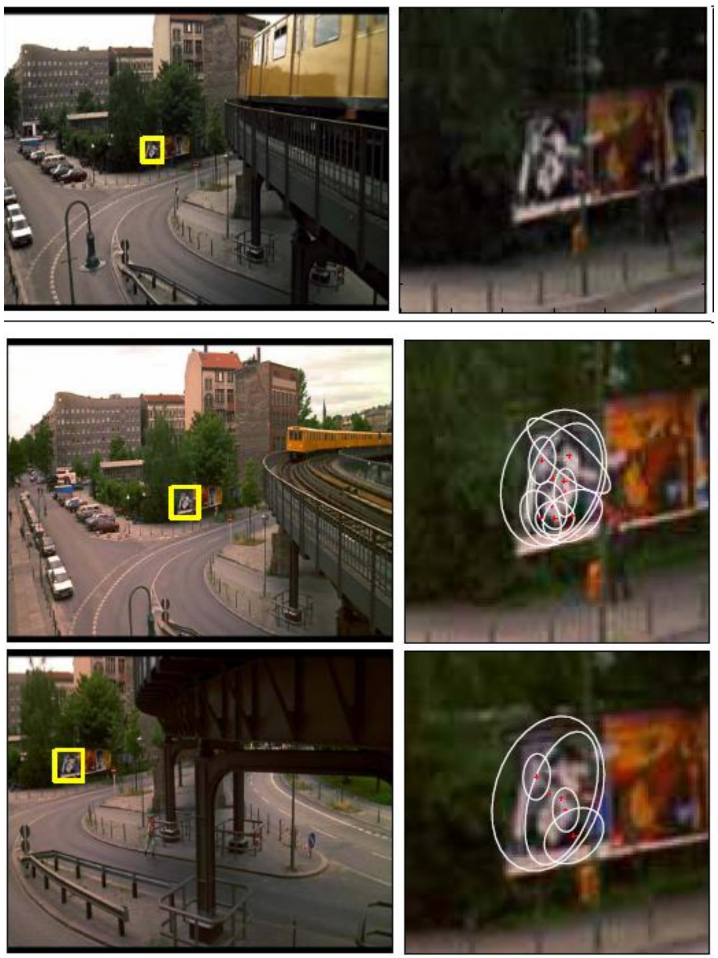


Feature Extraction
Feature Matching

- Similar salient image patches (different viewpoints)



