Figure 1: Rate and distortion behavior for four types of scenes: Test, Normal, Easy, Difficult.

2.1. Types of scenes

We can classify video scenes into four important types, as sketched in Fig. 1. The first is the *test* scene. As with CBR codec design, this is a moderately difficult scene which is expected to have good quality at a specified rate. This scene identifies the acceptable target rate and SNR for the coder. The process of tuning algorithms to reduce artifacts and performing careful subjective studies on the test sequence remains unchanged for a VBR design.

The second scene class we will denote as *normal* frames. These are comparable to the test scene, or a bit less complex, and basically require the full bandwidth resource. They result in a good quality level, i.e. a quality completely acceptable to the viewer for long periods.

The third type are the *easy* scenes, which are substantially simpler than the test sequence. For these scenes, there is an important difference between the CBR and VBR designs. A greedy CBR approach will use the available peak bandwidth and yield an SNR which is much higher than the target established by the test scenes. The optimization criterion is usually taken as the expected signal to noise ratio, E(SNR), which will indicate quality improvement even after the distortion drops below the level where the viewer becomes insensitive (or indifferent) to further picture refinement. Clearly, a greedy allocation of bandwidth, keeps the rate consistently very close to the peak, and allows no statistical multiplexing gain (SMG). A traditional approach using E(SNR) as a performance measure will lead to the conclusion that VBR has no advantage over CBR coding. An incorrect but common belief among codec designers, that excess bits can always be put to good use, is probably due to using only short, difficult test scenes, where resources are always scarce.

The fourth type are the *difficult* scenes. These should be very rare, because they are more difficult than the test scene and result in a noticeable degradation at the allocated rate. The techniques of buffer control, bit allocation etc., devised to minimize the perceived distortion under the rate constraint are as necessary for VBR coding as for CBR. It may be possible to exceed the allocated rate momentarily through a policing mechanism such as the leaky bucket. However this is completely equivalent to a larger smoothing buffer, and the known techniques still apply. The distinction between coding for a target SNR rather than an absolutely maximized average SNR does come into play for the scenes where frames of different complexity are mixed, and the coder should avoid increasing the SNR for the easier frames beyond the target threshold.

2.2. Coding criteria: constant-Q, target-R and target-SNR

To illustrate and further understand the idea of coding for a *target* SNR, we coded a five-minute sequence (7200 frames) from the movie “Star Wars” using monochrome JPEG coding [1] with 6 different quantization scales (0.1, 0.4, 0.8, 1.2, 1.6, 2.5). From the time series of rate, $R(t)$, and quality, SNR(t) we obtain valuable information about the four scene types described above, including relative frequency, time correlation structure etc.

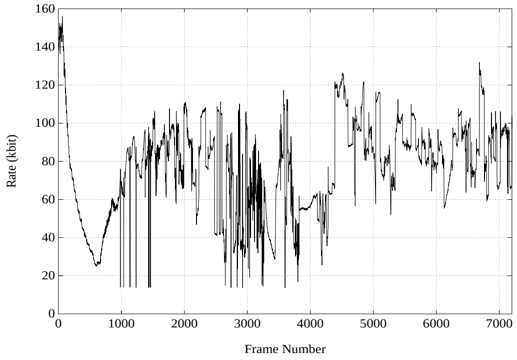

Figure 2: Rate time series with constant Quantizer of 0.4. Peak/mean rate = 156118.0/76482.6 = 2.04.

Figures 2 - 7 show $R(t)$ and SNR(t) for three rules governing the choice of quantizer for each frame. The first system (*constant-Q*, Figs. 2 and 3) has a constantly fixed quantizer level (0.4). This has often been cited as an easy way to generate VBR video, and is sometimes mistaken to be *constant quality.* As can be seen, over a long scene quality is not constant. The second case (*target-R*, Figs. 4 and 5) has a target rate, which makes it essentially like a simple CBR coding where there is no buffering between frames. We use the finest quantizer for which $R \leq R_{Target}$. The final case (*target-SNR*, Figs. 6 and 7) has a target SNR, which yields consistent quality as closely as possible given the available quantizers. For each frame, we use the coarsest quantizer for which $SNR \geq SNR_{Target}$.

