ANALYSIS OF UTILITY FUNCTIONS FOR VIDEO

Cheolhong An and Truong Q. Nguyen

UCSD, ECE Dept, La Jolla, California 92093-0407
chan@ucsd.edu, nguyent@ece.ucsd.edu
http://videoprocessing.ucsd.edu/

ABSTRACT

In this paper, we formulate the utility functions of distortion and Peak Signal-to-Noise Ratio (PSNR) which are generally used for the performance evaluation of video coding applications as the utility functions. The convexity property of these utility functions is analyzed in the original domain and transformed domain of an optimization variable. From this analysis, we derive joint optimization scheme with congestion control through the utility matching between TCP layer and video coding layer. Experimental results show that the overall PSNR increases and the variation of quality among the utility functions is reduced.

Index Terms— Video, distortion-rate model, utility function, convexity, joint optimization.

1. INTRODUCTION

Since the publication of the seminal paper [1] by Kelly et al. in 1998, the framework of Network Utility Maximization (NUM) has found many applications in network rate allocation algorithms, internet congestion control protocols, user behavior models and network efficiency-fairness characterization.

Consider a communication network with logical links, wired or wireless, each with a fixed capacity of c_l bps and each source transmitting at a source rate of x_s bps. Each source emits one flow using a fixed set $L(s)$ of links in its path and has a utility function $U_s(x_s)$. Each link l is shared by a set $S(l)$ of sources. In order to maximize the network utility, the problem in (1) is formulated and solved by optimization methods such as a dual-based distributed algorithm using the Lagrangian duality

$$\max_{\mathbf{x}\succeq\mathbf{0}} \sum_s U_s(x_s) \quad s.t. \sum_{s\in S(l)} x_s \le c_l, \forall l \tag{1}$$

If each user's utility function is a strictly concave function, the problem in (1) can be a convex optimization problem which has a concave object function with convex inequality constraint functions and affine equality constraint functions. A concave function $U_s(x_s)$ can be transformed into a convex function $-U_s(x_s)$ or $\frac{1}{U_s(x_s)}$ and the maximization of a concave function is equivalent to the minimization of a convex function. If we have a convex optimization problem, the solution of a convex optimization is the global optimum. Furthermore, the optimal solution can be obtained by the Lagrangian duality, if the problem in (1) is a convex optimization problem and satisfies the Slater's qualification condition which means that there is a solution vector $\mathbf{x}$ to satisfy the inequality condition strictly, that is, $\sum_{s\in S(l)} x_s < c_l$[2]. If the channel capacity c_l is larger than zero, we can always find bit rate x_s to satisfy the inequality condition strictly. Therefore, concavity of utility function is mainly discussed in this paper to use a dual-based distributed algorithm for the problem (1). Many other methods to solve optimization problems are explained in [3].

The main question arising from the expression in (1) is what the utility function of each source is. The utility functions which are considered as performance measures of an application layer are defined by the different types of resource allocation [1][4] and can be implicitly determined by given protocols such as Transport Control Protocol (TCP) [5] or Media Access Control (MAC) protocol [6] from reverse engineering. Reference [7] defines user level utility with respect to Peak Signal-to-Noise Ratio (PSNR).

In this paper, we formulate the distortion function $D(x_s)$ or $PSNR(x_s)$ [1] which are generally used for the performance evaluation of video coding applications as the utility functions since they are highly related to the objective of video coding, that is, the maximization of the video quality given bit rate constraints assuming PSNR is the objective tool measuring the video quality. Reference [8] shows that rate-adaptive real time applications have inelastic flows. However, we assume that video utility functions are operating above their minimum bit rate to guarantee minimum quality.

The rest of the paper is organized as follows. In section 2, various Distortion-Rate (D-R) models and a PSNR-Rate (P-R) model for video coding are considered as user's utility functions and the convexity of the utility functions is analyzed on the original domain and transformed domain of an optimization variable in section 3. One example of a video utility function for the joint optimization with congestion control is shown in section 4. Section 5 concludes the paper.

