# Contents

**1 Introduction and Motivation 1**

1.1 Digital communication systems and compression . . . . . . . . . . . . 1

1.2 Performance of image and video encoders . . . . . . . . . . . . . . . . 3

1.2.1 Codebook design . . . . . . . . . . . . . . . . . . . . . . . . 5

1.2.1.1 Scalar quantization and entropy coding . . . . . . . . 6

1.2.1.2 Vector quantization . . . . . . . . . . . . . . . . . . 7

1.2.1.3 Codebook design for image and video sources . . . . 8

1.2.2 Encoding algorithms and adaptivity . . . . . . . . . . . . . . . 10

1.3 Adaptivity, bit allocation and delay constraints . . . . . . . . . . . . 13

1.4 Optimization framework . . . . . . . . . . . . . . . . . . . . . . . . 15

1.4.1 Defining a cost function . . . . . . . . . . . . . . . . . . . . . 16

1.5 Problem formulation and complexity . . . . . . . . . . . . . . . . . . 19

1.5.1 Formulation . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

1.5.2 Encoding delay . . . . . . . . . . . . . . . . . . . . . . . . . 20

1.5.3 Data complexity . . . . . . . . . . . . . . . . . . . . . . . . 21

1.5.4 Tools . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

1.6 Overview and contribution of the thesis . . . . . . . . . . . . . . . . . 25

**2 Optimal Buffer Constrained Independent Bit Allocation 27**

2.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27

2.2 Problem definition . . . . . . . . . . . . . . . . . . . . . . . . . . . 31

2.2.1 Quantization and cost criterion . . . . . . . . . . . . . . . . . 31

2.2.2 Delay and buffer size . . . . . . . . . . . . . . . . . . . . . . 32

2.2.3 Possible system configurations . . . . . . . . . . . . . . . . . 35

2.3 Problem formulation: integer programming . . . . . . . . . . . . . . . 37

2.3.1 A first formulation of the problem . . . . . . . . . . . . . . . . 37

2.3.2 Continuous vs. discrete optimization . . . . . . . . . . . . . . 38

2.3.3 Constrained vs. unconstrained optimization . . . . . . . . . . 38

2.3.4 Integer programming formulation . . . . . . . . . . . . . . . . 40

2.4 Optimal algorithm and fast approximations . . . . . . . . . . . . . . 42

2.4.1 Optimal algorithm . . . . . . . . . . . . . . . . . . . . . . . 42

2.4.1.1 Dynamic programming solution using the Viterbi algorithm . . . . . . . . . . . . . . . . . . . . . . . . 42

2.4.1.2 Buffer state clustering to reduce complexity in the VA 46
2.4.1.3 Use of VA techniques with limited memory . . . . . 47
2.4.2 Fast approximations . . . . . . . . . . . . . . . . . . . . . . . . 50
2.4.2.1 Constant slope optimization for budget–constrained allocation . . . . . . . . . . . . . . . . . . . . . . . . 51
2.4.2.2 Iterative constant slope optimization for buffer-constrained allocation . . . . . . . . . . . . . . . . . . . . . . . . 55
2.4.2.3 Fast iterative constant slope optimization based on heuristics . . . . . . . . . . . . . . . . . . . . . . . . 57
2.4.2.4 Generalized Lagrangian optimization . . . . . . . . . 59
2.4.3 Complexity . . . . . . . . . . . . . . . . . . . . . . . . . . . 61
2.4.3.1 Optimization using the VA . . . . . . . . . . . . . 61
2.4.3.2 Solution using constant slope optimization . . . . . . 62
2.4.3.3 Experimental comparison . . . . . . . . . . . . . . 63
2.5 Applications . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 64
2.5.1 Optimal buffer size . . . . . . . . . . . . . . . . . . . . . . . 65
2.5.2 Benchmarking . . . . . . . . . . . . . . . . . . . . . . . . . . 67
2.5.3 Statistical multiplexing of ATM sources . . . . . . . . . . . . . 68
2.6 Conclusions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 72

