Vector Quantization in Content Based Image Retrieval

Sebastian Böhmer

Student at the
Albert-Ludwigs University Freiburg

Feb. 4, 2005

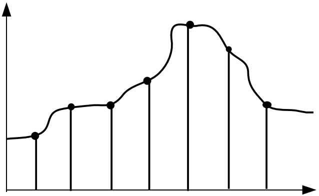
Contents

1. **Introduction in VQ**
   - Quantization
   - Vector Quantization
   - VQ Advantage
2. **VQ Algorithms**
   - Generalized Lloyd Algorithm (GLA)
   - Tree-structured Codebook Representation
3. **VQ detailed**
   - Trainingset
   - Problems of VQ
4. **Implementation**
   - Range Normalization
   - Partial Distance Search
5. **Test Results**
   - first PR-Graph
   - Feature Enhencement

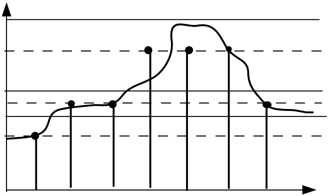
Contents

1. Introduction in VQ
   - Quantization
   - Vector Quantization
   - VQ Advantage
2. VQ Algorithms
   - Generalized Lloyd Algorithm (GLA)
   - Tree-structured Codebook Representation
3. VQ detailed
   - Trainingset
   - Problems of VQ
4. Implementation
   - Range Normalization
   - Partial Distance Search
5. Test Results
   - first PR-Graph
   - Feature Enhencement

Quantization



$\Rightarrow$



continuous values

quantized values

quantization = continuous values mapped to distribution representative values

VQ formal[Theodoridis]

$$Q : R^k \rightarrow C = \{c_1, \ldots, c_i, \ldots, c_n\} | c \in R^k\}$$

C = codebook, n = codebook size and c_i = codewords

For given $T = \{t_1, \ldots, t_i, \ldots, t_m\} \subseteq R^k$ (trainingset)
Q creates partition of T

$$P = \{p_1, \ldots, p_i, \ldots, p_n\}$$

with $p_i = \{t | t \in T \wedge Q(t) = c_i\}$

Optimal Vector Quantizer [Zhu]

A optimal quantizer has two conditions:

- *Nearest Neighbor Condition:*
  For each p_i, if

$$t \in p_i \quad : \quad d(t, c_i) \leq d(t, c_j) \quad \forall j \neq i$$

- *Centroid Condition:*
  Given P

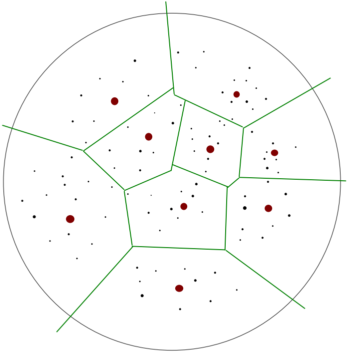
$$c_i = \frac{\sum\limits_{t \in p_i} t}{k_i},$$

where k_i is the cardinality of p_i.

Vector Quantization

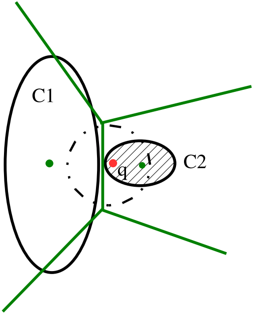
VQ is used for:

- Encoding (loosely)
- Clustering ($\Rightarrow$ classification)



VQ Advantage

The advantage fo VQ in CBIR could be clear distinction of classes



(q Query feature - dashed circle = normal retrieval - hatched area = VQ retrieval)

Contents

1 Introduction in VQ
- Quantization
- Vector Quantization
- VQ Advantage

2 **VQ Algorithms**
- Generalized Lloyd Algorithm (GLA)
- Tree-structured Codebook Representation

3 VQ detailed
- Trainingset
- Problems of VQ

4 Implementation
- Range Normalization
- Partial Distance Search

5 Test Results
- first PR-Graph
- Feature Enhencement

populated Algoritms are:

1. Generalized Lloyd Algorithm (GLA)
2. Pairwise Nearest Neighbour Algorithm (PNNA)
3. Hierarchical, Fuzzy, ...

GLA

1. Given a training set $T = \{t_1, \ldots, t_m\}$, initial codebook $C_1 = \{c_1, \ldots, c_n\}$ and a threshold δ.
2. Use C_k to get the partition P based on *nearest neighbor condition*, and compute the average distortion.

$$D_k = \frac{\sum_{i=1}^{n} \sum_{t \in p_i} d(t, c_i)}{m}$$

   where m is the number of images in the partition P and n the codebook size.
3. If $\frac{D_{k-1} - D_k}{D_k} \leq \delta$, stop the iteration. C_k would be the fi nal codebook.
4. Create a new codebook C_{k+1} by using the *centroid condition* with partition P.
5. Increase the iteration and go on with point 2.

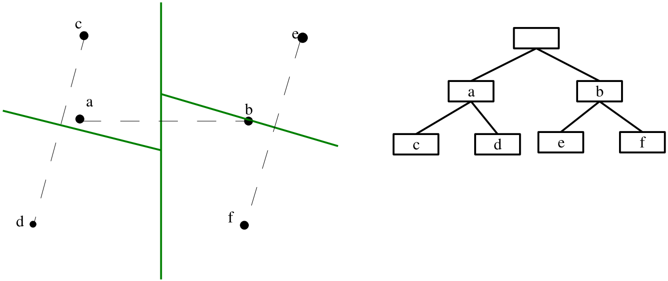
The Tree-structured Codebook is a binary tree with decisively vectors in the nodes and codewords in the leafs.[Cosman]

How to quantize with Tree-structured Codebook

1. Given vector v, start at root
2. compair the distance between v and the left and right nodes

$$d_l = d(v, n_l) \qquad d_r = d(v, n_r)$$

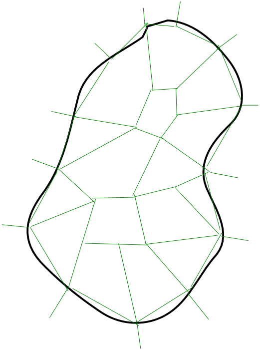
   If $d_1 < d_r$ select the right branch, else select the left branch.
3. go on with point 2 until the leaf is reached



Contents

1. Introduction in VQ
   - Quantization
   - Vector Quantization
   - VQ Advantage
2. VQ Algorithms
   - Generalized Lloyd Algorithm (GLA)
   - Tree-structured Codebook Representation
3. **VQ detailed**
   - Trainingset
   - Problems of VQ
4. Implementation
   - Range Normalization
   - Partial Distance Search
5. Test Results
   - first PR-Graph
   - Feature Enhencement

A good quantization of a class looks like this



Therefor

- codewords which approximate class distribution
- good codebook size

necessary

The codebook is created with the trainingset, a subset of the database.

The trainingset should approximate the class distribution.

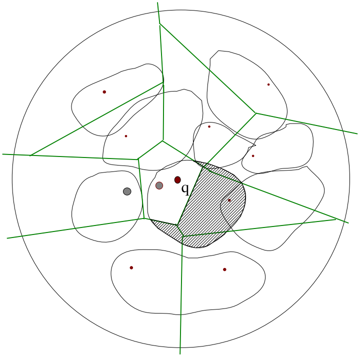
$\Longrightarrow$

- a-priori knowledge of the classes necessary
- or calculate density distribution of trainingset.

Best: trainingset = database.

Problems of VQ

- small codebook size
  $\Rightarrow$



misclassifi cation

- too high codebook size $\Rightarrow$ zero codewords
  (zero codeword = cluster without vectors)
  Solution: Reinitiate codewords with random vectors at the codebook generation

Contents

1 Introduction in VQ
- Quantization
- Vector Quantization
- VQ Advantage

2 VQ Algorithms
- Generalized Lloyd Algorithm (GLA)
- Tree-structured Codebook Representation

3 VQ detailed
- Trainingset
- Problems of VQ

4 **Implementation**
- Range Normalization
- Partial Distance Search

5 Test Results
- first PR-Graph
- Feature Enhencement

Test Implementation

contains:

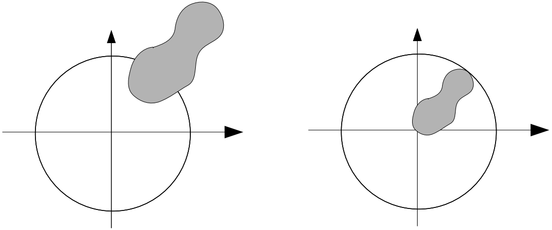
- GLA
- Reinitialization of zero codewords with random vectors
- Normalization over the range for weighting
- Partial Distance Search

Range Normalization

For combining different feature types the feature types have to bee normalized.
To conserve the class shape I used the normalization over the range.

$$\hat{f} = \frac{f - f_{min}}{||f_{max} - f_{min}||}$$

(with $||f_{min}|| \leq ||f||$ and $||f_{max}|| \geq ||f||$)
computed for every feature type



Partial Distance Search

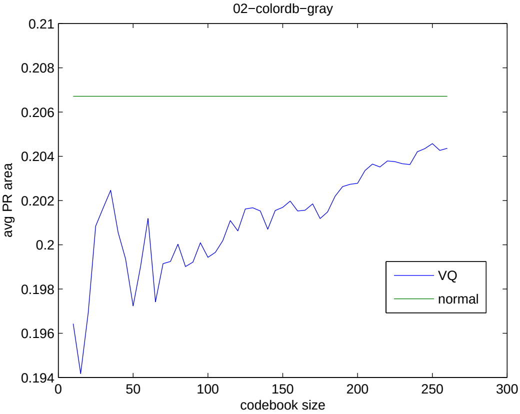
Partial Distance Search

index $= 1$;
$d_{min} = (x - c_i)^2$;
for $i = 2$ to n do
    $d = 0$;
    for $j = 1$ to m do
        $d = d + (x(j) - c_i(j))^2$;
        if $(d > d_{min})$ break; end
    end
!   if $(d < d_{min})$
        $d_{min} = d$;
        index $= i$;
    end
end

Contents

1. Introduction in VQ
   - Quantization
   - Vector Quantization
   - VQ Advantage
2. VQ Algorithms
   - Generalized Lloyd Algorithm (GLA)
   - Tree-structured Codebook Representation
3. VQ detailed
   - Trainingset
   - Problems of VQ
4. Implementation
   - Range Normalization
   - Partial Distance Search
5. **Test Results**
   - first PR-Graph
   - Feature Enhencement

first PR-Graph



Feature Enhencement

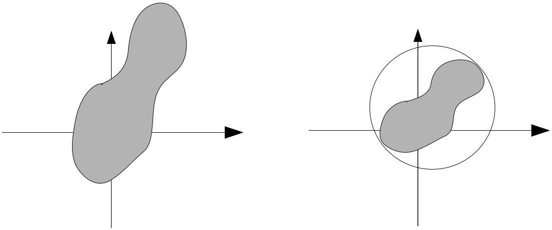
Normalization by the standard deviation of the class

$f = (f_1, \dots, f_n)$

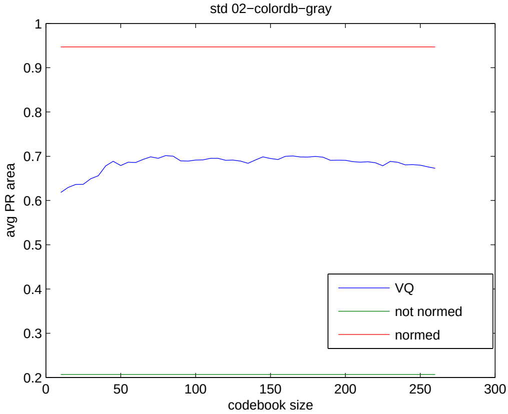
K : feature space $\rightarrow$ class

S : class $\rightarrow \sigma = (\sigma_1, \dots, \sigma_n)$

$$\hat{f} = \left(\frac{f_i}{S(K(f))_i}, \dots, \frac{f_n}{S(K(f))_n} \right)$$



Feature Enhencement



Frequency matrix F

class \ cluster	1	2	3	4
a	0	4	0	0
b	1	2	5	2
c	3	1	0	0
d	0	2	6	3

class a complete in cluster 2
class b is very big

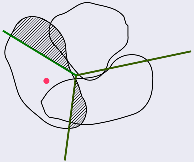
class d lays on b and has high density

The count of features for class x in cluster c

Evaluation 1/2

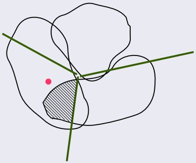
- mis-classification by classes

$$m1(c) = \frac{\sum_i F(c,i) - \max_i(F(c,i))}{\max_i(F(c,i))} = 1 - \frac{\max_i(F(c,i))}{\max_i(F(c,i))}$$