2. UTILITY FUNCTIONS FOR VIDEO

From the information theory, a D-R model [2] in (2) is induced from the Independent Identically Distributed (IID) gaussian process with variance σ^2 [9]

$$D(R) = \sigma^2 2^{-2R} \tag{2}$$

According to different distributions and quantization methods, the above D-R model can be generalized into (3) [9]

$$D(R) = \epsilon^2\sigma^2 2^{-2R} = \beta e^{-\alpha R} \ (\beta, \alpha > 0) \tag{3}$$

It is generally well known that a D-R model (3) only matches well with experimental results in a high bit rate region. A P-R function

This work is supported by CWC and matching fund from UC Discovery program

[1] $PSNR(x_s) = 10log_{10}\frac{255^2}{D(x_s)}$

[2] we use the bit per pixel R instead of the bit per second x_s without index s of each source for the simplicity : $R_s = \frac{x_s}{f_r \times f_w \times f_h}$ where f_r is the number of frames per second and $f_w \times f_h$ is the number of pixels per frame.

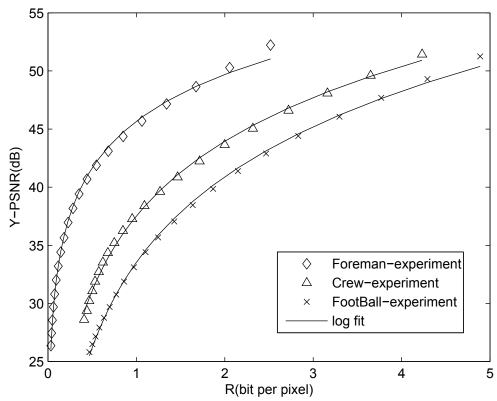


Fig. 1. PSNR vs. Rate for several videos.

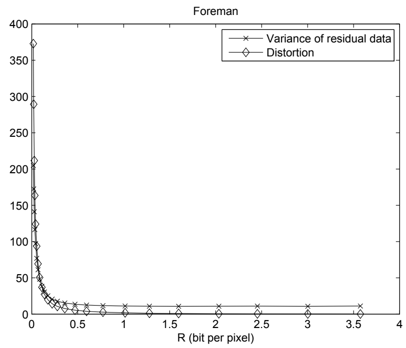


Fig. 2. Variance and distortion vs. Rate.

$PSNR(R)$ from (3) makes it clear, since $PSNR(R)$ has a linear relation with R as follows :

$$PSNR(R) = 10log_{10}\frac{255^2}{\beta e^{-\alpha R}} = a_1 R + a_2 \tag{4}$$

Figure 1 shows that the linear model (4) does not match well with the experimental $PSNR(R)$ which is highly nonlinear especially in a low bit rate region. Moreover, the video quality of many applications is between 28 and 45dB which is a highly nonlinear area. A D-R model (5) is a variation of (3) shown in [10]

$$D(R) = \beta e^{-\alpha R^{\gamma}} \ \ (\beta > 0, \ \ 0 < \gamma, \alpha < 1) \tag{5}$$

The main reason of the mismatch between mathematical models and experimental results is that the distortion of mathematical models is obtained at given variance of the Discrete Cosine Transform (DCT) coefficients. In the image compression techniques such as JPEG and JPEG2000 [9], input data of a quantizer are the DCT coefficients of natural image pixels. Therefore, the variance of input data does not depend on the quantization step size. However, in the video coding techniques such as H.264 [11], residual data of a current frame which are the difference between original pixels and predicted pixels generated from inter or intra prediction are transformed and quantized. Inter or intra predicted pixels are the sum of predicted pixels and quantized residual data of a previous frame or a current frame, respectively [11]. Therefore, residual data have different variance according to the quantization step size which controls bit per pixel R shown in Figure 2. Reference [12] shows that the variance of residual data is highly correlated to the variance of the DCT coefficients. Consequently, the variance of residual data relates to the distortion shown in Figure 2. In a high bit rate region, the variance of residual is almost same so that experimental results match well with (3) but in a low rate region the variance changes rapidly so that mathematical models are different from experimental results. Therefore, the input variance of a quantizer changes with respect to R so that a D-R model (3) needs to be modified as follows :

$$\begin{aligned} D(R) &= \epsilon^2\sigma^2(R)e^{-\alpha R} = \epsilon^2(a_1 e^{-a_2 R} + a_3)e^{-\alpha R} \\ &= ae^{-bR} + ce^{-dR} \ \ (a, b, c, d > 0) \end{aligned} \tag{6}$$