**3 Rate Constraints for Packet Video Transmission Based on Joint Source/Network Criteria 75**
3.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 75
3.2 Comparison of VBR and CBR video transmission . . . . . . . . . . . 79
3.2.1 Delay vs. Distortion trade-off . . . . . . . . . . . . . . . . . . 79
3.2.2 Comparison of CBR and constrained VBR . . . . . . . . . . . 81
3.2.3 VBR vs. CBR: Network aspects . . . . . . . . . . . . . . . . . 83
3.2.3.1 Resource allocation within the network . . . . . . . . 83
3.3 Greedy versus Non-Greedy coding . . . . . . . . . . . . . . . . . . 86
3.3.1 Coding Design . . . . . . . . . . . . . . . . . . . . . . . . . . 86
3.3.1.1 Types of scenes . . . . . . . . . . . . . . . . . . . . 87
3.3.1.2 Coding criteria: constant-Q, target-R and target-SNR 89
3.3.2 Coder-Network Interface . . . . . . . . . . . . . . . . . . . . . 93
3.3.2.1 Greedy coding and network policing . . . . . . . . . 93
3.3.2.2 Network issues . . . . . . . . . . . . . . . . . . . . 95
3.4 Single and double leaky buckets . . . . . . . . . . . . . . . . . . . 97
3.5 Conclusions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 102

**4 Optimal Bit Allocation for Dependent Quantization 105**
4.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 105
4.2 Dependent coding problem formulation . . . . . . . . . . . . . . . 107
4.2.1 Optimal independent allocation – the constant slope condition 107

4.2.2 General formulation . . . . . . . . . . . . . . . . . . . . . . . 108
4.2.3 Examples . . . . . . . . . . . . . . . . . . . . . . . . . . . . 109
4.2.4 Solution based on Lagrange multipliers . . . . . . . . . . . . . 111
4.2.5 Complexity . . . . . . . . . . . . . . . . . . . . . . . . . . . 114
4.2.6 Monotonicity . . . . . . . . . . . . . . . . . . . . . . . . . . 114
4.3 Temporal dependency: the MPEG case . . . . . . . . . . . . . . . . . 116
4.3.1 A particular case of MPEG: I-B-I . . . . . . . . . . . . . . . . 116
4.3.2 Pruning conditions implied by monotonicity . . . . . . . . . . 119
4.3.3 General MPEG bit allocation . . . . . . . . . . . . . . . . . . 122
4.3.3.1 Suboptimal heuristics . . . . . . . . . . . . . . . . . 124
4.3.3.2 Discussion . . . . . . . . . . . . . . . . . . . . . . . 125
4.4 Buffer constrained allocation for dependent coders . . . . . . . . . . . 126
4.5 Appendix: Proof of Theorem 1 . . . . . . . . . . . . . . . . . . . . . 129

**5 Modeling and Optimization of a Multiresolution Image Retrieval System 131**
5.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 131
5.2 System definition . . . . . . . . . . . . . . . . . . . . . . . . . . . . 135
5.2.1 Multiresolution browsing . . . . . . . . . . . . . . . . . . . . . 135
5.2.2 System Model . . . . . . . . . . . . . . . . . . . . . . . . . . 136
5.2.3 Probability of sufficient quality . . . . . . . . . . . . . . . . . 138
5.3 Analysis and Results . . . . . . . . . . . . . . . . . . . . . . . . . . 139
5.3.1 Dynamic Analysis . . . . . . . . . . . . . . . . . . . . . . . . 140
5.3.1.1 Separate channels and large image set . . . . . . . . 140
5.3.1.2 Separate channels and small image set: non constant t 144
5.3.1.3 Shared Resources . . . . . . . . . . . . . . . . . . . 147
5.3.2 Static Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . 148
5.3.2.1 Single subresolution case . . . . . . . . . . . . . . . 149
5.3.2.2 Multiple layers . . . . . . . . . . . . . . . . . . . . 152
5.3.3 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . 157
5.4 Conclusions and Future Work . . . . . . . . . . . . . . . . . . . . . . 159

