- New tool for video coding
- Original contribution $\rightarrow$ lattices embedding
- Design of a complete vector quantizer
  - ▷ multistages quantizing method
  - ▷ determination of the optimal lattice
  - ▷ labeling of the codebook points
  - ▷ processing of the outlying source vectors
  - ▷ bit allocation

Perspectives

- Progressive image coding
- Codebook updating (adaptive coding)

Plan

1. Context of the study
2. Vector Quantization (VQ)
3. Lattice VQ and Tree-Structured VQ
4. Tree-Structured Lattice VQ (TSLVQ)
5. Experimental results
6. Conclusion

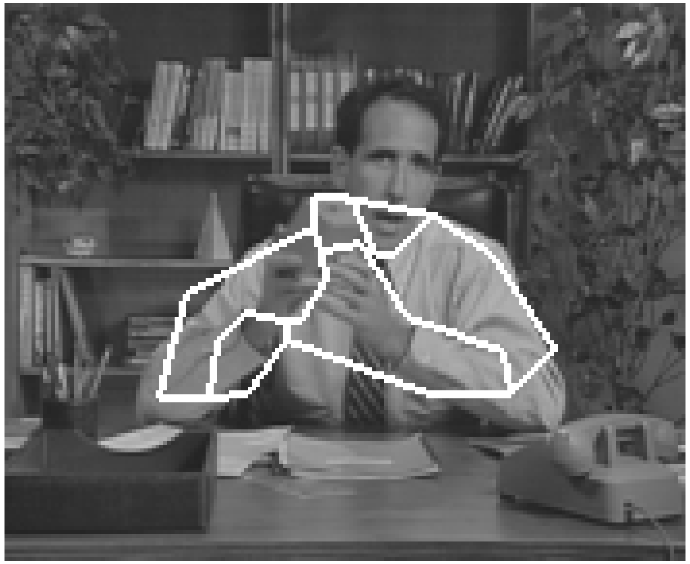
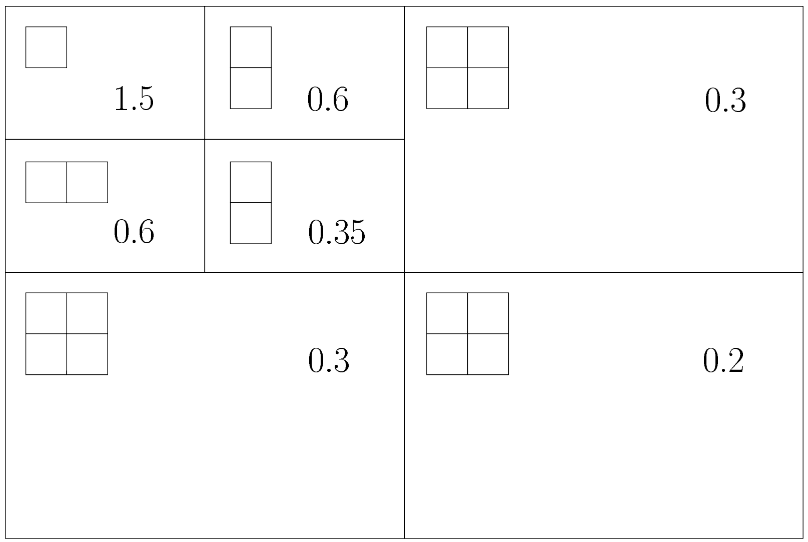
 

Image sequences coding

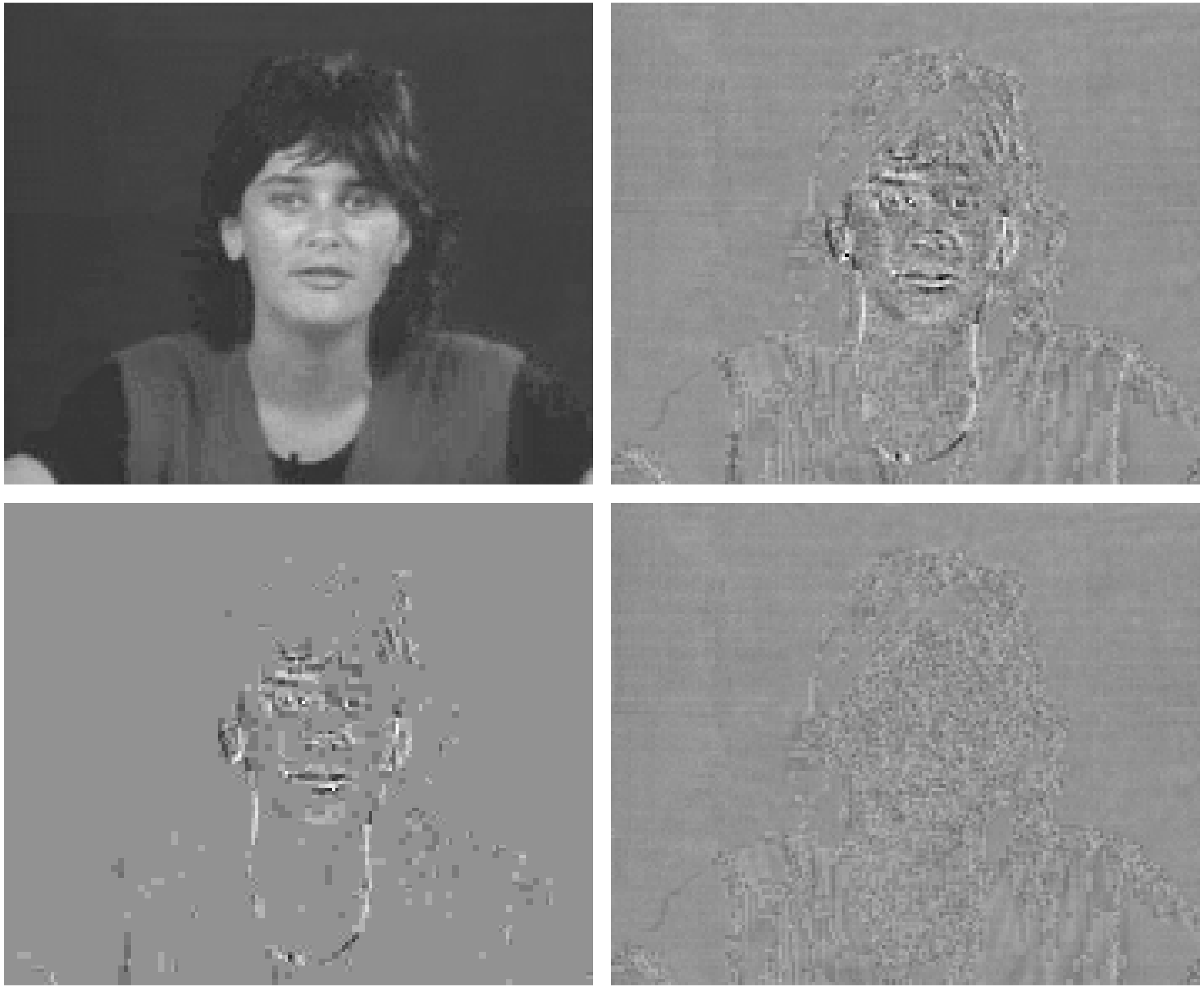
image sequence	image number	PSNR [dB]	entropy [bpp]	maximal time of encoding [s/image]
Salesman	200	33.86	0.238	1.5
MissAmerica	107	39.38	0.064	1.3

Region-based coder

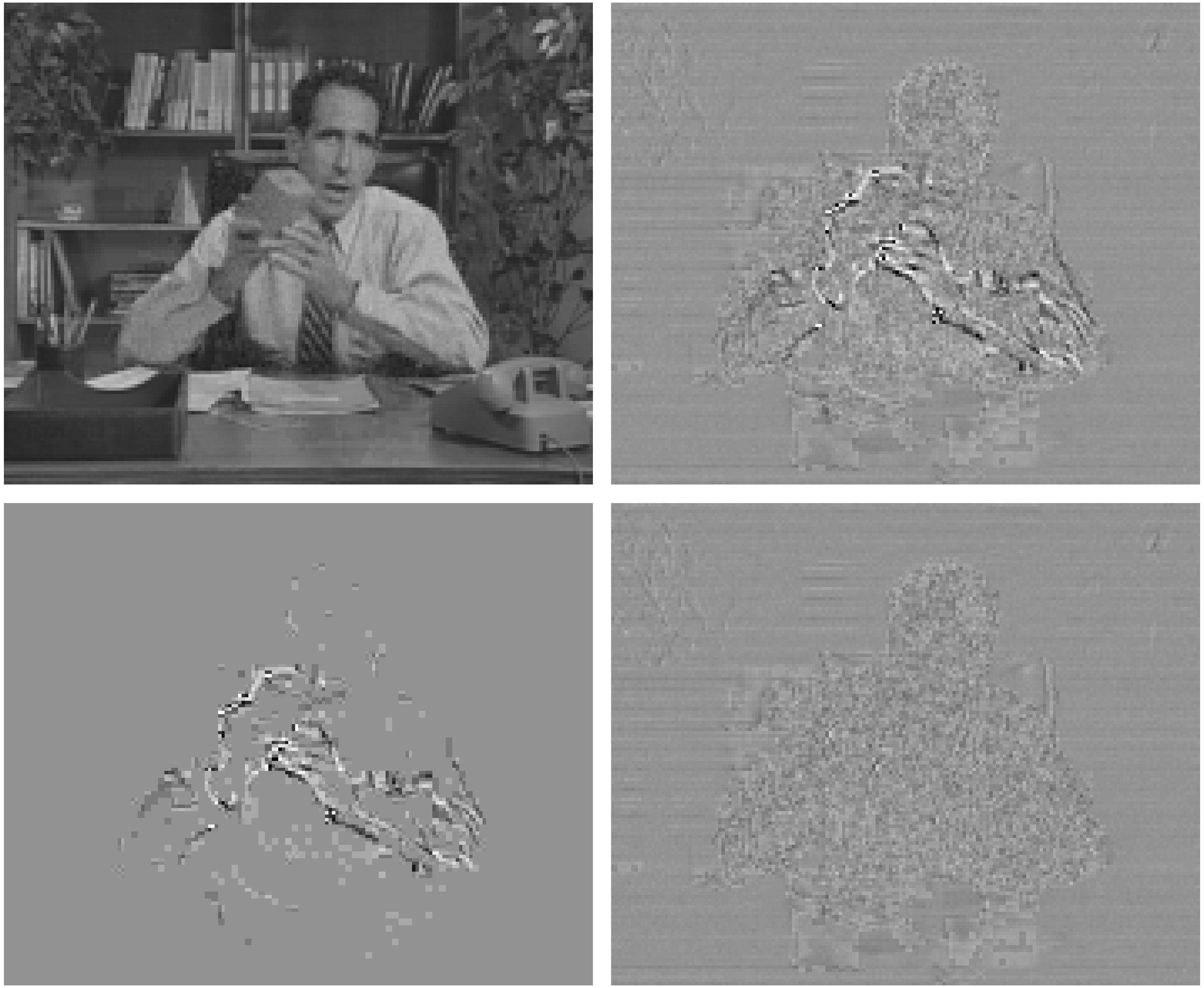
- Very low bit rate
- Motion estimation
  → polygonal shapes (Nzomigni95, Pateux96)
- Dyadic wavelet transform (Mallat89, Daubechies88)
- Multiresolution codebook (Antonini91)
- Bit allocation → thresold 0.2 bpp [final rate 0.175 bpp]