PSNR can be considered as a utility function in addition to the distortion. It is different from [7] which defines a utility function of quality with respect to PSNR. Here, PSNR itself is a utility function with respect to bit rate x_s

$$PSNR(R_s) = m_s log(R_s) + k_s \ \ (m_s, k_s > 0) \tag{7}$$

A P-R model (7) is fitted to experimental results (H.264 reference software model JM11.0 [13] is used for this experiment) shown in Figure 1. They match well with experimental results in the usual operating bit rate $(R < 2)$. A D-R model (8) is induced from (7)

$$D_s(R_s) = h_s R_s^{-j_s} \ \ (h_s, j_s > 0) \tag{8}$$

3. CONVEXITY OF UTILITY FUNCTIONS

In order to formulate the problem in (1) as a convex optimization problem, the object function must be a concave function. Therefore, we are focusing on the concavity of each utility function, because if each utility function is a concave function, the sum of the utility functions is a concave function [2]. Furthermore, the problem (1) can be decomposed by the partial Lagrangian. Then the maximization of the problem (9) is decomposed into each source's maximization problem. Interested readers can refer to [14] for more detailed explanation.

$$Q(\lambda) = \max_{\mathbf{x}\succeq\mathbf{0}} \sum_s \left(U_s(x_s) - x_s \sum_l \lambda_l \right) + \sum_l \lambda_l c_l \tag{9}$$

$$\min_{\lambda\succeq\mathbf{0}} \ Q(\lambda) \tag{10}$$

D-R models in eqs. (3, 5, 6 and 8) are strictly convex functions and a P-R model (7) is a strictly concave function if the constants of the equations satisfy the conditions. Convexity or concavity of a function can be verified by checking whether its Hessian is positive definite or negative definite, respectively [2]. If (7) or negative of eqs. (3, 5, 6 and 8) are used for the utility functions, the problem (1) is a convex optimization problem.

Moreover, references [14][15][16][17] show that optimization problems with coupled constraints can be decoupled after transformation of optimization variables and concavity of a utility function needs to be reevaluated since convexity and concavity are not an intrinsic feature of a function. One of simple examples is the geometric programming which is not a convex optimization problem but it is transformed into a convex problem [2]. Similarly, convex functions can become non-convex functions after transformation. This section analyzes the effect of transformation to the utility functions which are represented in section 2. A basic transformation which is used in

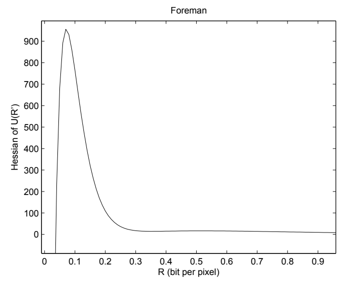


Fig. 3. Hessian of $U(R')$ for (6).

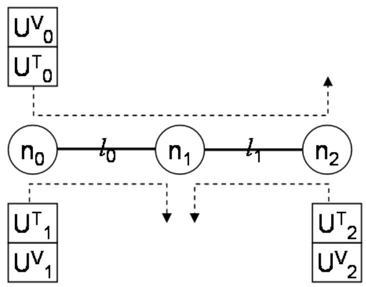


Fig. 4. Network configuration for experiment.