**6 Adaptive Quantization Without Side Information [1] 161**
6.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 161
6.1.1 Related work . . . . . . . . . . . . . . . . . . . . . . . . . . . 163
6.1.1.1 Adaptive quantization . . . . . . . . . . . . . . . . . 164
6.1.1.2 Adaptive lossless compression . . . . . . . . . . . . . 165
6.2 Adaptation algorithm . . . . . . . . . . . . . . . . . . . . . . . . . . 167
6.2.1 Estimation of input distribution . . . . . . . . . . . . . . . . . 167
6.2.1.1 A simple non-iterative approach . . . . . . . . . . . . 172

[1] For publications related to this chapter see [73]

6.2.1.2 The zero frequency problem . . . . . . . . . . . . . . 173

6.2.1.3 Estimation of the dynamic range . . . . . . . . . . . 173

6.2.2 Quantizer design for the estimated distribution . . . . . . . . 176

6.2.3 Determining the speed of adaptation . . . . . . . . . . . . . 178

6.3 Convergence of the adaptive quantizer . . . . . . . . . . . . . . . . . 179

6.3.1 Asymptotic performance under fine quantization assumption . 180

6.3.2 Stationary performance . . . . . . . . . . . . . . . . . . . . . . . 181

6.3.2.1 Example 1: Closed form solution . . . . . . . . . . . 181

6.3.2.2 Example 2: solution using Lloyd-Max with known pdf 182

6.3.2.3 Discussion . . . . . . . . . . . . . . . . . . . . . . . . 184

6.4 Experimental results . . . . . . . . . . . . . . . . . . . . . . . . . . 186

6.4.1 Advantages of adaptivity . . . . . . . . . . . . . . . . . . . . . 187

6.4.2 Loss due to adaptivity . . . . . . . . . . . . . . . . . . . . . . 189

6.5 Conclusions and future work . . . . . . . . . . . . . . . . . . . . . . 190

**References** **192**

# List of Figures

1-1 Block diagram of a generic communications system. The source outputs blocks that are encoded and transmitted. Note that the diagram depicts an encoder formed by a transform (T), quantizer (Q) and entropy coder (E), as well as an output buffer. However this is not the only configuration we will consider. . . . . . . . . . 4

1-2 Operational vs. traditional rate distortion formulation. The traditional R-D formulation assumes a model for the source and is concerned with obtaining bounds on achievable performance, without providing constructive methods to approach those bounds. In an operational R-D framework the optimization has to find good actual operating points among the discrete set of those available (depicted as in individual points in the figure). . . . . . . . . . . . . . . . . 17

1-3 Lagrangian optimization. For a given real positive number $\lambda$ the operating point $x_i$ which minimizes $D(x_i) + \lambda R(x_i)$ is good in the sense of being on the convex hull of the R-D characteristic. For given $\lambda$, the point that minimizes the lagrangian cost is that point that is first "hit" by a line of slope $\lambda$ and is thus guaranteed to lie on the convex hull. Using several values of $\lambda$ enables to map the convex hull and search for points that meet a budget. . . . . . . . 24

1-4 Dynamic programming. The problem is to find the minimum cost path in a trellis where each branch has a cost and the costs are additive. If the optimal path from $A_1$ to $A_4$ passes through $A_3$ then the subpath from of this solution that goes from $A_1$ to $A_3$ is also the optimal path from $A_1$ to $A_3$. . . . . . . . . . . . . . . . . . . . . . 25

2-1 Block diagram of the encoding system. The buffer control mechanism determines the quantizer to be used for each input block. Given the set of quantizers, the system can be characterized by the rate, $r_{ij}$, and distortion, $d_{ij}$, for each block and quantizer. . . . . . 31

2-2 Individual and composite Rate-Distortion characteristics. Constrained and unconstrained optimization. (a) Each block in the sequence has a different Rate-Distortion characteristic. In this example there are three quantizers available to code each block. (b) For a given choice of quantizers for the blocks in the sequence, we can obtain R-D points to form the composite characteristic. No point in the convex hull meets the budget $R_1$. The optimal solution $R^*$ is not a convex hull solution. (c) $R^*$ is not a feasible solution with the chosen buffer size. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39

2-3 Trellis diagram to be used for the Viterbi algorithm solution. Each branch corresponds to a quantizer choice for a given block and has an associated cost, while its length along the vertical axis is proportional to the rate. For instance, quantizer 1 at stage $i$ produces a distortion $d_{i,1}$ and requires rate $r_{i,1}$. A path will correspond to a quantizer assignment to all the blocks in the sequence. . . . . . . . 44