Feature Extraction
Feature Matching



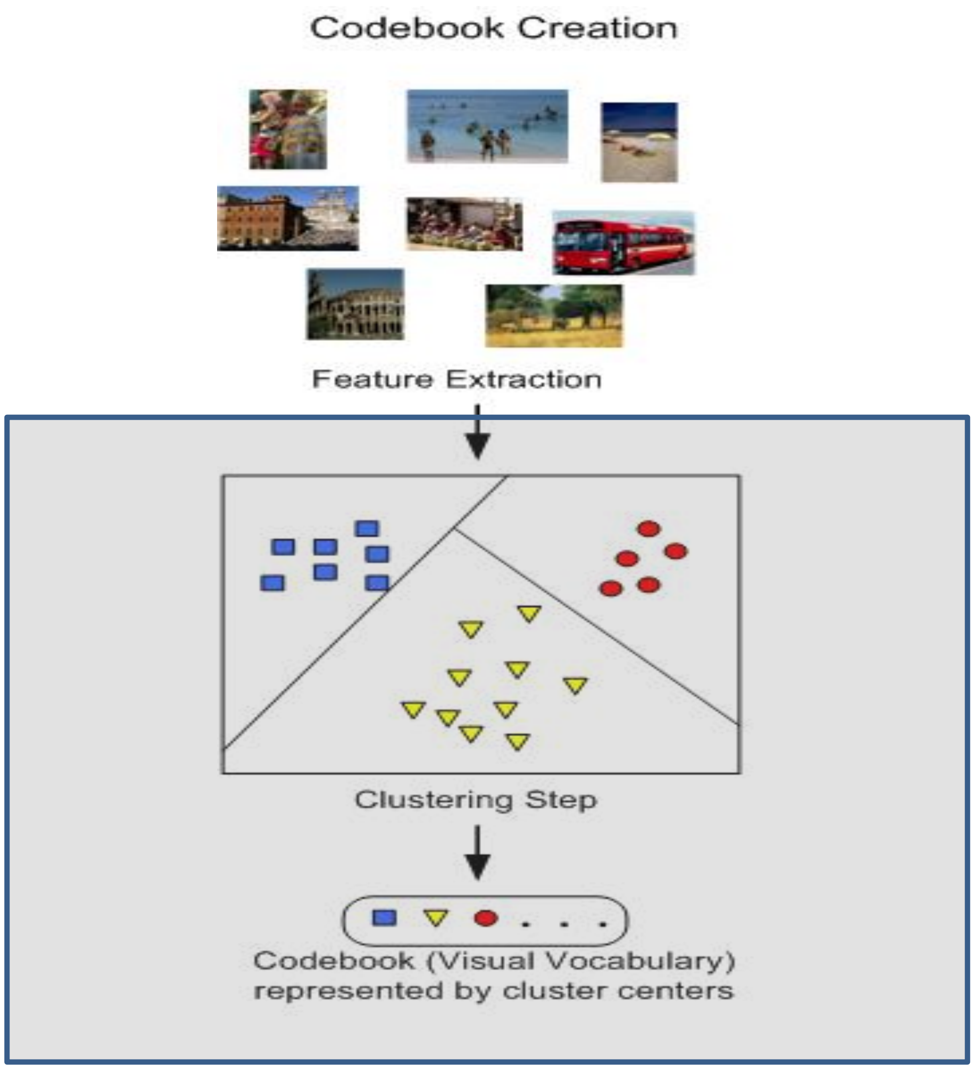
Feature Extraction

- Use Cases
  - Object Recognition
  - Image Stitching
    - Show Photosynth from Microsoft (http://photosynth.net/default.aspx)
  - Etc.
- Produces too many features if you are considering a vast amount of images
- Combine similar features of different images

BoVW Classification Pipeline Codebook Generation / Word assignment

- Codebook Generation
  - K-Means (traditional approach)
  - Fuzzy C-Means (fuzzy codebooks)
- Word Assignment
  - K-Nearest Neighbor Search

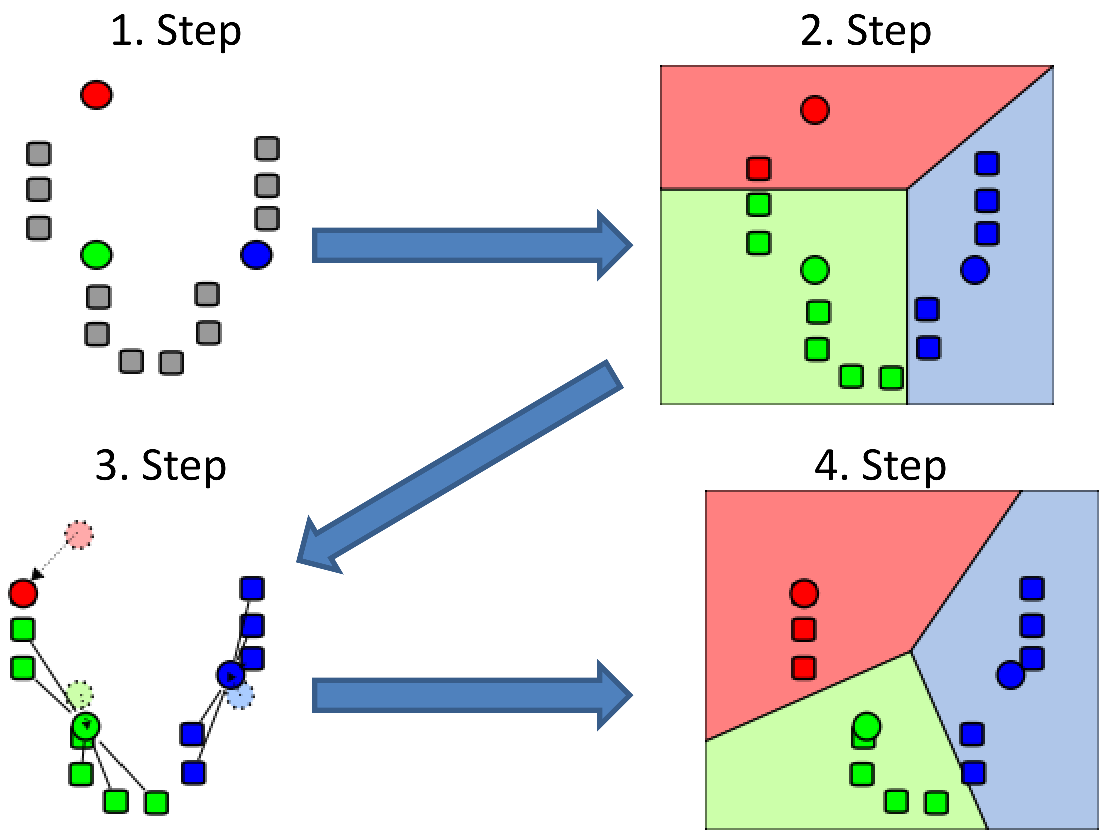
BoVW Classification Pipeline Codebook Generation