[14][15][16] is $R' = logR$. After transformation, the Hessian of the utility function is

$$\frac{d^2U(R')}{dR'^2} = \frac{d^2U(R)}{dR^2}\left(\frac{dR}{dR'}\right)^2 + \frac{dU(R)}{dR}\frac{d^2R}{dR'^2} \tag{11}$$

$$= R\left(\frac{d^2U(R)}{dR^2}R + \frac{dU(R)}{dR}\right) \tag{12}$$

$U(R')$ of (3) is a strictly convex function if $R > \frac{1}{\alpha}$. If α is $2log2$, R should be larger than $\frac{1}{2log2} \approx 0.72$ bit per pixel. $U(R')$ of (5) is also a strictly convex function if $R > (\frac{1}{\alpha})^{\frac{1}{\gamma}} > \frac{1}{\alpha}$. The utility functions of (3) and (5) have inelastic flows in a low bit rate region and they can be classified as a sigmoidal utility function. Consequently, they need to operate in a high bit rate region to be convex. However, as mentioned in section 2, these utility functions are not exactly modeled in a low bit rate region. Therefore, we are focusing on the other models. The utility function of (6) has a saddle point between $\frac{1}{b}$ and $\frac{1}{d}$ assuming $b > d$. From Figure 3, a saddle point of (6) exists near 0.05 bit per pixel for the foreman sequence. Therefore, $U(R')$ of (6) can be considered as a convex function in a general operating region. If $R > 0$, transformed utility functions of (7) and (8) are strictly a concave and convex function, respectively.

4. JOINT OPTIMIZATION WITH CONGESTION CONTROL

Many examples of joint optimization show TCP layer's utility as user utility [14]. In this section, the relation between TCP layer's utility and a video coding utility, especially, a PSNR utility function (7) is considered. User Datagram Protocol (UDP) is usually used for the transmission of video data but TCP can be used (the industrial home networking standard developed by Digital Living Network Alliance (DLNA) uses the HyperText Transfer Protocol (HTTP) over TCP as mandatory protocols and Real-Time transport Protocol (RTP) over UDP as optional protocols for media transport [18]). Reference [14] shows that the utility function of TCP Vegas, FAST and Scalable TCP is $\alpha_s d_s log x_s$. α_s is an unknown variable which is related to control the TCP window size shown in (14) (In case of proportionally fair resource allocation, $\alpha_s = \frac{\alpha(const)}{d_s}$ [4]) and d_s denotes the round-trip propagation delay which is usually estimated minimum Round-Trip Time (RTT) [19][14].

Table 1. PSNR weight and TCP parameter

Sequence	**Weight** (m_s)	**TCP parameter** (α_s)
Foreman	5.8	0.29
Crew	9.28	0.98
Football	10.62	1.062
Bus	11.05	1.105

From this result, the maximization problem of the sum of TCP utility functions is equivalent to the maximization of sum of PSNR utility functions if (13) is satisfied, i.e.,

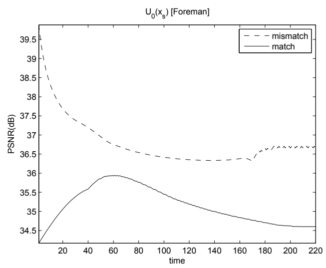
$$m_s = \alpha_s d_s, \;\; x_s^V = x_s^T \tag{13}$$

where x_s^V and x_s^T are the bit rate of a video and TCP utility function, respectively. The first condition of (13) is satisfied, if a TCP parameter α_s is adjusted according to the weight m_s which is related to video sequences listed in Table 1. These weights are obtained from curve fitting of (7) to the simulation results of H.264 reference software model JM11.0 [13]. Usually, video sequences with higher motion have a higher weight. For the joint optimization, α_s is locally updated without any interaction with the other video utility functions and their TCP protocols through the utility matching between a video encoder (e.g. H.264 encoder) and TCP shown in (13). The second condition of (13) is generally satisfied since a video encoder controls its bit rate to prevent a buffer between a video encoder and TCP overflow or underflow which means that input bit rate of a buffer x_s^V is almost same as output bit rate x_s^T.