2-4 The problem seen from the VA point of view: (a) The R-D characteristics of the blocks with available quantizers. (b) Equivalent representation. Each of the branches corresponds to the choice of a specific quantizer and has an attached cost. The length of the branch along the vertical axis is proportional to the rate. (c) All possible paths for the three blocks considered. Paths 1 and 2 cannot be used because of overflow. 1 and 3 are, respectively, the minimum and maximum distortion paths. . . . . . . . . . . . . . . . . . . . . . 45

2-5 SNR vs. cluster factor: The cluster factor indicates how many states or nodes are clustered at each trellis stage. Note that the suboptimality is negligible for cluster factors as high as 100, i.e. when the number of nodes, and thus the complexity, is reduced by a factor of 100. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48

2-6 Surviving paths using the full VA. The problem has finite memory: for a sufficient length, all surviving paths share the same initial path. $B_{max}$ is the buffer size in bits and $L$ is the number of blocks on which the VA is run. (a) $B_{max} = 400$ and $L = 100$, (b) $B_{max} = 500$ and $L = 100$. Note that as the buffer size is increased the length of the common path decreases. . . . . . . . . . . . . . . . . . . . . . . 49

2-7 SNR vs. number of blocks, $n$, used to obtain the VA solution. We divide $n$ by the normalized buffer size so that the horizontal axis represents the window of the VA expressed in "number of buffer sizes". In the experiment, a quantizer is chosen for block $i$ based on the best path from $i$ to $i+n$. To select the path, we force it to end at an arbitrary buffer position at stage $i+n$ (normally mid-buffer) thus explaining the non-monotonicity of the resulting function. . . 50

2-8 Moving Average of the Optimal Solution. The average is taken over 300 blocks. Note that the average is always close to the channel rate (100). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 51

2-9 For each block, minimizing $d_{ix(i)} + \lambda r_{ix(i)}$ for a given $\lambda$ is equivalent to finding the point in the R-D characteristic that is "hit" first by a "plane wave" of slope $\lambda$. . . . . . . . . . . . . . . . . . . . . . . . . 53

2-10 All possible "constant slope" solutions. Note that not all final states can be reached with a constant slope path. However, if a path exists, it is optimal. In this example $B_{max} = 3000$ bits. . . . . . . . . . . . 55

2-11 Differences in quantization choices between Algorithms 2.1 and 2.3. Note that the quantization obtained using iterative constant slopes and the VA is different for only a few blocks. . . . . . . . . . . . 57

2-12 An example of Algorithm 2.4. From block $i$ to block $j$ the allocation is not recomputed. Then, as the buffer state exceeds the threshold, the allocation is recomputed. . . . . . . . . . . . . . . . . . . . . . 58

2-13 SNR of suboptimal VA (top) and Algorithm 2.4 with 10% heuristic threshold. When a sufficient number of blocks is considered, the heuristic approximation comes within 0.05 dB of the distortion for the optimal solution. . . . . . . . . . . . . . . . . . . . . . . . . 59

2-14 SNR vs. normalized buffer size. The normalized buffer size ($B_{max}/r$) represents the size of the memory expressed in number of blocks coded at the nominal rate of $r$ bits. Note that for different values of $r$ the curves have a similar shape and the optimal normalized buffer size is roughly the same. . . . . . . . . . . . . . . . . . . . . . . . 65

2-15 Comparison of "fixed" activity–based allocation and optimal buffer–constrained allocation. In both figures the channel rate is $r = 64$ bits/block and the buffer size $B_{max} = 2000$ bits. $WSNR$ denotes the activity weighted SNR. (a) Buffer evolution for "fixed" control. For the chosen buffer size and nominal quantizer, there is underflow at both the beginning and the end of the frame. (b) Buffer evolution under VA allocation. . . . . . . . . . . . . . . . . . . . . . . . . . 69

2-16 Buffer evolution in the common buffer for the 4 sources when the allocation is computed using a combined or multiplexed VA (solid line) or an independent VA (dashed line). The resulting total SNR (obtained by averaging the MSE of all sources) increases as was to be expected when a combined VA is used for the allocation. The independent VA uses 1/4 of the rate and buffer size of the combined scenario independently for each source. . . . . . . . . . . . . . . . 71