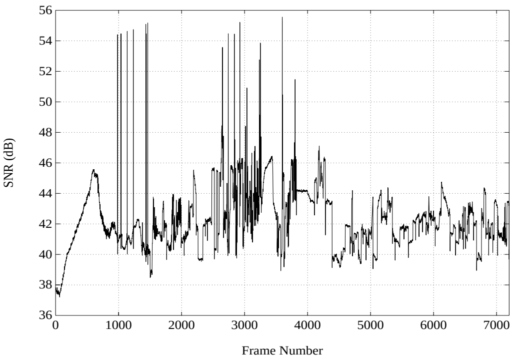

Figure 3: SNR time series with constant Quantizer of 0.4. Peak/mean dist = 16345.8/5545.4 = 2.95.

The first sixty frames include a sequence of text near the beginning of the movie, and represents the worst case of the 5-minute series. We use this as the *test scene*, to determine the tradeoff between R and SNR. The most interesting and striking feature of these three schemes is that the (worst case) rates and distortions for this test scene are practically identical. In this sense they are equivalent in quality.

The constant-Q system makes a good reference, since it indicates the natural frame complexity. Observing the target-R system in comparison, it is clear that many *easy* frames have their rate boosted to the allocated (i.e. CBR) level and their distortion is lowered far below the required level of the test scene. The target-SNR system, in contrast, keeps the distortion very close to a constant level, and its rate is somewhat more bursty than the constant-Q system, the peak is the same and the mean is lower.

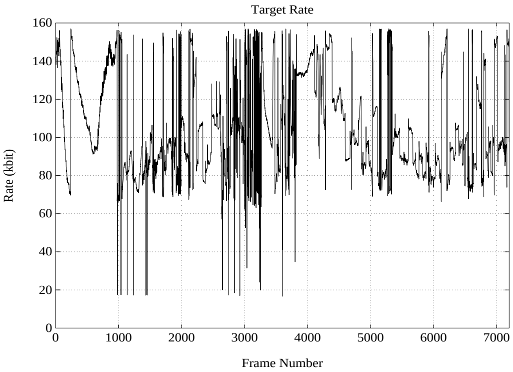

Figure 4: Rate time series with target Rate of 157000 b/frm. Peak/mean rate = 156994.0/105110.8 = 1.49.

Since the target-R system peak to mean ratio is 1.5, we conclude that 33% of the bandwidth used by a CBR packet video service can be made available by simply allowing the smoothing buffer to underflow. (Timing recovery - which is the only benefit of padding to avoid underflow - can be done explicitly through side information. This is necessary in the presence of cell loss anyway.) Another 38% can be recovered by choosing

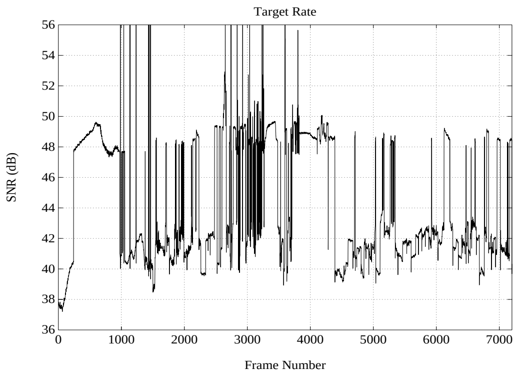

Figure 5: SNR time series with target Rate of 157000 b/frm. Peak/mean dist = 16345.8/4614.1 = 3.54.

quantizers by a target-SNR rule instead of a target-R rule. The target-SNR trace shows that only the remaining 29% of the bandwidth is utilized for necessary video information. The network bandwidth allocation (e.g. the leaky bucket rate parameter), though, still has to be set at the peak rate for the test scene. The "recovered" bandwidth is only available through statistical multiplexing in the parts of the network where several or many sources share a pipe.

In this movie, there are three or four scenes more difficult than our test scene (i.e. they require higher rate or result in higher distortion). The algorithms used to optimize the coding of such scenes under tight resource constraints remain the same for VBR as for CBR codec design. A large smoothing buffer (beyond one frame) will not improve the examples shown because the peak allocated rate is always sufficient (this is true for CBR coding as well). Where it is not, however, we can use the buffer to re-allocate rate across the several buffered frames. If an *easy* frame occurs within a *difficult* scene, we should still not optimize it below the target quality threshold. This will yield more bits to use on the critical frames.

We treat only intra-frame coding here because we can conveniently make a frame-wise choice of $R(t)$ and SNR(t) from the six choices of Q. For mixed inter/intra-frame coding (e.g. MPEG), the smoothing buffer averages bandwidth across frames coded with two or three different algorithms. To make a fair evaluation of rate and quality for a non-greedy algorithm, we would have to explicitly take into account the rate allocation algorithm which attempts to optimally trade off resources among the buffered frames (see next Section). (Even in our example, we ignore the possibility of changing quantization level within the frame.) The main result, that non-greedy quantization allows substantial statistical multiplexing gain while retaining allocated rate and an upper bound on distortion, are still valid.