- mis-classification by clusters

$$m2(c) = 1 - \frac{\max_i(F(i,c))}{\max_i(F(i,c))}$$



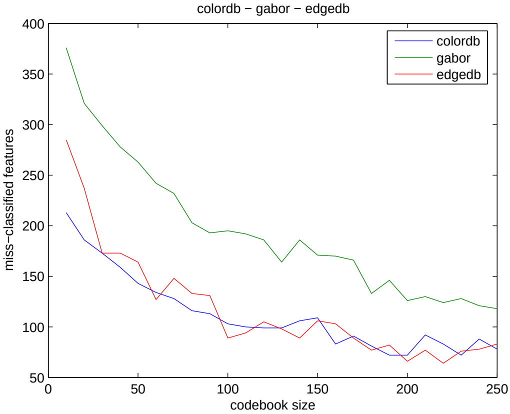
Evaluation 2/2

codebook quality

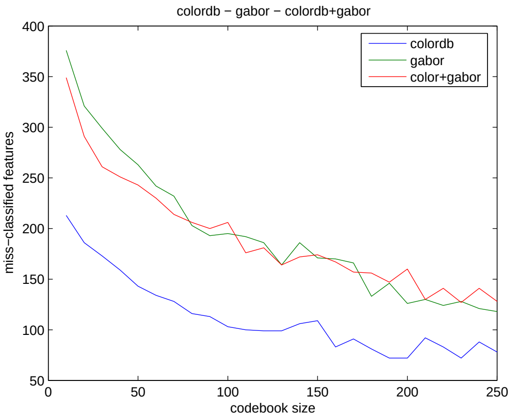
$$\sum_i \left(\left(\sum_j F(j,i) \right) - \max_k F(k,i) \right)$$

(assumption: more cluster than classes)

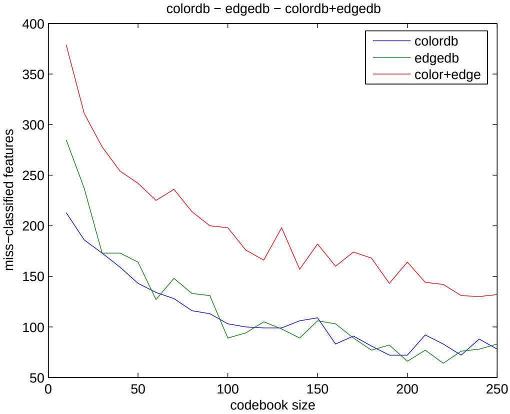
other VQ Evaluation Method



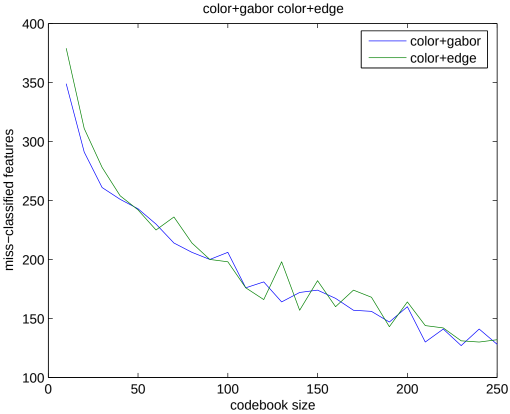
other VQ Evaluation Method



other VQ Evaluation Method



other VQ Evaluation Method



Thank you for audience

References:
[Theodoridis] : ”Pattern Recognition” Sergios Theodoridis, Konstantinos Koutroumbas; Academic Press, 1999
[Cosman] : ”Using Vector Quantization for Image Processing”, P.C. Cosman, K.L. Oehler, E.A. Riskin, R.M. Gray; IEEE 1993
[Zuh] : ”Keyblock: An Approach for Content-based Image Retrieval” L. Zhu, A. Zhang, A. Rao, R. Srihari; SUNY at Buffalo