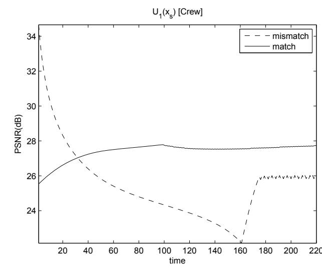
We show some experimental results regarding the utility matching at the network configuration of Figure 4. There are three video source utility functions U_k^V, TCP utility functions U_k^T and node n_k $k \in \{0, 1, 2\}$, respectively. U_0^V, U_1^V and U_2^V send the foreman, crew and football sequence which have a different weight m_s listed in Table 1, respectively. Link l_0 and l_1 have 0.5 bit per pixel and 1 bit per pixel capacity and the round-trip propagation delay time of each link d_s is 10 second. TCP Vegas algorithm is used for congestion control [19]. The effect of the delay of video utility functions is ignored assuming network layers guarantee the maximum jitter and delay of networks using MAC, Quality of Services (QoS) and buffering. Video utility functions control exactly their bit rate according to their TCP bit rate and overheads of other protocols between two utility functions are ignored which means $x_s^V \approx x_s^T$.

Two different scenarios are tested. First, video and TCP utility functions do not match. In this case, TCP assumes that its utility function $\alpha log x_s^T$ (without loss of generality, we assume $\alpha = 1$) which does not match a video utility function (7). In the other case, TCP utility functions are considered as $m_s log x_s^T$ which match video utility functions (a constant k_s of (7) does not affect to the optimal solution). In this scenario, video utility functions need to signal their weight m_s to their TCP and each TCP updates α_s listed in Table 1. In both cases, video utility functions do not solve any optimization problem but control their source rate x_s^V to follow TCP bit rate x_s^T. TCP adjusts its window size $w_s(t)$ at time t according to :

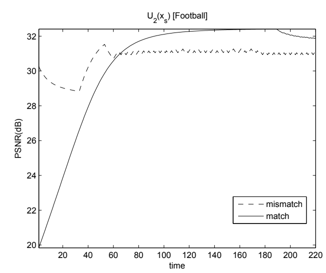
$$w_s(t+1) = \begin{cases} w_s(t) + \frac{1}{T_s(t)}, & \text{if } \frac{w_s(t)}{d_s} - \frac{w_s(t)}{T_s(t)} < \alpha_s \\ w_s(t) - \frac{1}{T_s(t)}, & \text{if } \frac{w_s(t)}{d_s} - \frac{w_s(t)}{T_s(t)} > \alpha_s \\ w_s(t) & \text{else} \end{cases} \tag{14}$$



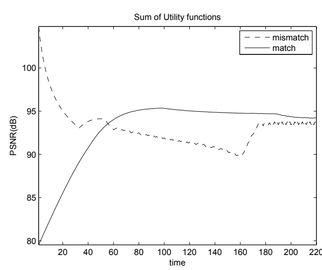
(a) PSNR of U_0^V



(b) PSNR of U_1^V



(c) PSNR of U_2^V



(d) PSNR of sum of utilities

Fig. 5. Experimental results for the match and mismatch utility functions.

where $T_s(t) = d_s + \sum_l \lambda_l(t)$ is the round-trip time of source s at time t and then TCP bit rate $x_s^T(t) = \frac{w_s(t)}{T_s(t)}$. These operations of TCP correspond to solving the primal problem (9). A link algorithm (e.g. DropTail) which solves the dual problem (10) feedbacks a dual variable $\lambda_l(t)$ to sources which use its link. Interested readers refer to [19] for more detail explanation.

Figure 5 shows PSNR of video utility functions for the match and mismatch cases. In both cases, TCP and link algorithms always try to achieve an optimal solution for TCP's utility function but in the mismatching case, the optimal solution is not optimum to the video utility function. Therefore, sum of matched utility functions is always larger than sum of mismatched utility functions at the equilibrium ($time > 200$) due to the global optimum of the convex optimization problem (1) represented in Figure 5(d).