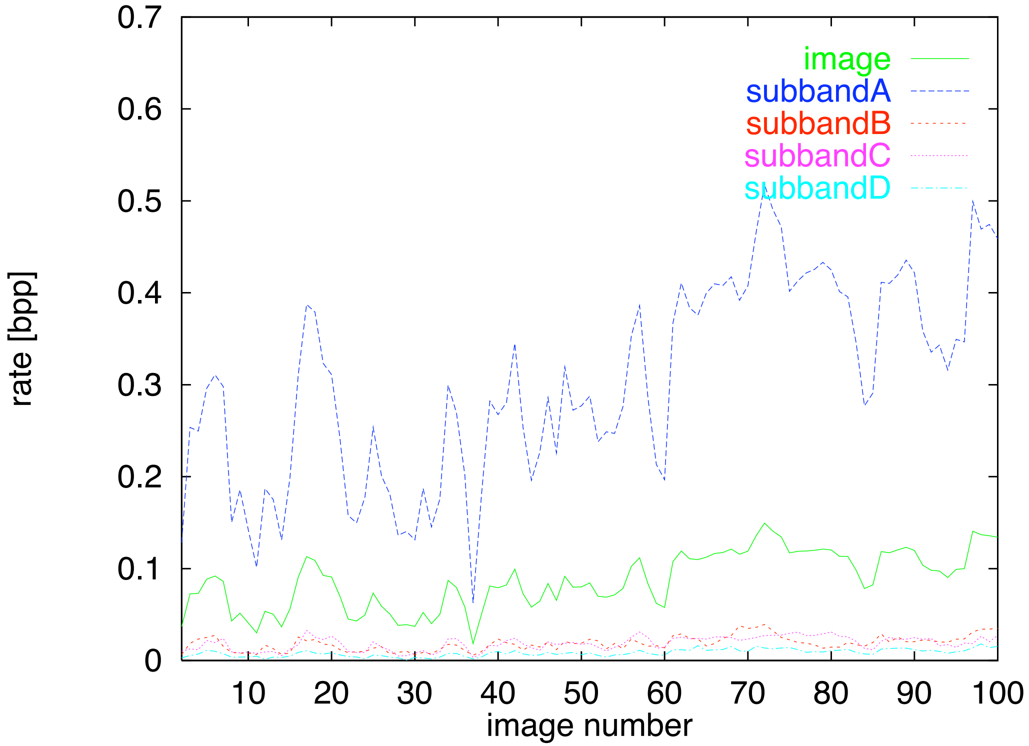
Coding of the image sequence “MissAmerica”

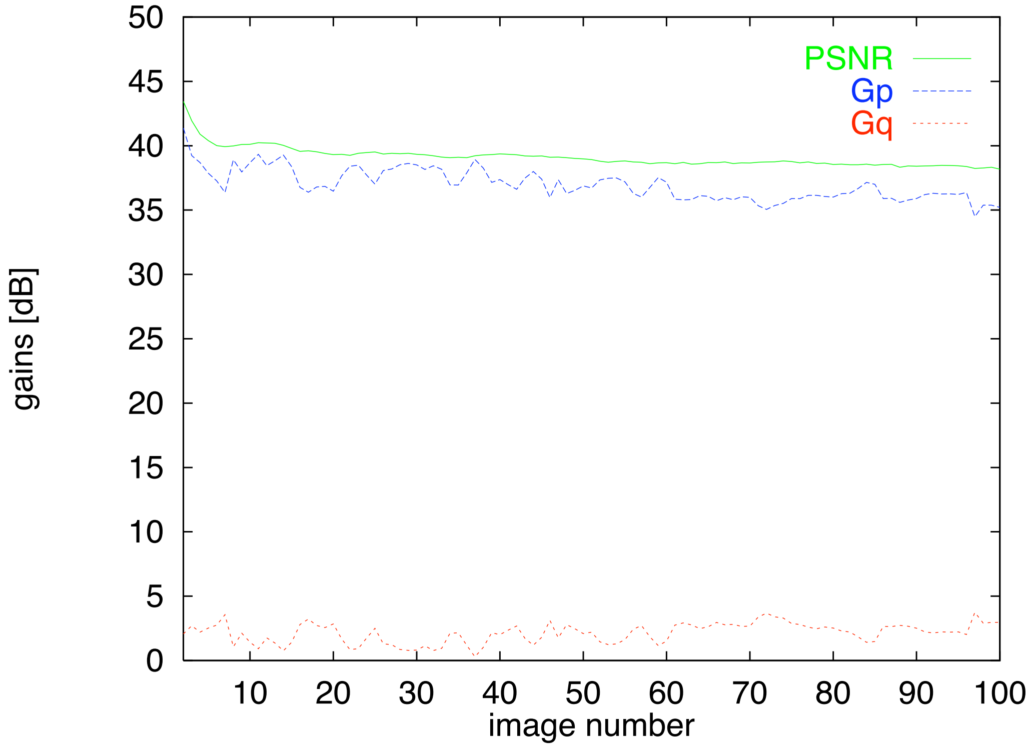


Coding of the image sequence "Salesman"

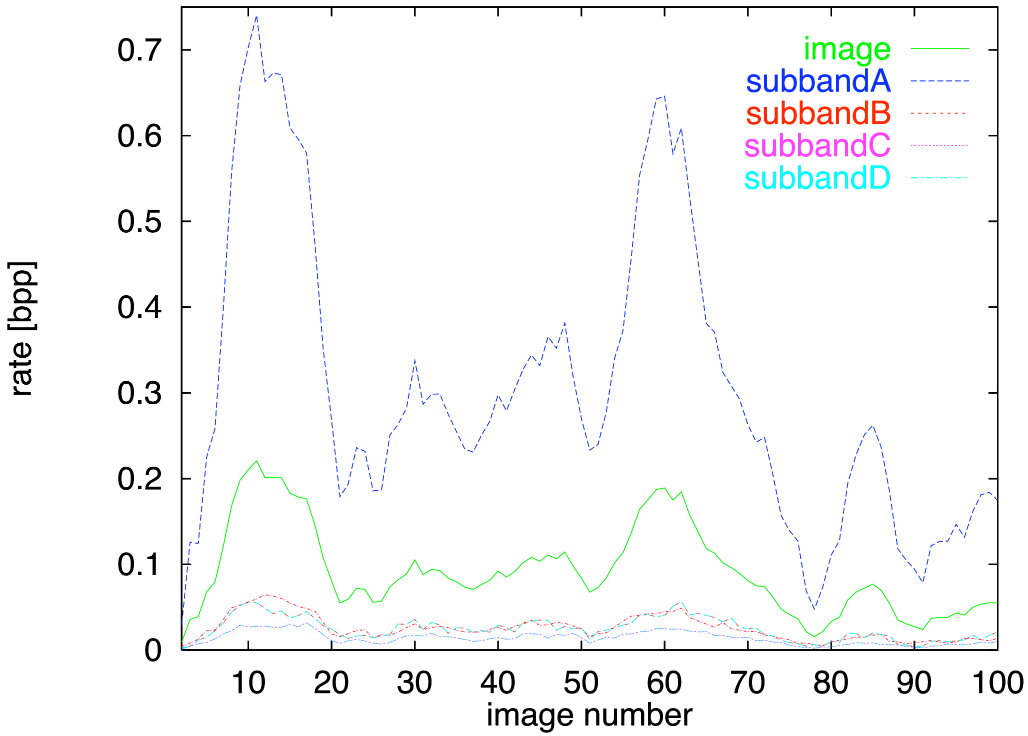


Coding of the image sequence “MissAmerica”





Coding of the image sequence “Salesman”