1) Cluster extracted local features
2) Cluster Centers denoting a cluster represent visual words
3) Visual Words form a codebook

BoVW Classification Pipeline
Codebook Generation with K-Means

1. Step: Choose cluster centers randomly
2. Step: assign samples to the nearest cluster centers by using a distance metric
3. Step: re-compute cluster centers (choose mean value of samples belonging to a cluster)
4. Step: GOTO Step 2 if cluster centers change or stop if they do not change or maximal iteration depth is reached



BoVW Classification Pipeline
Codebook Generation with K-Means

- Example:
  data: (1,1,1); (2,2,2); (3.5,3.5,3.5); (5,5,5); (6,6,6);
  Step 1) Cluster Centers: (1,1,1); (5,5,5)
  Step 2)
    Cluster (1,1,1): (1,1,1); (2,2,2)
    Cluster (5,5,5): (3.5,3.5,3.5); (5,5,5); (6,6,6)
  Step 3)
    Re-compute (1,1,1): (1.5,1.5,1.5)
    Re-compute (5,5,5): (4.83, 4.83, 4.83)
  Step 4) & 2)
    Cluster (1.5,1.5,1.5): (1,1,1); (2,2,2)
    Cluster (4.83, 4.83, 4.83): (3.5,3.5,3.5); (5,5,5); (6,6,6)
  Step 3)
    Re-compute (1.5,1.5,1.5): (1.5,1.5,1.5)
    Re-compute (4.83, 4.83, 4.83): (4.83, 4.83, 4.83)
  Step 4) Cluster Centers do not change => end of algorithm

BoVW Classification Pipeline
Codebook Generation with K-Means

- http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html

BoVW Classification Pipeline
Codebook Generation with Fuzzy C-Means

$\mu_A : X \to [0,1]$ determines the degree of an element $x \in X$ belonging to a set A

Set A denotes a cluster of local features

The sum of membership values of an element x to all clusters is one $\sum_{A_i} \mu_{A_i}(x) = 1$

Membership function is used to assign datapoints $\vec{d} \in D$ to clusters $c_i \in C$ with $\dot{\bigcup}_{c_i \in C} c_i = D$ and to compute cluster centers $\vec{m_i} \in M$

Parameter $m \in [1, \infty)$ is called fuzzifier and controls the membership function

Compute new cluster centers based on the actual degree of membership of a vector to a cluster

$$\vec{m_i} = \frac{\sum_{\vec{d} \in D} \mu_{c_i}(\vec{d})^m \vec{d}}{\sum_{\vec{d} \in D} \mu_{c_i}(\vec{d})^m}$$

Compute degree of membership of a vector to a cluster

$$\mu_{c_i} = \frac{1}{\sum_{m_k \in M} \left(\frac{L_2(m_i,d)}{L_2(m_k,d)}\right)^{\frac{2}{m-1}}}$$

Optimize objective function

$$f = \sum_{\vec{d} \in D} \sum_{\vec{m_i}=1}^{c} L_2(\vec{d}, \vec{m_i})^2 \mu_{c_i}(\vec{d})^m$$