2-17 Individual buffer occupancies for each source. (a) Source 1, (b) Source 2, (c) Source 3, (d) Source 4. The independent buffer evolution (dashed line) corresponds to a buffer size of 2000. The combined or multiplexed buffer evolution (solid line) is simulated based on the combined allocation and assuming that the channel rate was the same as in the independent case, i.e. 1/4 of the total rate. Observe that the most demanding sources (2,3,4) increase their effective rates in the combined case at the expense of the least demanding source (1). Overall the same total rate is used and, as seen in Figure 2-16, the overall SNR is improved when doing a combined allocation. . . 73

3-1 Three configurations for transmission of VBR coded video. Note that the control box sets a quantization parameter $Q$. (a) Typical configuration for studying the statistical behavior of video source and modeling the output bit rate. (b) Self policing for transmission over a packet network. (c) Transmission over a CBR link. . . . . . 77

3-2 Rate and distortion behavior for a sequence containing four types of scenes: Test, Normal, Easy, Difficult. Note that the scales are not important: we just try to qualitatively illustrate the typical behavior. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 87

3-3 Rate time series with constant Quantizer of 0.4. Peak/mean rate = 156118.0/76482.6 = 2.04. . . . . . . . . . . . . . . . . . . . . . . . 89

3-4 SNR time series with constant Quantizer of 0.4. Peak/mean dist = 16345.8/5545.4 = 2.95. . . . . . . . . . . . . . . . . . . . . . . . 90

3-5 Rate time series with target Rate of 157000 b/frame. Peak/mean rate = 156994.0/105110.8 = 1.49. . . . . . . . . . . . . . . . . . . 91

3-6 SNR time series with target Rate of 157000 b/frame. Peak/mean dist = 16345.8/4614.1 = 3.54. . . . . . . . . . . . . . . . . . . . 91

3-7 Rate time series with target SNR of 36.8 dB. Peak/mean rate = 156118.0/46070.8 = 3.39. . . . . . . . . . . . . . . . . . . . . . . . 96

3-8 SNR time series with target Distortion of 18000. Peak/mean dist = 17998.4/14120.2 = 1.27. . . . . . . . . . . . . . . . . . . . . . . . 96

3-9 SNR trace with (a) greedy and (b) non-greedy rate control. Note that the SNR remains the same for the most difficult scene, but does not exceed the target for easier scenes in the non-greedy case. . . . 97

3-10 Buffer occupancy trace with (a) greedy and (b) non-greedy rate control. Note substantial buffer underflow for easy scenes in non-greedy case. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 98

3-11 Worst case average rate for several window sizes. As an example: the point $(1, R \cdot N)$ indicates that the maximum bit rate that can be used by a single frame is $R \cdot N$. The range of average rates allowed under the constraint is represented the area under the curve. . . . 99

3-12 Motivation for using a double leaky bucket. The worst case short term behavior is determined by the short bucket, while the long term average is set by the long bucket. As before the range of admissible average rates is represented by the area under whichever curve is closer to the $x$-axis, for a given window size. . . . . . . . . . . . . 102

3-13 Worst case burst curve for a LB(3,60) that has been chosen for the non-greedy source of Fig. 3-10(b). The window is short ($N = 3$) and thus the leak rate has to be large enough to permit the larger frames to be sent. The drawback is that the long term average is 60kbit/frame, while the actual sequence's average was 46.3 kbit/frame. The greedy sequence of Fig. 3-10(a) would also be admissible. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 103

3-14 Worst case burst curve for a LB(60,55). The non-greedy sequence of Fig. 3-10(b) is also admissible under this LB. Note that the longer window $N_b = 60$ enforces a lower long term average. However, there is the danger that a compliant source may generate bursts of up to 3000 kbit/frame! . . . . . . . . . . . . . . . . . . . . . . . . . . . 104

3-15 Effect of combining two LB's. The resulting worst case burst curve shows both the lower long term average (which tends to 55kbit/frame) and smaller short term bursts (less than 180 kbit/frame). . . . . . 104