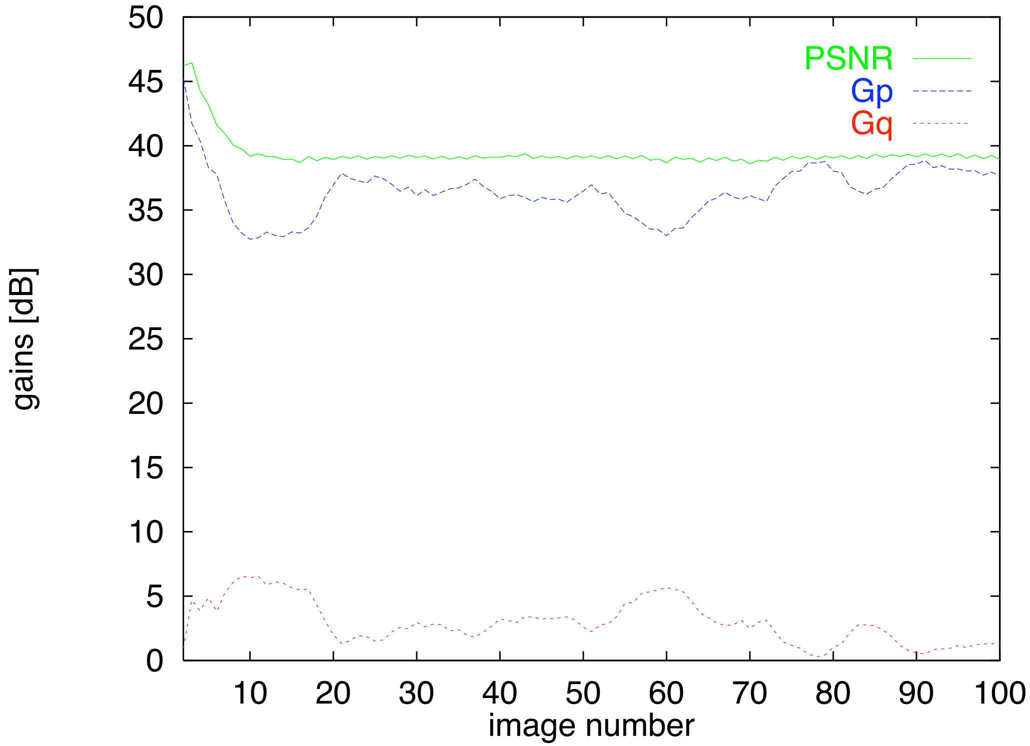


Image sequences coding

image sequence	number of images	PSNR [dB]	entropy [bpp]	maximal time of encoding [s/image]
Salesman	200	39.07	0.201	2
MissAmerica	107	38.98	0.162	2
Claire	200	38.03	0.157	2

Codebooks design (before bit allocation)

Image sequence "Salesman"

subband label	training sequence size	cpu time [s]	number of code vectors	entropy [bpp]	training ratio
A	5 images	6.75	43	0.992	884
B	10 images	13.95	358	0.427	186
C	10 images	14.13	434	0.496	153
D	148 images	48.50	1248	0.087	189

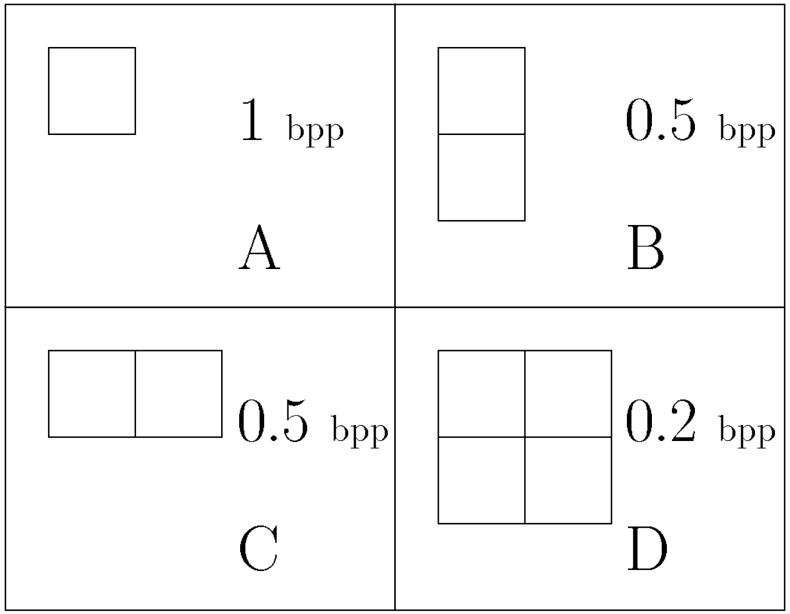
Bit allocation

Threshold 0.2 bpp $\rightarrow$ final rate 0.188 bpp

subband label	number of code vectors	entropy [bpp]
A	19	0.416
B	64	0.111
C	108	0.134
D	1234	0.086

MPEG-based coder

- Motion estimation → "block matching"
- DCT 2x2, intra-band configuration
- Codebooks



- 4 steps
  - ▷ training sequences → open loop coder
  - ▷ codebooks design
  - ▷ bit allocation
  - ▷ image sequence coding → closed loop coder
- Formulae
  - ▷ $PSNR = 10.\log_{10} \; 255^2 / \left(\frac{1}{N_x.N_y}.d(e, e_q)\right)$
  - ▷ prediction gain $G_p = 10.\log_{10} \; 255^2 / \left(\frac{1}{N_x.N_y}.d(e)\right)$
  - ▷ quantization gain $G_q = 10.\log_{10} \; d(e)/d(e, e_q)$
  - ▷ codebook entropy
- QCIF image sequence
- Sparc-Station 5 [110 Mhz] computer

Plan

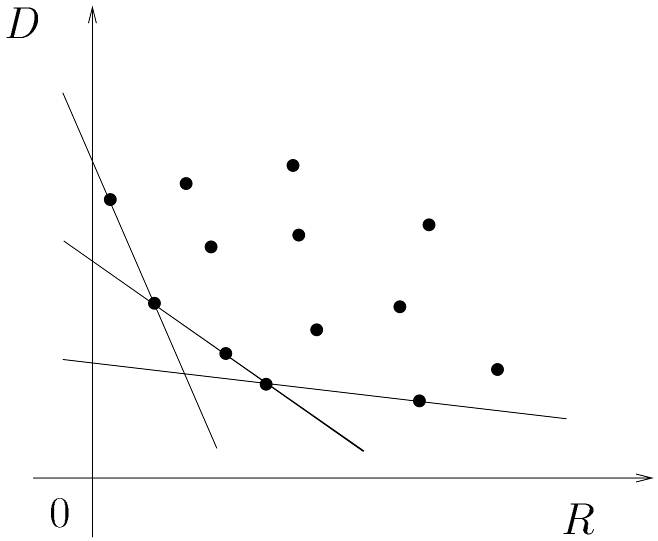
1. Context of the study
2. Vector Quantization (VQ)
3. Lattice VQ and Tree-Structured VQ
4. Tree-Structured Lattice VQ (TSLVQ)
5. Experimental results
6. Conclusion

Conclusion

- Partition of the space according to :
  - ▷ the source distribution
  - ▷ the rate vs. distorsion tradeoff
- Simple labeling method
- Z^k → simple processing for the outlying source vectors
- Fast quantizing → complexity $O(h.k)$
- Bit allocation

Optimal bit allocation

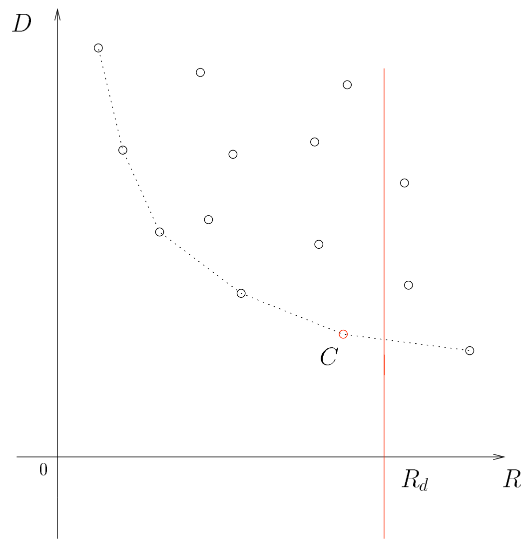
Singular value of λ (Shoham88)



- From a first point of the hull,
  by successive calculations of singular values $\rightarrow$ global convex hull
- $\lambda \rightarrow$ search the BFOS criterion with the maximal value
  among the subbands

Lagrange multiplier

- N^M combination of quantizers → complex
- $\min(D + \lambda.R) \iff \sum_{j=0}^{M-1} \min_{q_{j,i}} \left(d_{j,i} + \lambda.r_{j,i}\right)$
- Algorithm (general form)
  1. convex hull of each subband → directly obtained when growing the tree
  2. global convex hull → search C



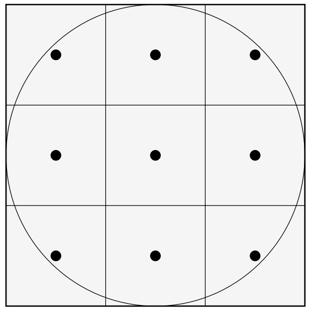
Bit allocation

- M subbands
- $\min D$ subject to $R \leq R_d$
- Lagrangian methods (Shoham88, Ramchandran93)
  - ▷ N quantizers $q_{j,i}$ for each subband j
    $\rightarrow$ different configurations of the tree
  - ▷ for one combination of quantizers

$$D = \frac{1}{M} . \sum_{j=0}^{M-1} d_{j,i} \quad \text{and} \quad R = \frac{1}{M} . \sum_{j=0}^{M-1} r_{j,i}$$

  - ▷ cluster of points $\rightarrow$ search on the convex hull

Processing of a source vector whose energy is too large



- Detection
  - ▷ $F = b_{min}.\rho/\sqrt{\mathcal{E}_{max}}$
  - ▷ L_∞ norm of a vector $\mathbf{u}$ within the cube :

$$L_\infty(\mathbf{u}) = \max_{i=1,\ldots,k} |u_i| \;\leq\; (b_{min} \times \rho)$$

  - ▷ L_∞ norm of a vector $\mathbf{x}$ which can be quantized :

$$L_\infty(\mathbf{x}) = \max_{i=1,\ldots,k} |x_i| \;\leq\; \sqrt{\mathcal{E}_{max}}$$