BoVW Classification Pipeline Codebook Generation with Fuzzy C-Means

1. Randomly select n cluster centers.
2. Determine membership of each data point to each cluster (using the cluster center).
3. Compute f_{last}.
4. Re-compute cluster centers based on the determined membership values.
5. Determine membership of each data point to each cluster (using the cluster center).
6. Compute f_{actual}.
7. Step
    1. If $|f_{actual} - f_{last}| < \epsilon$ stop.
    2. Else set f_{last} to f_{actual} and start over with step 4

BoVW Classification Pipeline
Codebook Generation with Fuzzy C-Means

- Example

  data: (1,1,1); (2,2,2); (3.5,3.5,3.5); (5,5,5); (6,6,6);

  Step 1) Cluster Centers: (3.5,3.5,3.5); (6,6,6)

  Step 2) membership of data points

	Cluster 1	Cluster 2
(1,1,1)	0.7291294502293906	0.2708705497706093
(2,2,2)	0.8023718068784541	0.19762819312154578
(3.5, 3.5, 3.5)	1.0	0
(5,5,5)	0.35910843920785	0.6408915607921499
(6,6,6)	0	1.0

Step 3) flast = 18.61699733894557

BoVW Classification Pipeline Codebook Generation with Fuzzy C-Means

Step 4) re-compute cluster centers: (2.600989135231607, 2.600989135231607 2.600989135231607) ; (5.543211468303816, 5.543211468303816 5.543211468303816)

Step 5) membership of data points

	Cluster 1	Cluster 2
(1,1,1)	0.8160809498452128	0.18391905015478716
(2,2,2)	0.9265313076469203	0.07346869235307985
(3.5, 3.5, 3.5)	0.7636566806303208	0.2363433193696928
(5,5,5)	0.1069887144898244	0.8930112855101756
(6,6,6)	0.05380052021007118	0.9461994797899289

Step 6) factual = 7.425776219371355

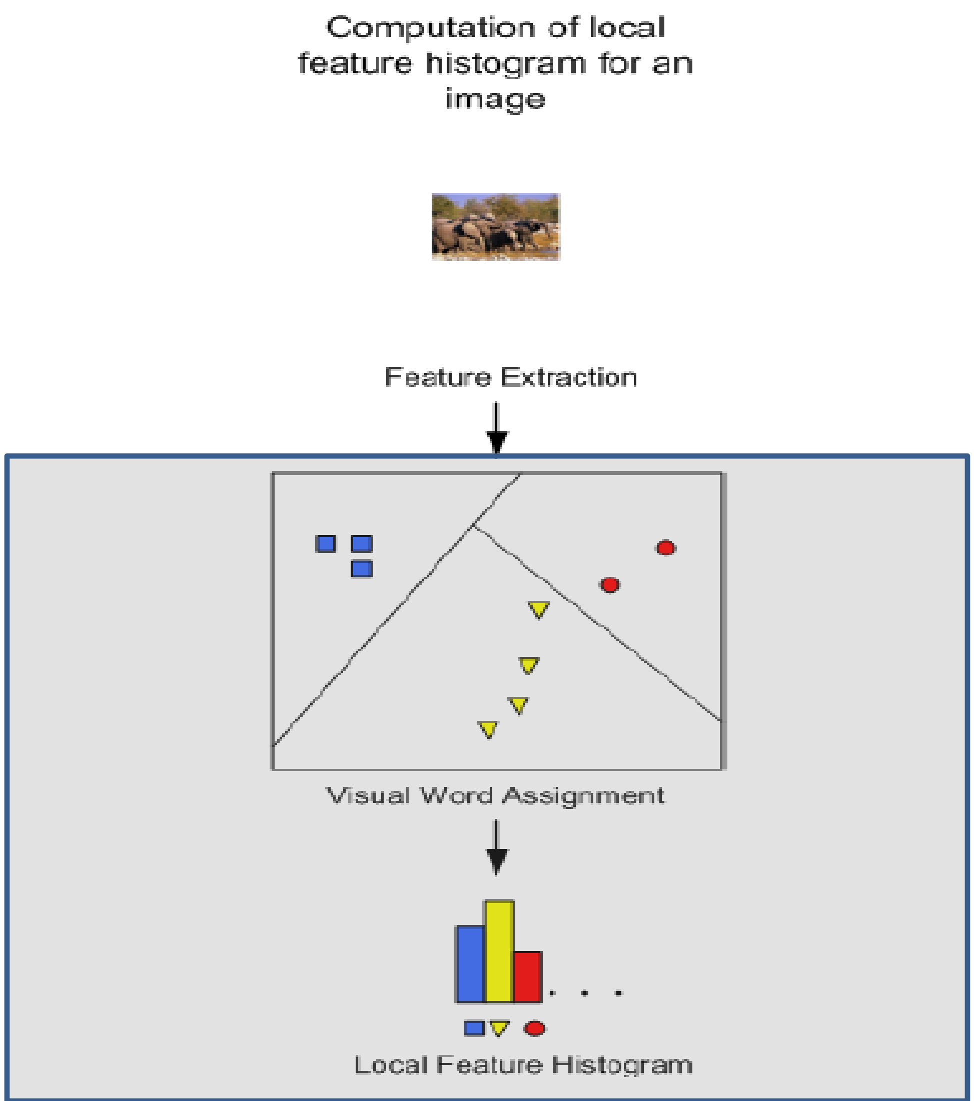
Step 7) 18.61699733894557 - 7.425776219371355 = 11.19122 > ϵ => goto step 4