3. CODER-NETWORK INTERFACE

In this section we examine the network resource allocation and policing function. The leaky bucket (LB) mechanism alone does nothing to promote non-greedy coding. The network can, however, provide proper mechanisms and incentives to ensure good SMG, without precluding consistently good quality video.

3.1. Greedy coding and network policing

In order to allocate resources in the network, there has to be some description of the traffic generated by a source, and the performance required of the network. The leaky bucket is a reasonable mechanism for specifying such an agreement, *not* because it specifies the mean rate (e.g. for billing) and the size of substantial peaks which are somehow reliably multiplexed; but because it can be used to specify an allocated rate for the single source (which is close to the peak rate), and a bound on the jitter imposed by network queues, cross traffic etc.

Since the policing mechanism regulates the coder output by dropping the violating cells, it makes sense to incorporate this function into the coder rate control algorithm. The leaky bucket however, behaves exactly as a constant rate circuit preceded by a smoothing buffer. To illustrate, the coder can generate cells at the circuit rate (or LB leak rate). Given an empty smoothing buffer (or full token bucket) an additional number of cells equal to the buffer (or bucket) size can be generated before overflowing the buffer (or exceeding the LB contract). Thus avoidance of buffer overflow with a constant rate channel is equivalent to compliance with the leaky bucket constraint and therefore the techniques for rate allocation familiar to CBR coder designers remain applicable. (Note, however, that the LB does not delay the cells as the smoothing buffer does.)

In the previous example the test scene corresponded to the most difficult scene in the sequence. To compare greedy and non-greedy coding for scenes more difficult than the test scene (i.e. with relatively scarce resources), we choose $R = 60kbit/frame$ and target SNR$= 41dB$. The examples of figures 8 and 9 compare non-greedy and greedy buffer control strategies. The greedy buffer control is an optimal bit allocation [2] which maximizes the average SNR for the given rate (here given by the LB leak rate), and the constraint that the buffer (given by the LB bucket size) does not overflow. The non-greedy version has a simple modification, which enforces an upper bound for the SNR per frame (of about 41 dB).

For the greedy algorithm, the SNR changes depending on the scene complexity (see Fig. 8(a)) while the buffer is almost never in underflow (see Fig. 9(a)). By contrast, the non-greedy version produces much more consistent quality (see Fig. 8(b)), while using less network resources as shown by the frequent occurrence of buffer underflow (see Fig. 9(b)). For the full 5 minute segment (with buffer size of 120kbit), the mean buffer output rate is 59kbit/frame for the greedy optimization, and only 46kbit/frame for the non-greedy case. The SMG is necessarily less in this case than in the previous target-SNR example since we have chosen an operating point with higher target SNR and a lower rate constraint. This results in more *difficult* and less *easy* scenes. Note (see Fig. 8) that the non-greedy algorithm reduces the SNR for those scenes above the target SNR but maintains it for those scenes near or below the target SNR. In other words, the non-greedy version of the algorithm *does not affect the worst case or difficult scenes.*

3.2. Network issues

To encourage users to adopt a non-greedy coding algorithm requires a different price structure than that for fixed bandwidth circuits. The network reuses some of the (peak) bandwidth allocated to the user, so the benefit can be returned to the user in the form of a lower tariff. Just as other aspects of coding and networking are standardized, so the statistical behavior of video traffic can be agreed upon, monitored and enforced. By definition, it is not possible (as some may wish) to enforce statistical agreements instantaneously. However it is surely possible to design and operate a communications system with statistical traffic enforcement.

Statistical multiplexing only occurs where there are several sources sharing a resource. Therefore we should expect this to be reflected in traffic description; i.e. as more sources are combined, the bandwidth allocated for each source R_a, decreases with N. The function $R_a(N)$ depends on the nature of the source [3]. So to define the network policing algorithm we must have an understanding of both the coder and network traffic control tasks.

Our results have shown that a "bounded-rate", non-greedy VBR coding scheme is practically equivalent to CBR coding in the sense of having the same allocated peak rate and distortion level on the test scene. The difference between allocated and mean bandwidth may be recovered statistically by the network.