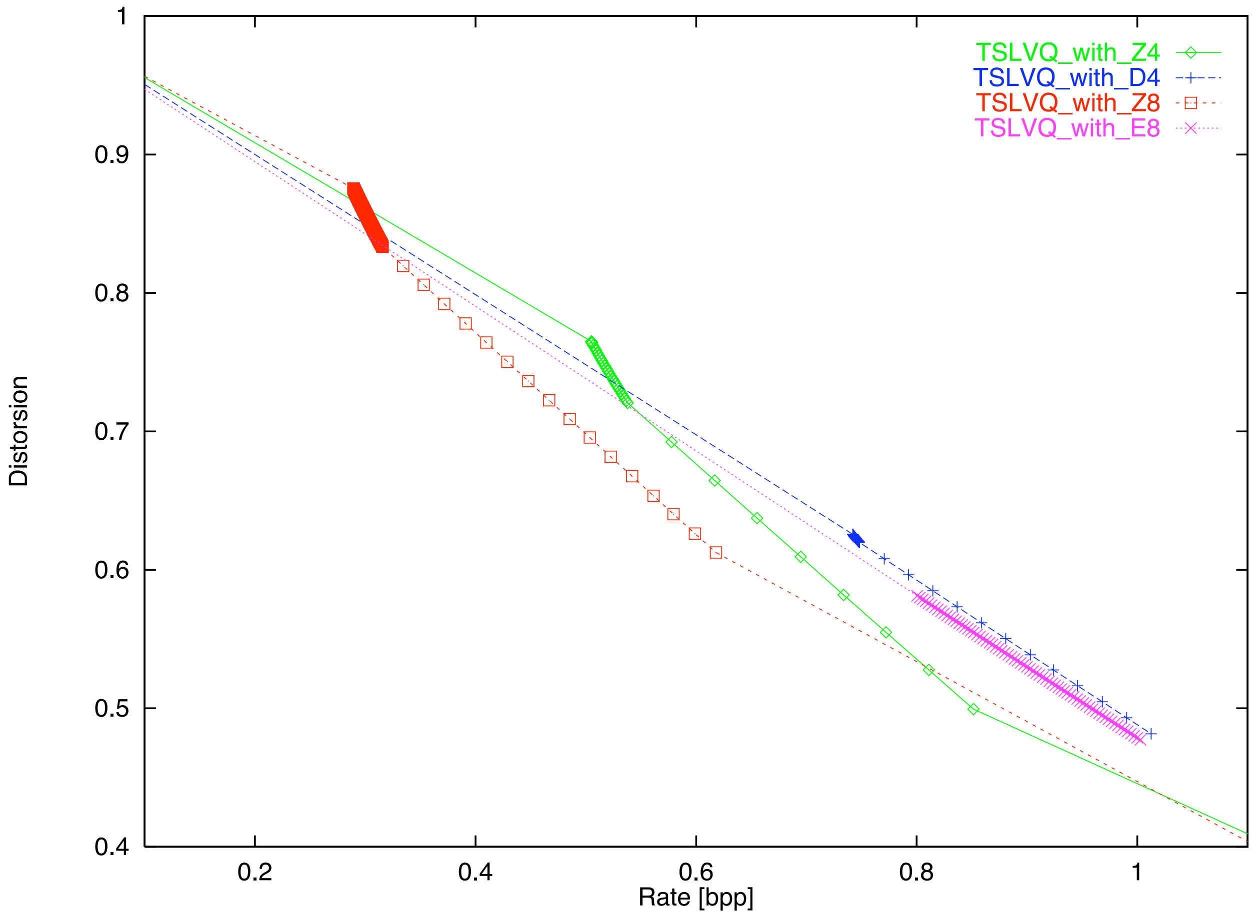
- Processing

$$\text{If} \quad |x_i|_{i=1,\ldots,k} > \sqrt{\mathcal{E}_{max}} \implies x_i = \mathsf{sign}(x_i).\sqrt{\mathcal{E}_{max}}$$

Labeling of the codebook vectors

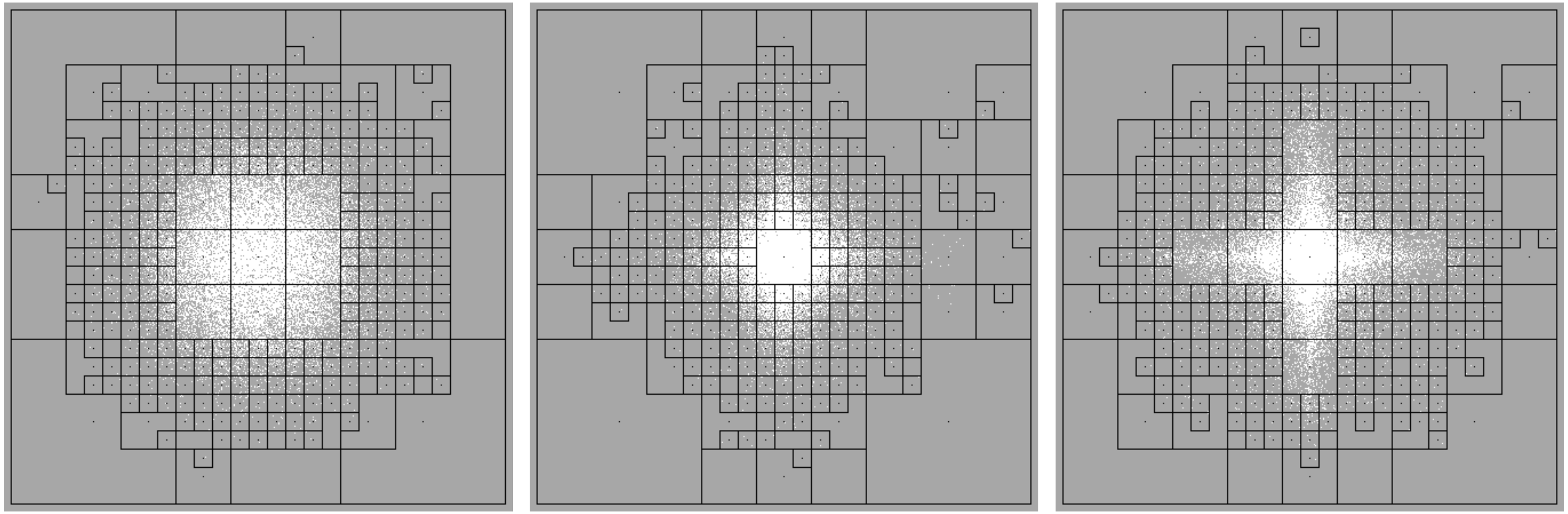
1. Look-up table $\rightarrow$ index of the truncated lattice points
2. Scan the tree in order to number the nodes
3. Re-scan the tree and store for each node :
   - the children numbers
   - the father number
   - the index of the corresponding lattice point
   - the entropy code word (for the leaves)

Optimal lattice



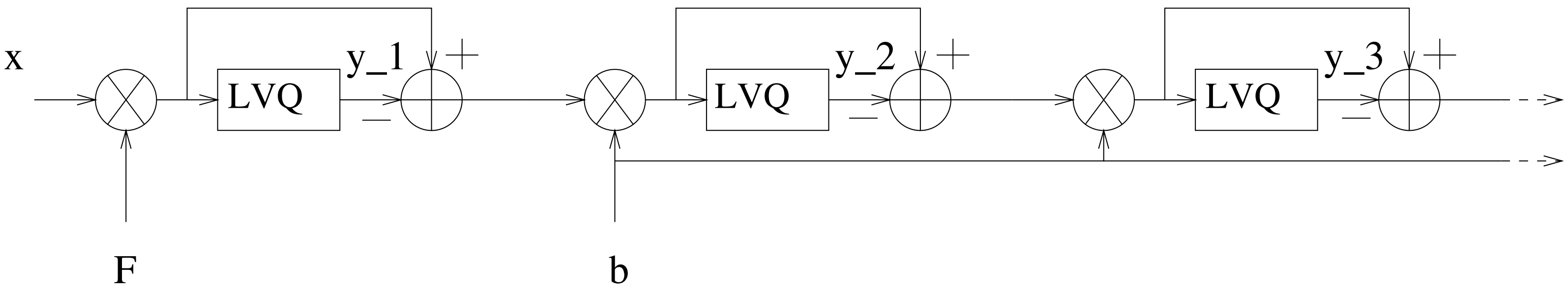
$\Longrightarrow Z^k$ is optimal

Unbalanced tree design (greedy approach)

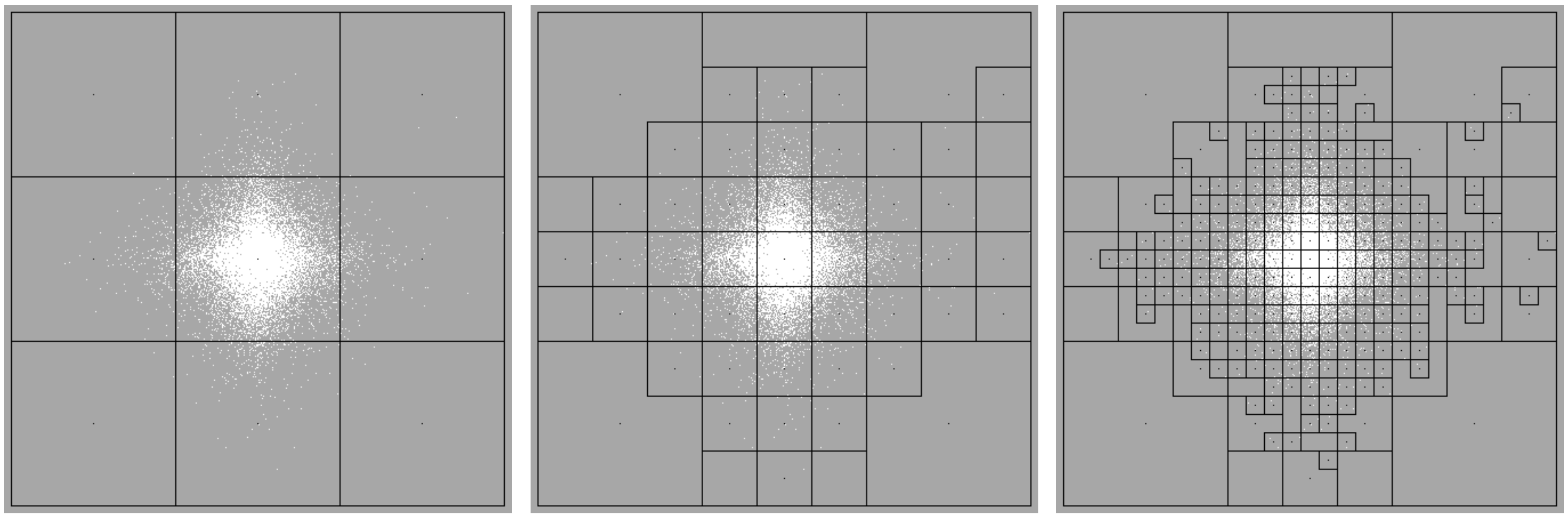


- Partition adapted to the source distribution
- “Dead zone”