Fuzzy C-Means vs. K-Means

Fuzzy C-Means	Cluster 1 (1.7361657620326942 1.7361657620326942 1.7361657620326942)	Cluster 2 (5.28809932021391 3 5.28809932021391 3 5.28809932021391 3)
(1,1,1)	0.9253490612413994	0.07465093875860056
(2,2,2)	0.9735043034201096	0.026495696579890357
(3.5,3.5,3.5)	0.5048795735307412	0.4951204264692587 5
(5,5,5)	0.030246934145457 18	0.969753065854542 9
(6,6,6)	0.0719494506302739	0.9280505493697261

K-Means	Cluster 1 (4.83, 4.83, 4.83)	Cluster 2 (1.5,1.5,1.5)
(1,1,1)	0	1
(2,2,2)	0	1
(3.5,3.5,3.5)	1	0
(5,5,5)	1	0
(6,6,6)	1	0

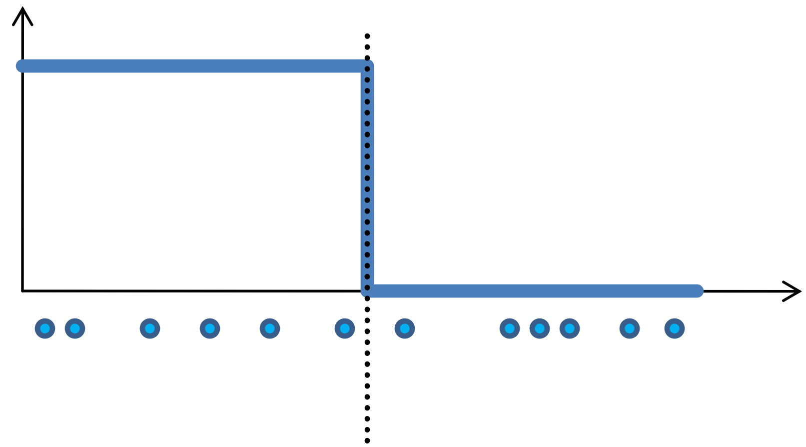
BoVW Classification Pipeline
Word assignment



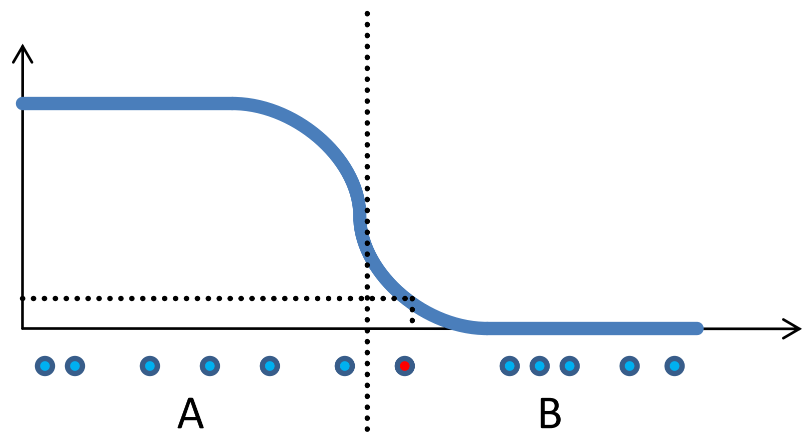
1) Assign extracted local features to the nearest visual words by employing a nearest neighbor search
2) Create a local feature histogram denoting the distribution of extracted local features over the pre-computed clusters (represented by cluster centers, which depict the visual words)