4-1 Operational R-D characteristics of 2 frames in a dependent coding framework, where frame 2 depends on frame 1. (a) Independent frame's R-D curve. (b) Dependent frame's R-D curves. Note how each quantizer choice for frame 1 leads to a different $R_2 - D_2$ curve. The Lagrangian costs shown are $J = D + \lambda R$ for each frame. . . . 109

4-2 Overview of typical source coding environments. . . . . . . . . . . . . 110

4-3 Example of search for the minimal $\lambda$. Choosing quantizer 2 would be optimal if only the first frame were considered. However choosing quantizer 1 for frame 1 is better overall because the gain for frame 2 compensates the suboptimality for frame 1. . . . . . . . . . . . . 113

4-4 Typical MPEG coding framework. (a) The MPEG frames: the I frames are independently coded, the P frames are predicted from previous I or P frames, and the B frames are interpolated from adjacent I and/or P frame pairs. (b) Temporal dependency in the MPEG framework. Note that the B frames are leaves in the dependency tree, since no prediction is generated using them. . . . . . . 117

4-5 The I-B-I special case of MPEG. Finding an R-D convex hull point corresponding to a $\lambda$ is equivalent to finding the smallest cost path through the trellis. Each trellis node corresponds to a quantizer choice for the I frames, monotonically ordered from finest to coarsest, and is populated with the associated Lagrangian cost ($J(I) = D(q) + \lambda R(q)$). The branches correspond to the B frame pairs, and are populated with their minimum Lagrangian costs ($J(B) = \min[D(q) + \lambda R(q)]$) for the particular I frame quantizer choices given by each branch's end nodes. For quality slope $\lambda$, the optimal total cost path is obtained with the Viterbi algorithm. The "dark line" path joins the smallest cost I frame nodes. Monotonicity implies that all dashed line paths can be pruned out. . . . . . . . . . . . . 118

4-6 Pruning conditions obtained from monotonicity. (a) $J_1(i_2) + J_2(i_2, j)$ is the minimum Lagrangian cost of all branches terminating in node $j$. Therefore (see Lemma 1), the $(i_3, j)$ branch can be pruned. (b) $J_2(i, j_1))$ is the minimum Lagrangian cost of all branches originating from node $i$. Therefore (see Lemma 2), the $(i, j_2)$ and the $(i, j_3)$ branches can be pruned. (c) Diagram used for the proof of Lemma 1. 120

4-7 General MPEG "trellis" diagram extension of Fig. 3. Here, the inclusion of the P frames prevents the decoupling of the B frame pairs, and the entire tree has to be grown. Note that each stage of the trellis is represented by "vector" branches whose dimension grows exponentially with the dependency tree depth. . . . . . . . . 122

4-8 Tree pruning using the monotonicity property (Lemmas 1,2). The numbers are the cumulative Lagrangian costs for a typical example for $\lambda = 10$. Branches pruned at each stage are shown with dashed lines. In this example, the number of R–D points generated is cut down from 363 (exhaustive) to only 36 with no loss of optimality. . 126

4-9 Tree pruning using monotonicity as well as a "greedy" heuristic for the same conditions as those of Fig. 6. The number of R-D points generated is now 24, at a slight loss of optimality (total Lagrangian cost is 77.91 versus optimal cost of 77.24). . . . . . . . . . . . . 127

5-1 Multiresolution image retrieval system: typical user interaction and corresponding system parameters. . . . . . . . . . . . . . . . . . . 136

5-2 System model for a multiresolution image retrieval system. $t$ is the probability an image is one of the targets. $P(\alpha)$ is the probability that $\alpha$ percent of the total bits provide sufficient quality. $B$ is the image size. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 137

5-3 Example of $P(\alpha)$ functions for several values of $m$. Note that the larger $m$ values are more realistic. . . . . . . . . . . . . . . . . . . . 139

5-4 Model for the system as closed queueing network. . . . . . . . . . 141

5-5 Norton equivalent network. . . . . . . . . . . . . . . . . . . . . . . . 142

5-6 Total delay as a function of $\alpha$ for two values of $t$. In all cases we have that $\alpha_{opt} = 0.3012$. The other parameters are set to $M = 10, m = 5, \mu_1 = 0.1, \mu_2 = 0.01, \lambda = 0.1$. Note that the trade-off is practically identical for both values of t. . . . . . . . . . . . . . . . . . . . . . . 144