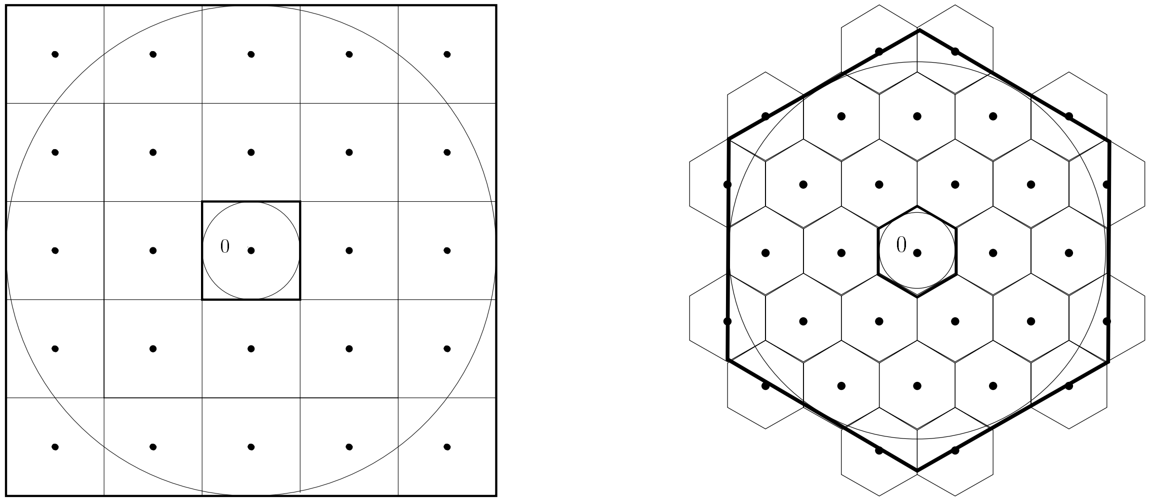
Quantization scheme



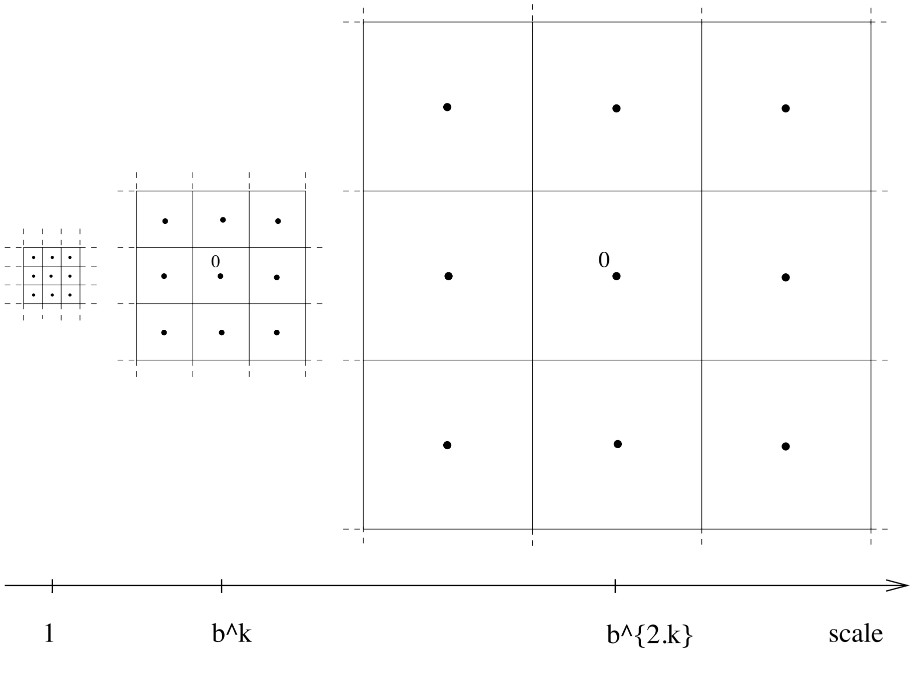
$F = b.\rho/\sqrt{\mathcal{E}_{max}}$

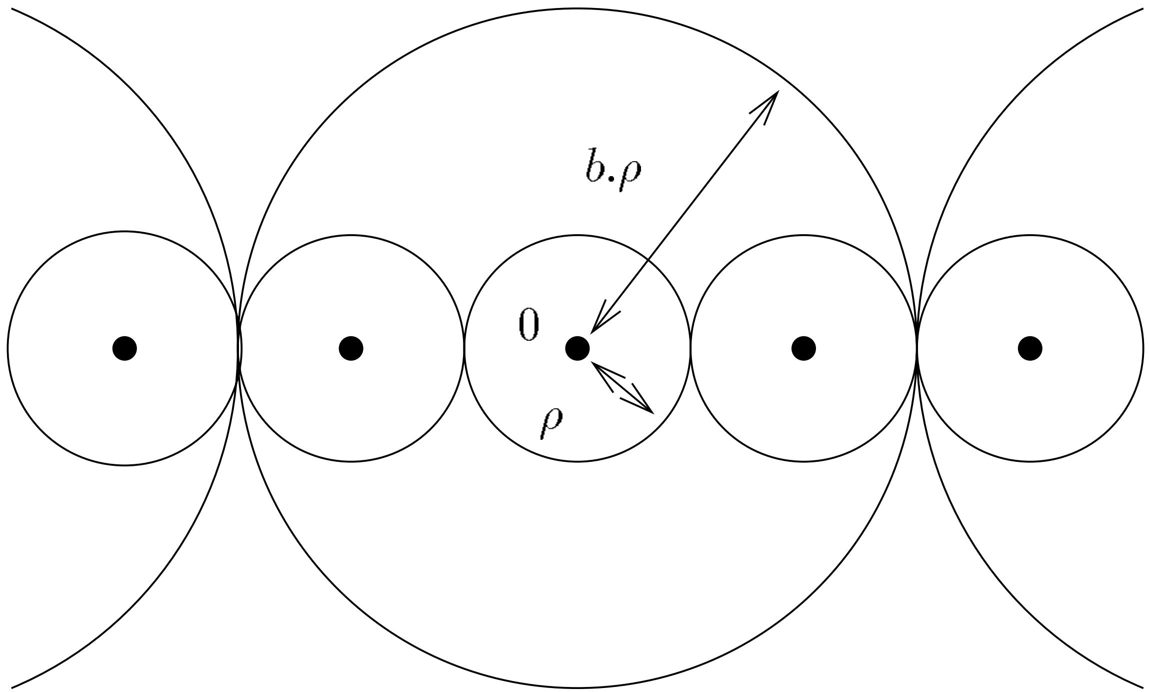


- A tree-structured codebook
- Progressive splitting $\rightarrow b = b_{min} = 3$



Hierarchy of embedded lattices





- Packing radius of the support lattice : ρ
- Packing radius of the dilated lattice :

$$b.\rho \text{ avec } b \in \mathbf{R} \ / \ b > 1$$

$$\Longrightarrow \qquad b = 2.n + 1 \ / \ n \in N^*$$

Embedded Lattices

- Support Lattices Z^k, D_k, E_8, Λ_{16} $\rightarrow$ fastest quantizing algorithms
- **Embedding** :
  by contracting it, embed a truncated lattice in its Voronoï cell
- **Optimal embedding** :
  the rescaled truncated lattice covers exactly the Voronoï cell
- **Sub-optimal embedding** :
  the rescaled truncated lattice covers maximaly the Voronoï cell