BoVW Classification Pipeline
Word assignment

- Hard Assignment

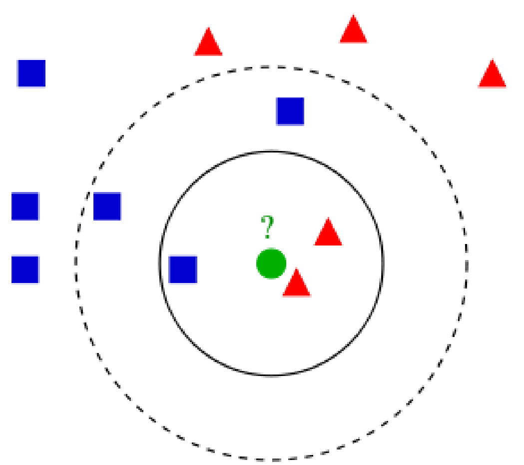


- Soft Assignment



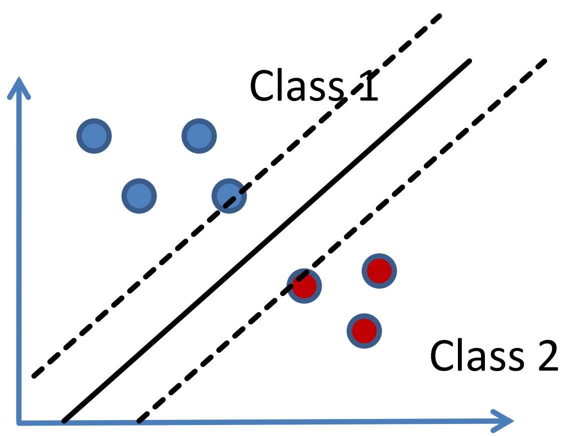
BoVW Classification Pipeline Classification

- Use machine learning algorithms to classify and search for similar objects
- K-Nearest Neighbor Search
  - Search for similar objects based on a chosen distance metric (e.g. Euclidean Distance)



BoVW Classification Pipeline
Classification

- Support Vector Machine (SVM)
  - Classify objects into two different categories by utilizing a hyper plane (in 2-dimensional space a simple line would separate the different samples from each other)



-Choose the hyper plane, which maximizes the distance to the nearest data point on each side
-Those data points are called support vectors
-New points are classified by computing the distances to the support vectors