5-7 Total delay as a function of $\alpha$ for several values of $\mu_2$. In all cases we have that $\alpha_{opt} = 0.3012$. The other parameters are set to $M = 10, m = 5, \mu_1 = 0.1, t = 0.05, \lambda = 0.1$. Note that for $\mu_2 = 0.01$ the delay due to the database access, $\mu_1 = 0.1$, is still significant so that optimizing the transmission results in modest gains. Conversely, for the other two values of $\mu_2$ transmission dominates the delay. . . . 145

5-8 Total delay as a function of $\alpha$ for several values of the number of users. In all cases we have that $\alpha_{opt} = 0.3012$. The other parameters are set to $m = 5, \mu_1 = 0.1, \mu_2 = 0.01, t = 0.05, \lambda = 0.1$. Note that, as all users share the database, increases in $M$ imply that the database delay becomes more significant and the gains obtained by optimizing the transmission are smaller. . . . . . . . . . . . . . . . . . . . . . 146

5-9 Total delay as a function of $\alpha$ for two values of the number of users when the commnunication resources are shared. In all cases we have that $\alpha_{opt} = 0.3012$. The other parameters are set to $m = 5, \mu_1 = 0.1, \mu_2 = 0.01, t = 0.05, \lambda = 0.1$. The relative gain is practically the same regardless of the number of users because the transmission delay dominates as the links are shared. . . . . . . . . . . . . . . . 149

5-10 Total delay as a function of $\alpha$ for two values of $t$ when the communication resources are shared. In all cases we have that $\alpha_{opt} = 0.3012$. The other parameters are set to $m = 5, \mu_1 = 0.1, \mu_2 = 0.01, M = 5, \lambda = 0.1$ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 150

5-11 Overall delay at the optimal combination of $\alpha$'s for different values of $m$. Note how the delay can be significantly decreased by increasing the number of resolutions, at the cost of increased complexity. . . . 157

5-12 Example of total normalized delay for the range of possible values of $\alpha$ for several values of the parameter $m$ in the expression for $P(\alpha)$. Note how choosing the right value for $\alpha$ can reduce significantly the total delay. This example models the cost as in (5.14) where we assume $n \ll N_1$ so that the delay is approximately $1 - P(\alpha)(1 - \alpha)$ 158

6-1 Adaptive quantization algorithm. (a) The adaptation algorithm can be decomposed in two parts (i) the estimation of the input distribution based on past samples and (ii) the design of the new quantizer *given* the estimated distribution. (b) In the simplest case the adaptive algorithm uses a fixed finite window to estimate the distribution. In a more general case it would be necessary to change the speed of adaptation as well, so that the window size would also change over time. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 163

6-2 Notation used in the model estimation algorithm. The $b_i$'s denote the decision levels, with $b_0$ and $b_L$ denoting the outer boundaries of the finite support approximation. The $x_i$ are the knots of the piecewise linear approximation. In this figure there are as many knots as bins but in general one can use more knots than bins. Note that we depicted an $f(x)$ which is non zero outside the range determined by $b_0, b_L$ to emphasize the fact that these two boundaries have to be estimated and that the operation introduces some error. 171

6-3 Example 1. 3-bin case and pdf $f(x) = 3/4(1 - x^2)$ for $-1 \leq x \leq 1$, $f(x) = 0$, elsewhere. We represent the unique solution for the boundary (vertical axis) that meets the centroid and nearest neighbor conditions, for each of the initial conditions on the boundary (horizontal axis). Note that regardless of the initial conditions the range of solutions achieved is very restricted and close to the optimal solution obtained on the true pdf. As we have only three bins and the pdf is symmetric around zero, the quantizer is completely determined by a single parameter, the boundary of the middle bin. 183

6-4 Example 1. Result of applying successively the algorithm to the three bin case with $f(x) = 3/4(1 - x^2)$. Note that convergence is very fast. Just two iterations are sufficient. The vertical axis represents the choice for the initial quantizer. The horizontal axis represents the number of iterations. . . . . . . . . . . . . . . . . . . . 184