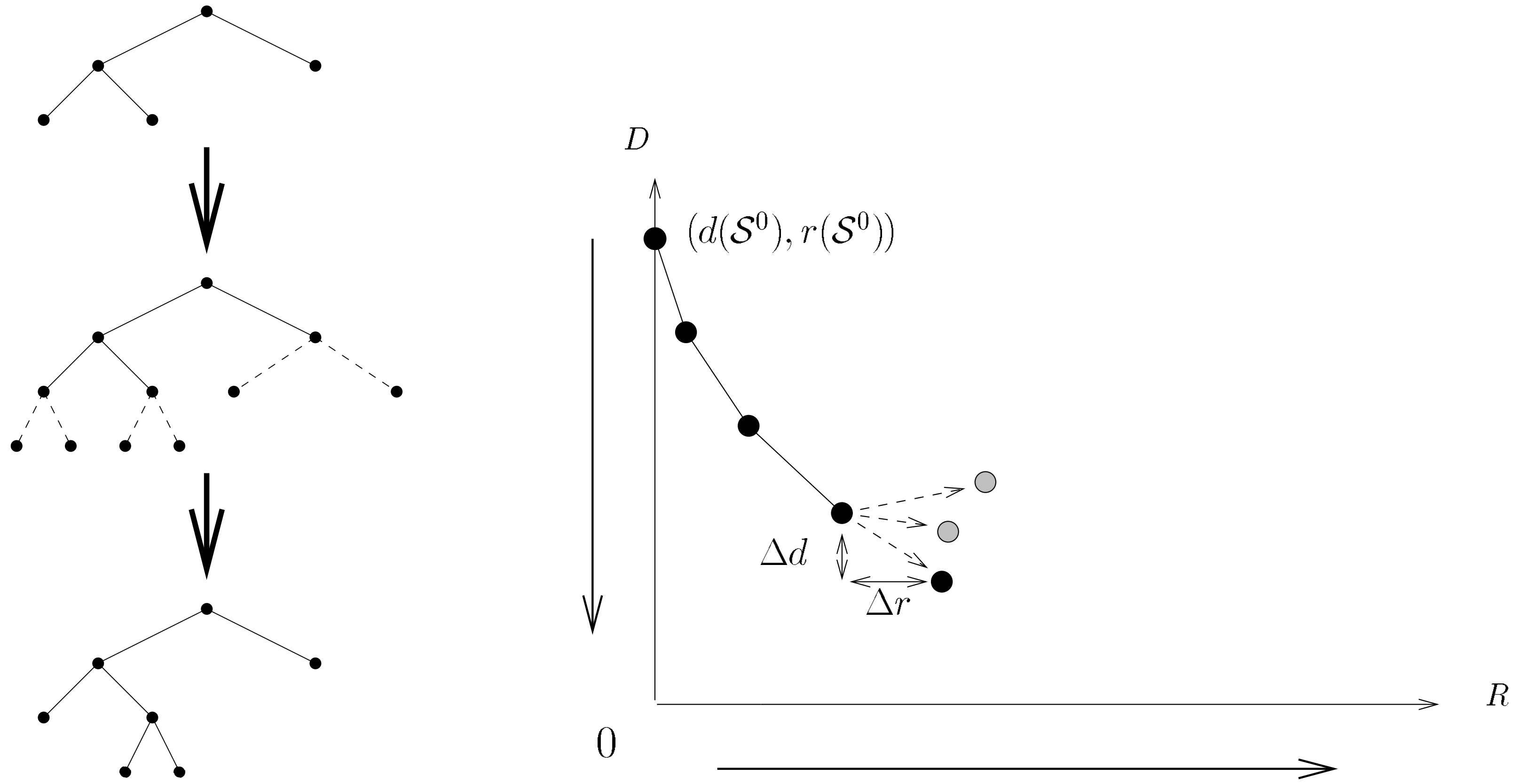
Plan

1. Context of the study
2. Vector Quantization (VQ)
3. Lattice VQ and Tree-Structured VQ
4. Tree-Structured Lattice VQ (TSLVQ)
5. Experimental results
6. Conclusion

Conclusion

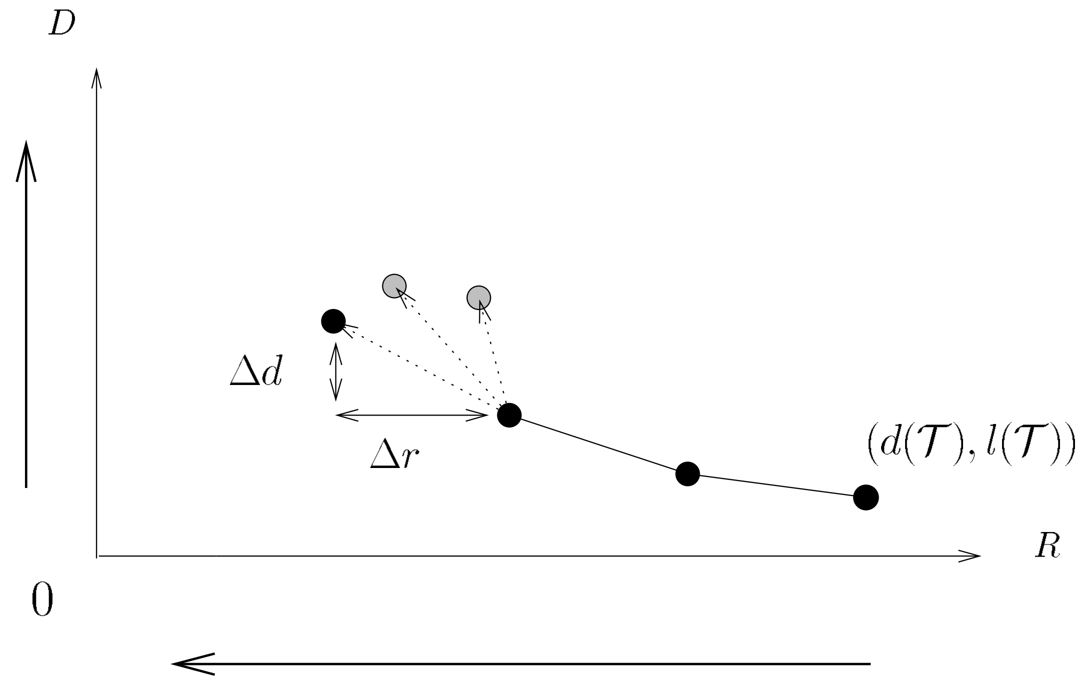
- Pruning algorithm
  - ▷ global approach
  - ▷ storage of the complete tree
- Greedy algorithm
  - ▷ local approach
  - ▷ limited storage

Greedy approach



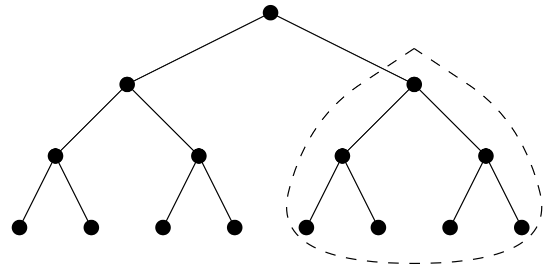
$\Longrightarrow$ Splitting of the leaf for which $\lambda(n_i)$ is maximal

Pruning principle



$\Longrightarrow$ Pruning of the branch for which $\lambda(n_i)$ is minimal

Tree pruning



- BFOS algorithm (Breiman84)
  1. complet tree $\mathcal{T}$
  2. successive pruning
- Characterisation of each branch $\mathcal{S}_{n_i}$
  - ▷ increase in distorsion if $\mathcal{S}_{n_i}$ is removed $\Delta d(\mathcal{S}_{n_i})$
  - ▷ decrease in rate if $\mathcal{S}_{n_i}$ is removed $\Delta r(\mathcal{S}_{n_i})$
  - ▷ **BFOS criterion** $\lambda(n_i) = \Delta d(\mathcal{S}_{n_i}) / \Delta r(\mathcal{S}_{n_i})$

Training

Characterisation of each node n_i

- Probability of reaching n_i

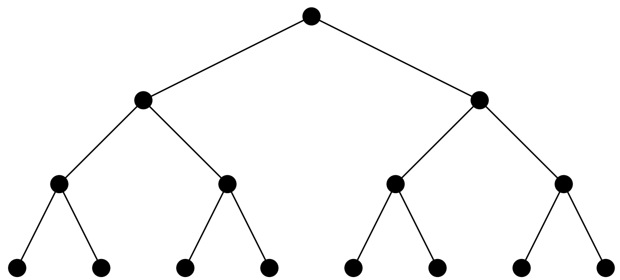
$$P(n_i) = \frac{\mathsf{card}(C_{n_i})}{\mathsf{card}(SA)}$$

- Average distortion

$$d(n_i) = \frac{1}{\mathsf{card}(C_{n_i})} \cdot \sum_{\mathbf{x} \in C_{n_i}, \mathbf{x} \in SA} \|\mathbf{x} - \mathbf{y}_{n_i}\|^2$$

- Entropy code length

$$r(n_i) = -\mathsf{log}_2 P(n_i)$$



- Encoding
  - ▷ complexity $O(\log_B L)$
  - ▷ tree storage
- Decoding
  - ▷ leaves
  - ▷ progressive reconstruction
- Unbalanced tree $\rightarrow$ variable rate
  - ▷ pruning approach (Breiman84, Chou89)
  - ▷ greedy approach (Makhoul85, Riskin91)

Plan

1. Context of the study
2. Vector Quantization (VQ)
3. Lattice VQ and Tree-Structured VQ
4. Tree-Structured Lattice VQ (TSLVQ)
5. Experimental results
6. Conclusion

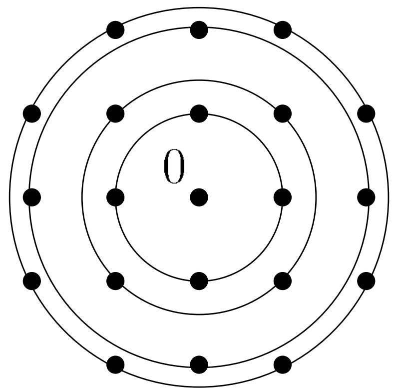
Conclusion

- Advantage
  - ▷ fast quantization
  - ▷ predefined codebook
- Drawback $\rightarrow$ method for simple sources

$\Longrightarrow$ TSLVQ : hierarchical packing of embedded truncated lattices

Labeling of the lattice points

- Index calculation $\rightarrow$ product code (Lamblin88, Moureaux94, Onno95)
  - ▷ sub-index for the sphere energy
  - ▷ sub-index for the point position