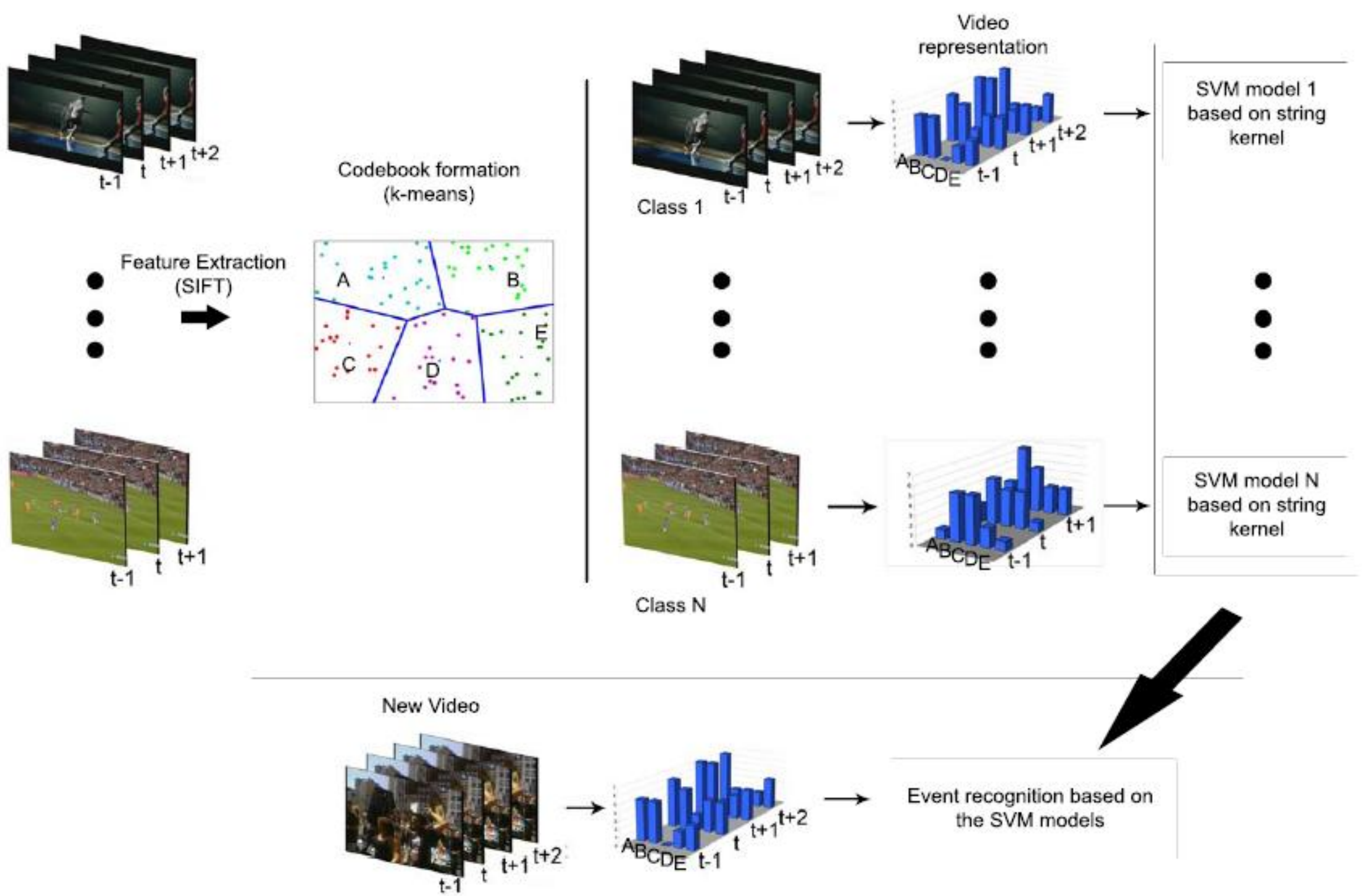
BoVW
Selected Use Cases

- concept detection in large image collections
- video event classification
- automatic tagging
- video clip summarization (present software prototype ... live demo)

Bag of Visual Words
Outro (1)

- What about videos?
- Traditional BoVW approach uses static features on a keyframe basis
- Doesn‘t consider temporal relations between frames within videos
- Consider temporal relations by employing sequences of Local Feature Histograms
- Compare sequences of different video clips to detect similar video events

Bag of Visual Words
Outro (2)



Bag of Visual Words
Outro (3)

- Other challenging topics
  - Feature Extraction
    - SURF
    - Maximally Stable Extremal Regions (MSER)
    - ……..
  - Codebook Generation / Word Assignment
    - Various soft assignment techniques
    - Visual Words weighting (e.g. with TF-IDF)
    - ……..

Selected Literature

- J. R. R. Uijlings, A. W. M. Smeulders, and R. J. H.Scha. Real-time bag of words, approximately. In S. Marchand-Maillet and Y. Kompatsiaris, editors, CIVR. ACM, 2009.
- R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, inuences, and trends of the new age. ACM Computing Surveys (CSUR), 40, 2008.
- D. G. Lowe. Object recognition from local scale-invariant features. In ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2, page 1150, Washington, DC, USA, 1999. IEEE Computer Society.
- J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo. Evaluating bag-of-visual-words representations in scene classication. In J. Z. Wang, N. Boujemaa, A. D. Bimbo, and J. Li, editors, Multimedia Information Retrieval, pages 197{206. ACM, 2007.
- Y.-G. Jiang and C.-W. Ngo. Bag-of-visual-words expansion using visual relatedness for video indexing. In SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pages 769{770,New York, NY, USA, 2008. ACM.