Another important point is that the variation of PSNR among the utility functions can be reduced since the optimal bit rate $x_s^{V*} \approx x_s^{T*} = \frac{m_s}{\sum_l \lambda_l^*}$ at given optimal dual variables λ_l^* from the Karush-Kuhn-Tucker condition for the primal problem (9) [19]. Thus, the utility functions which have a larger weight m_s increase their coding rate given dual variables. Table 1 shows that video sequences with higher motion requiring higher bit rate to get a same quality usually have a larger weight m_s. Therefore, quality variation among the sequences can be reduced because a higher bit rate is allocated to a utility function encoding higher motion video sequences by the weighted proportionally fair scheme which is realized by TCP and a link algorithm. Figure 5 shows that U_0^V which has a smaller weight m_s has a lower PSNR at the match case but the other utility functions which have a larger weight have a higher PSNR at the equilibrium ($time > 200$). Consequently, the variation of PSNR among the utility functions which send the different sequences is smaller among the matched utility functions than among the mismatched utility functions.

5. CONCLUSION

In this paper, we have shown that distortion and PSNR utility functions are convex and concave in original and transformed domain, respectively. Especially, the $PSNR(R)$ function can be represented as a weighted log utility function which has different weights according to video sequences. Because some TCP congestion control algorithms is a log utility function, video and TCP can be jointly optimized to maximize the overall PSNR through the utility matching. Consequently, video streaming application can cooperate well with current TCP congestion control algorithm of the weighted proportionally fairness.

6. REFERENCES

[1] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *J. Oper. Res. Soc.*, vol. 49, pp. 273–252, Mar. 1998.

[2] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[3] D. Palomar and M. Chiang, "Alternative distributed algorithms for network utility maximization: Framework and applications," *IEEE Transactions on Automatic Control, in press*.

[4] R. Srikant, *The Mathematics of Internet Congestion Control*, Birkhauser, 2004.

[5] S. H. Low, "A duality model of tcp and queue management algorithms," *IEEE/ACM Trans. on Networking*, vol. 11, pp. 525–536, May 2003.

[6] A. Tang, J. W. Lee, J. Huang, M. Chiang, and A.R. Calderbank, "Reverse engineering mac," *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 4th I. S.*, Apr. 2006.

[7] C. E. Luna, L. P. Kondi, and A. K. Katsaggelos, "Maximizing user utility in video streaming applications," *IEEE Trans. on Circuits and Syst. Video Techn.*, vol. 13, pp. 141–148, Feb. 2003.

[8] S. Shenker, "Fundamental design issues for the future internet," *IEEE Journal of Selected Areas in Comm.*, vol. 13, pp. 1176–1188, Sept. 1995.

[9] D. S. Taubman and M. W. Marcellin, *JPEG2000 : Image Compression Fundamentals, Standards and Practice*, Kluwer Academic Publishers, 2002.

[10] Q. Wu, S.-C. Chan, and H.-Y. Shum, "A convex optimization-based frame-level rate control algorithm for motion compensated hybrid dct/dpcm video coding," *Proc. IEEE ICIP*, 2006.

[11] ITU-T Recommendation H.264 & ISO/IEC 14496-10 AVC, *Advanced Video Coding for Generic Audiovisual Services*, 2003.

[12] A. k. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall International Editions, 1989.

[13] *H.264/AVC reference software (JM11.0), http://iphome.hhi.de/suehring/tml/download/*.

[14] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition: A mathematical theory of network architectures," *Proceedings of IEEE, in press*.

[15] J. W. Lee, M. Chiang, and R. A. Calderbank, "Price-based distributed algorithm for optimal rate-reliability tradeoff in network utility maximization," *IEEE Journal of Selected Areas in Comm.*, vol. 24, pp. 962–976, May 2006.

[16] M. Chiang J.-W. Lee and R. Calderbank, "Utility-optimal random-access protocol," *IEEE Transactions on Wireless Communications, in press*.

[17] C. W. Tan, D. P. Palomar, and M. Chiang, "Distributed optimization of coupled systems with applications to network utility maximization," *Proc. IEEE ICASSP*, vol. 5, pp. 981–984, May 2006.

[18] DLNA, *DLNA Overview and vison whitepaper, http://www.dlna.org/en/industry/about/dlna_white_paper_2006.pdf*, 2006.

[19] S. H. Low, L. Peterson, and L. Wang, "Understanding vegas : A duality model," *Journal of the ACM*, vol. 49, pp. 207–235, Mar. 2002.