Projection within a sphere

- Sphere radius $\sqrt{\mathcal{E}_t}$
- Training sequence $\mathcal{SA} = \left\{ \mathbf{x}_j = (x_1, \ldots, x_k)^T \;\; / \;\; j = 0, 1, 2, \ldots \right\}$
- Vector energy $\mathcal{E}(\mathbf{x}) = L_2(\mathbf{x})$
- $\mathcal{E}_{max} = \max_{\mathcal{E}} \{ \mathcal{E}(\mathbf{x}) \;\; / \;\; \mathbf{x} \in \mathcal{SA} \}$
- Scaling factor $F = \sqrt{\mathcal{E}_t / \mathcal{E}_{max}}$

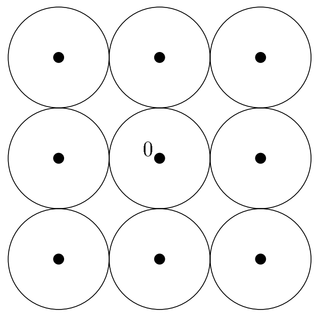
- Real source $\rightarrow$ vectors with energy greater than $\mathcal{E}_{max}$
  are processed separately

Coding scheme

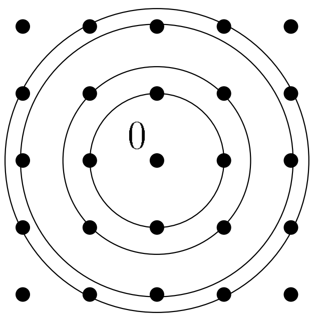
- Lattice Truncation
  - ▷ Shape
    - ▷ i.i.d Gaussian source → sphere (L_2)
    - ▷ i.i.d Laplacian source → pyramid (L_1)
    - ▷ correlated GG source → ellipse (ponderated L_2)
  - ▷ Truncation energy (Fisher86) $\mathcal{E}_t \to$ points number
- Source normalisation
  - ▷ before quantization
  - ▷ scaling factor

Characteristics

- Packing radius ρ



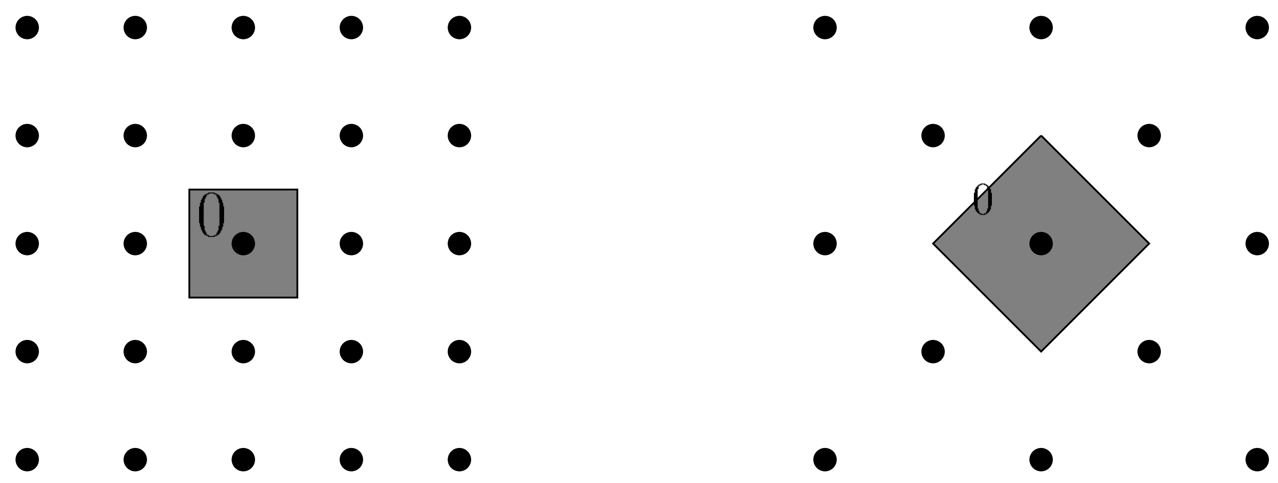
- Series Theta, Nu (Gaidon93), modified Theta (Moureaux94)



- Best quantizing lattices : A_2, D_4, E_8, Λ_{16}
- Fast quantizing lattices (Conway and Sloane83) : Z^k, D_k, E_8, Λ_{16}
  $\Longrightarrow$ complexity $O(k)$

Lattices Λ

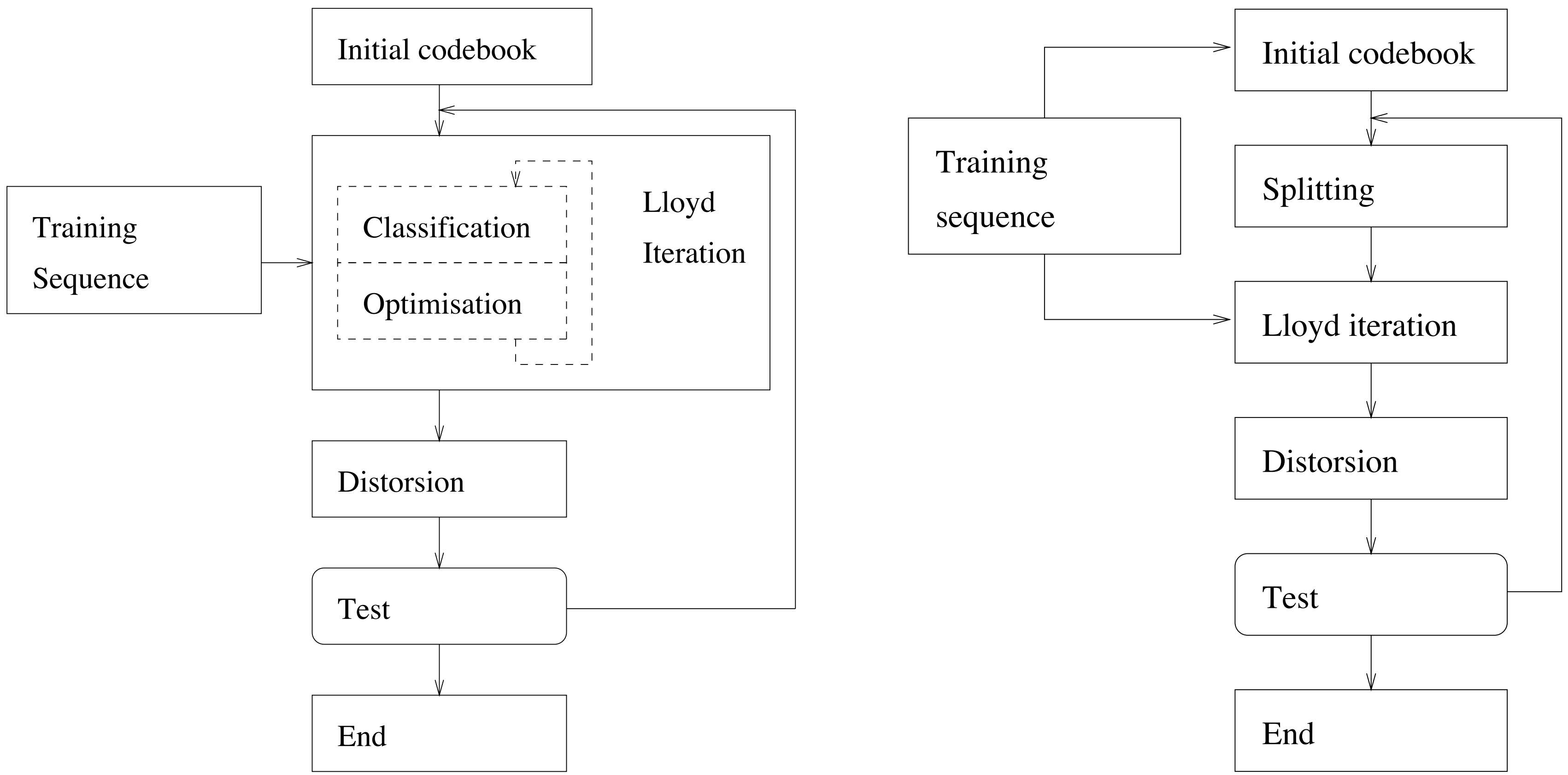
- Regular arrangement of identical spheres $\rightarrow$ spheres centers
  $0 \rightarrow$ origin
- $Z^k = \left\{ \mathbf{y} = (y_1,\ y_2,\ \ldots,\ y_k)^T \ / \ y_i \in Z \right\}$
- $D_k = \left\{ \mathbf{y} = (y_1,\ y_2,\ \ldots,\ y_k)^T \ / \ \mathbf{y} \in Z^k,\ \Sigma_{i=1}^{k} y_i = 0 \text{ mod}(2) \right\}$



Plan

1. Context of the study
2. Vector Quantization (VQ)
3. Lattice VQ and Tree-Structured VQ
4. Tree-Structured Lattice VQ (TSLVQ)
5. Experimental results
6. Conclusion

(nearly) Optimal VQ



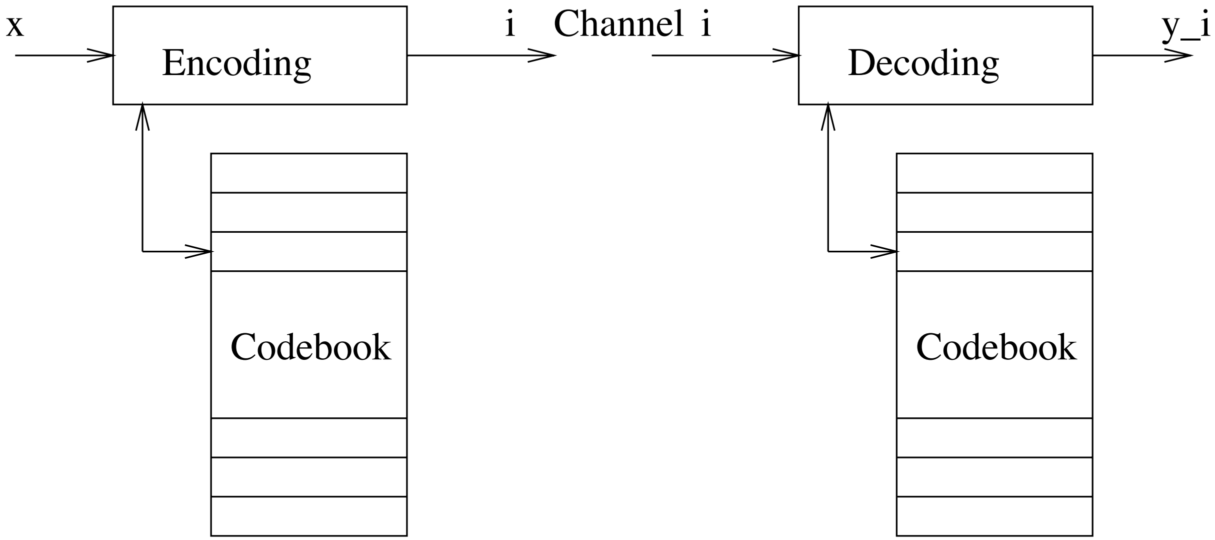
- Training
- Encoding $\rightarrow$ exhaustive research $O(L)$