6-5 SNR obtained after running the Lloyd-Max iteration on the $\hat{f}(x)$ obtained with different starting conditions. The horizontal axis represents the different choices for the initial quantizer with 1 being the uniform quantizer and 10 the Lloyd-Max quantizer obtained on $f(x)$. The top line indicates the performance of the Lloyd-Max quantizer design for the true pdf. The top graph represents the 3-bin case, the bottom one the 8-bin case. Note how the degradation due to using the approximation is smaller, as expected, in the 8-bin case. . . . . 185

6-6 SNR obtained after running successive Lloyd-Max iterations where at each stage the true pdf is used to generate the counts that will produce $\hat{f}(x)$. A 2 bit quantizer is used. The resulting quantizer is used as the starting condition of the following iteration. Note that convergence is fast and that even after only one iteration (with a uniform quantizer as the initial condition) the SNR is very close to that attained at convergence. . . . . . . . . . . . . . . . . . . . . . . 186

6-7 Loss in performance due to the adaptivity for several numbers of quantizer levels. The curve depicts the difference in SNR between the optimal Lloyd-Max quantizer and the quantizer obtained after iterating our algorithm. Note that the loss diminishes as the number of level increases. Also note that the decrease is not strictly monotonic. The error incurred between $f(x)$ and $\hat{f}(x)$ in the approximation is clearly strictly monotonic in the number of quantizers, but this may not be the case as far as the loss in performance is concerned.187

6-8 Comparison between adaptive and Lloyd-Max algorithms. (a) Mean mismatch. The Lloyd-Max quantizer is designed for a zero mean gaussian source. The adaptive algorithm maintains its performance constant. The variance of the source was 1. (b) Variance mismatch. The Lloyd-Max quantizer is designed for a variance 1 gaussian source. As the mismatch becomes significant the adaptive algorithm clearly out performs the Lloyd-Max quantizer. . . . . . . 188

6-9 Comparison of performance of Lloyd-Max and the adaptive algorithm. The SNR is the average measured over blocks of 2000 samples. (a) When a bimodal source is considered, the performance is much better than a Lloyd-Max design based on the complete sequence. The source switches between two states each producing different mean but same variance ($\sigma^2 = 1$). (b) When a stationary gaussian source ($\sigma^2 = 1$) is considered, the loss due to the adaptation is minimal. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 189

6-10 Comparison of performance of Lloyd-Max and the adaptive algorithm in the entropy constrained case. The average entropy of the quantizer is used. (a) When a bimodal source is considered, the performance is much better than a Lloyd-Max design based on the complete sequence. (b) When a stationary gaussian source ($\sigma^2 = 1$) is considered, the loss due to the adaptation is minimal. . . . . . . 190

# List of Tables

2.1 All possible configurations of encoding/decoding, their implications in terms of delay/memory constraints and some examples in the context of video coding . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36

2.2 Comparison of distortion and execution time for: (i) VA, (ii) a constant slope optimization with length 200 and, (iii) length 400, and (iv) heuristic approximation with 10% threshold. . . . . . . . . . . . . . . 63

3.1 Summary of the results of using the three approaches, constant-Q, target-R and target-SNR. Note that in all three cases the worst case frame is allocated the same rate of 156kbit. . . . . . . . . . . . . . . . 92

5.1 Optimal $\alpha$ for several expressions of $P(\alpha)$. Solutions obtained using the static analysis. . . . . . . . . . . . . . . . . . . . . . . . . . . . 151

5.2 Optimal $\alpha_1$ and $\alpha_0$ for several expressions of $P(\alpha)$. Solutions obtained using the static analysis in the two resolution case. The gain column indicates the reduction in total delay obtained when going from one to two resolutions. . . . . . . . . . . . . . . . . . . . . . . . . . . 152

6.1 Performance at different speeds of adaptation for a stationary source. Note that the adaptive algorithm was initialized with the optimal quantizer as designed by the Lloyd-Max algorithm on the actual data. The Lloyd-Max performance is 9.271 dB. . . . . . . . . . . . . . . . 191