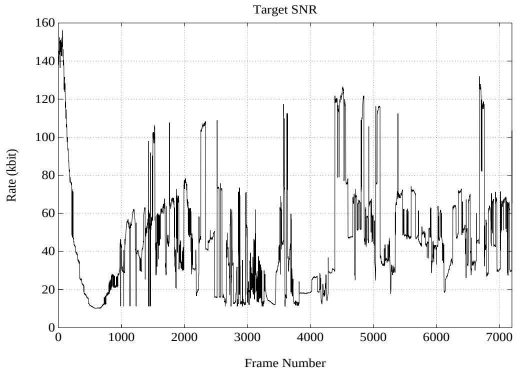

Figure 6: Rate time series with target SNR of 36.8 dB. Peak/mean rate = 156118.0/46070.8 = 3.39.

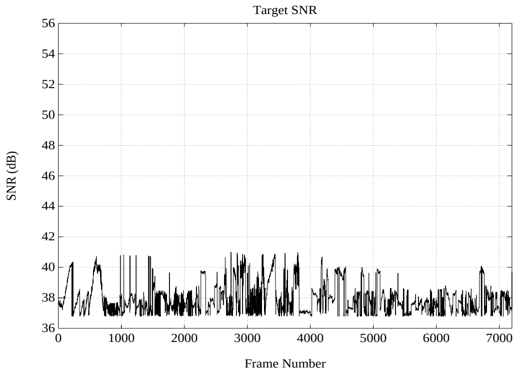

Figure 7: SNR time series with target Distortion of 18000. Peak/mean dist = 17998.4/14120.2 = 1.27.

REFERENCES

[1] "JPEG technical specification: Revision (DRAFT), joint photographic experts group, ISO/IEC JTC1/SC2/WG8, CCITT SGVIII," Aug. 1990.

[2] A. Ortega, K. Ramchandran, and M. Vetterli, "Optimal trellis-based buffered compression and fast approximations," *IEEE Trans. on Image Proc.*, 1993. To appear.

[3] M. W. Garrett, "Statistical analysis of a long trace of vbr coded video." Ph. D. Thesis Chapter IV, Columbia University, 1993.

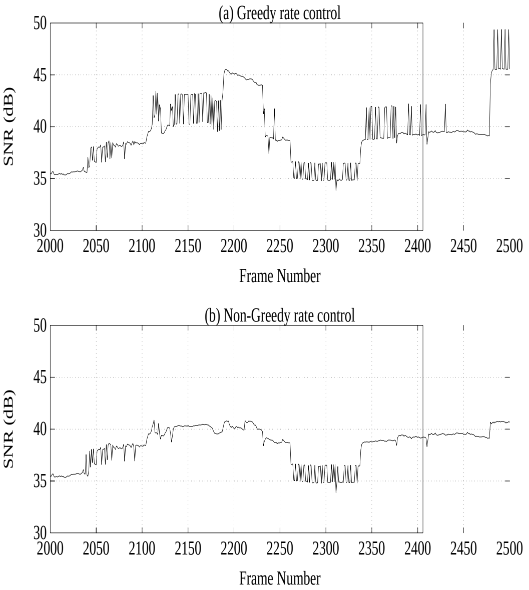

Figure 8: SNR trace with (a) greedy and (b) non-greedy rate control. Note SNR remains same for most difficult scene, but does not exceed target for easier scenes in non-greedy case.

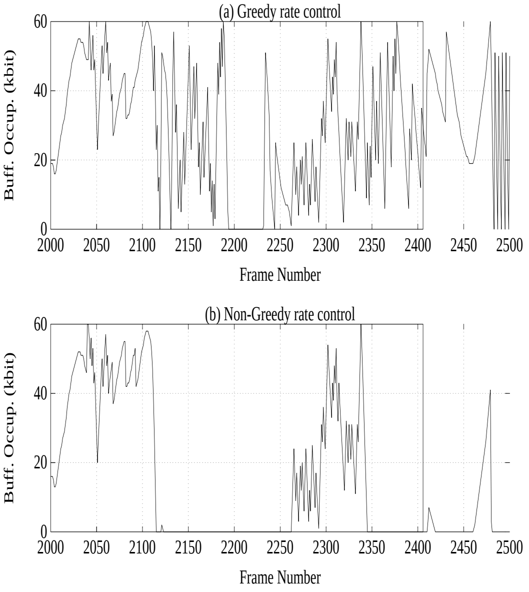

Figure 9: Buffer occupancy trace with (a) greedy and (b) non-greedy rate control. Note substantial buffer underflow for easy scenes in non-greedy case.