Performance evaluation

- Rate [bpp]
  - ▷ Rate constrained : $R = \frac{1}{k}.\log_2 L$
  - ▷ Entropy constrained : $R \simeq H(\mathcal{D})$
- Distorsion

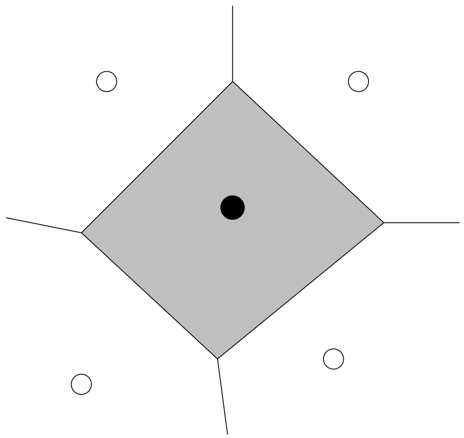
$$D = \frac{1}{k} \sum_{i=1}^{L} \int_{C_i} L_2(\mathbf{x}, \mathbf{y}_i).p_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x}$$

- Coding



- Encoding, “Nearest Neighbour” rule

$$C_i = \left\{ \mathbf{x} \in R^k / \mathcal{Q}(\mathbf{x}) = \mathbf{y}_i,\ si\ d(\mathbf{x}, \mathbf{y}_i) \leq d(\mathbf{x}, \mathbf{y}_j), \forall j \neq i \right\}$$



Principe

- Vector Quantizer, dimension k, size L

$$\begin{aligned} \mathcal{Q} : R^k &\longrightarrow \mathcal{D} \\ \mathbf{x} &\longmapsto \mathcal{Q}(\mathbf{x}) = \mathbf{y}_i \end{aligned}$$

$$\mathcal{D} = \left\{\mathbf{y}_i \in R^k / i = 1, 2, ..., L\right\}$$

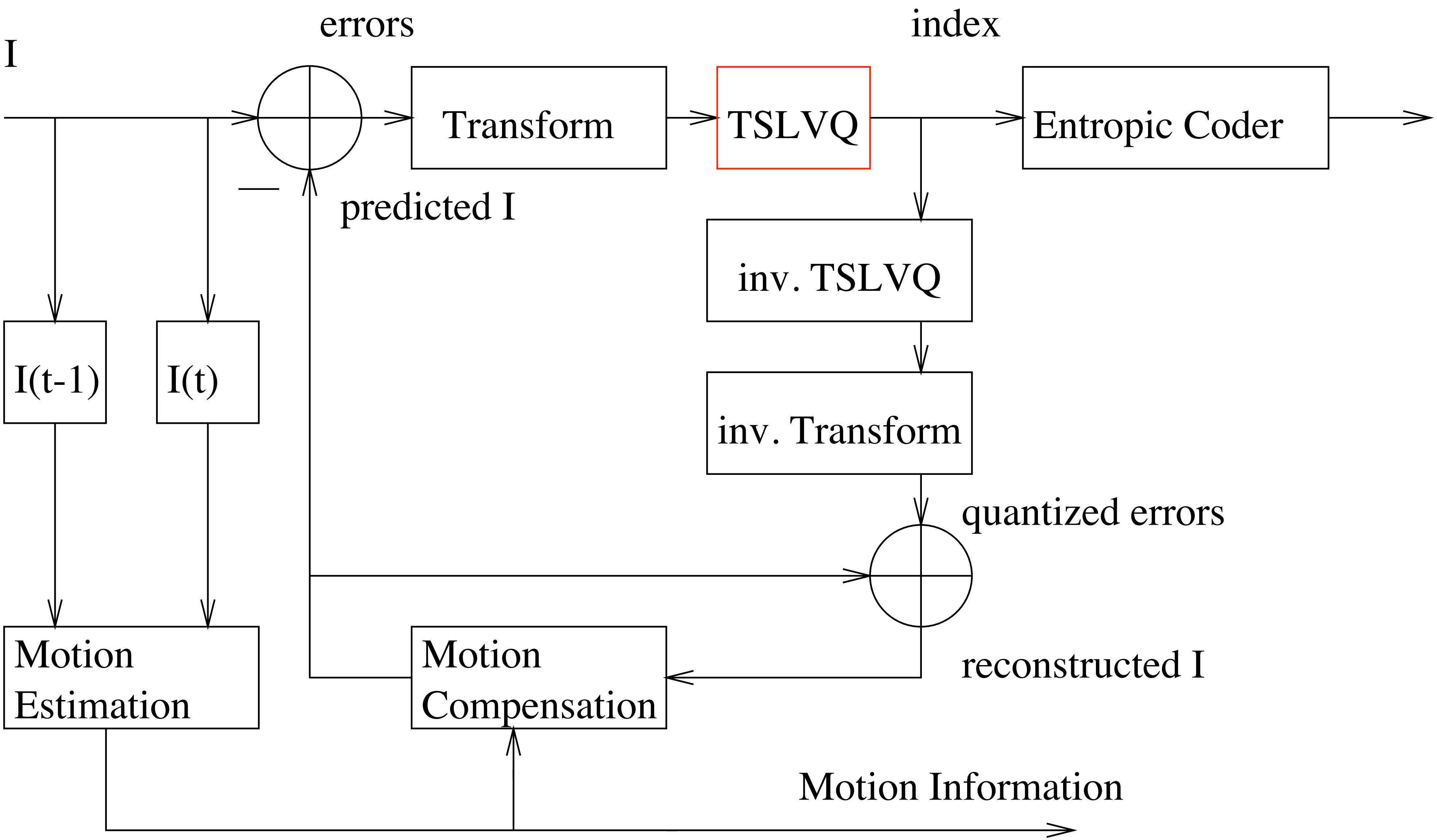
- R^k partition into L Voronoï cells

$$C_i = \left\{\mathbf{x} \in R^k / \mathcal{Q}(\mathbf{x}) = \mathbf{y}_i\right\}$$

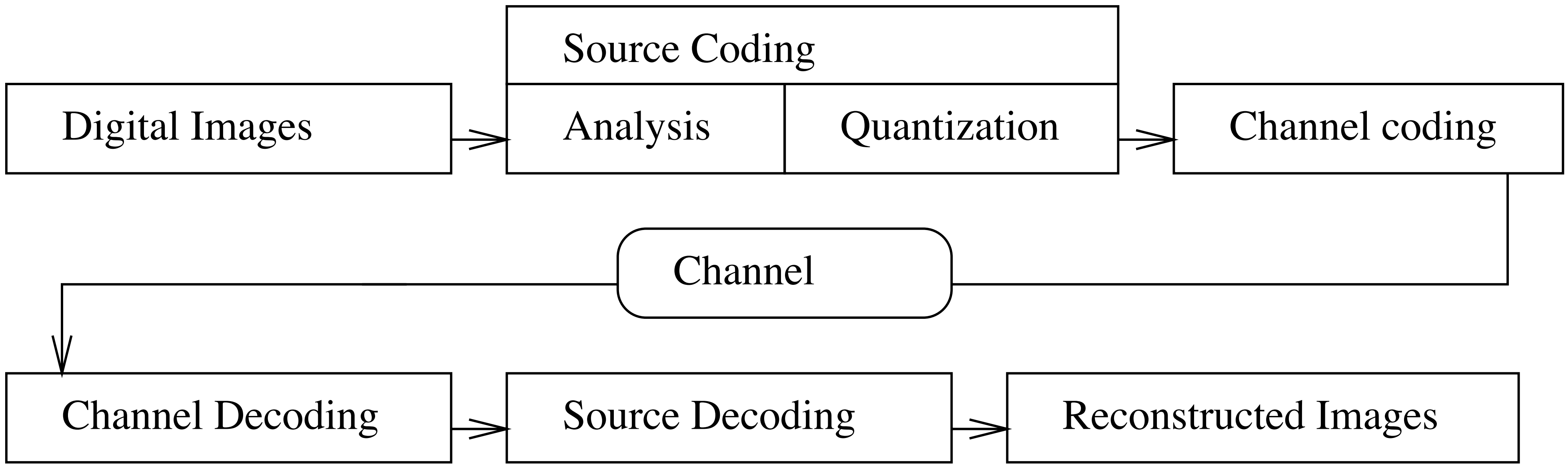
Plan

1. Context of the study
2. Vector Quantization (VQ)
3. Lattice VQ and Tree-Structured VQ
4. Tree-Structured Lattice VQ (TSLVQ)
5. Experimental results
6. Conclusion

- Hybrid approach



- Standarts design (H261, MPEG1&2, MPEG4)
- Image coding scheme



Plan

Tree-Structured Lattice Vector Quantization for the Compression of Digital Image Sequences

Vincent Ricordel

Tampere University of Technology

Signal Processing Laboratory