Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Video classification for video quality prediction

LIU Yu-xin[1], KURCEREN Ragip[2], BUDHIA Udit[3]

([1]*Sun Labs, Sun Microsystems Inc., Menlo Park, California 94025, USA*)

([2]*Multimedia Technologies Laboratory, Nokia Research Center, Nokia Inc., Irving, Texas 75039, USA*)

([3]*Mobilygen Corporation, Santa Clara, California 95054, USA*)

E-mail: zoeliu@ieee.org; ragip.kurceren@ieee.org; udit@mobilygen.com

Received Dec. 2, 2005; revision accepted Mar. 3, 2006

Abstract: In this paper we propose a novel method for video quality prediction using video classification. In essence, our approach can serve two goals: (1) To measure the video quality of compressed video sequences without referencing to the original uncompressed videos, i.e., to realize No-Reference (NR) video quality evaluation; (2) To predict quality scores for uncompressed video sequences at various bitrates without actually encoding them. The use of our approach can help realize video streaming with ideal Quality of Service (QoS). Our approach is a low complexity solution, which is specially suitable for application to mobile video streaming where the resources at the handsets are scarce.

Key words: Video classification, Video quality, No-Reference (NR), Quality of Service (QoS), Video streaming
doi:10.1631/jzus.2006.A0919 **Document code:** A **CLC number:** TN919.8

INTRODUCTION

Video content will be the main contributor to the future traffic in multimedia applications. Camcorders, digital cameras and lately mobile phones with video capturing capabilities have resulted in a wide spreading of multimedia to the masses. Quality of the captured/rendered videos is likely to be the major determining factor in the success of the new multimedia applications as well as product differentiation.

Video processing chain includes video capturing, video pre-processing, video encoding, video transmission, video decoding, video post processing, and video display. Video quality evaluation can be greatly beneficial to the design of each of the components in the creation-to-consumption processing chain, establish a benchmark, and hence improve the end user experience (Video Quality Experts Group, http://www.vqeg.org; Wang *et al*., 2003). Video quality is affected by multiple stages of processing, amongst which video compression is one of the major determining factors. Video compression techniques exploit two types of redundancies in videos, namely spatial and temporal redundancies (ITU-T H.264, 2003). There is a trade-off between the quality of the video and the amount of bits used to represent the video, i.e., the compression ratio. The higher the compression ratio is, the worse the end quality usually results.

Our paper considers the video coding artifacts and is aimed at predicting the video quality at various coding bitrates. In essence we are mainly aiming at two goals: (1) To predict video quality scores of raw video sequences compressed at different bitrates without actually encoding them. The encoding parameters such as the bitrate can be determined and tuned to achieve a target decoded video quality. How to determine the optimal coding parameters is critical to stream videos over heterogeneous networks with Quality-of-Service (QoS) (Zhang *et al*., 2005). (2) To estimate the decoded quality of compressed video bitstreams without referencing to the original uncompressed videos, which ultimately leads to the development of a No-Reference (NR) objective quality metric that may serve many applications such as monitoring the video quality in the middle of the transport network (Cheng and Lubin, 2005).

In this paper, we propose a novel approach that uses video classification for video quality prediction. We extract and examine the spatial and temporal features of an arbitrary video sequence. We create a database that includes sample video sequences with various spatial and temporal characteristics. The quality scores of these sample video sequences at different bitrates are known and used to predict the quality of a test video sequence outside the database. We implement video classification through the use of the extracted features and pattern match of a test video sequence to a known video class. We predict the quality of the test video sequence using the known quality scores of the best-matched video sequences. Our contribution is the overall architecture that uses video classification for the prediction of compressed video sequences.

PROPOSED ARCHITECTURE

The architecture of our proposed method is depicted in Fig.1.

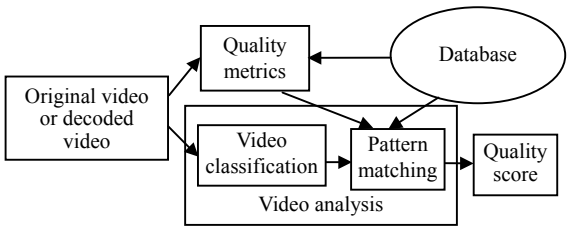


Fig.1 Proposed architecture for video quality prediction using video classification

Quality metrics: This block measures the coding distortions present in the compressed sequences. Common coding distortions such as blockiness and blurriness will be measured using metrics and will act as input to the pattern classifier.

Video classification: This block extracts features from the video sequence used for grouping the sequences into different categories based on the temporal and spatial characteristics. This block plays a key role in predicting the quality scores at different coding bitrates. The features extracted from this block are fed to the pattern classifier.

Pattern matching: The pattern classifier takes the results from the video classification block and the quality metric as inputs and uses them to match the properties of an arbitrary video sequence to the sample sequences in the database. The quality score corresponding to the best match in the database is used to measure the quality of the sequence under test.

Database: The database contains the necessary knowledge regarding the sample video sequences, including the features extracted from the video sequences, the video quality scores such as PSNR or other objective quality metrics, and achievable bitrates. The features extracted from the sample sequences in the database are used as training vectors for the pattern matching block.

VIDEO SPATIAL AND TEMPORAL FEATURE EXTRACTION

Video features that differentiate video sequences can be extracted from the source unprocessed video sequences. Video features can be roughly classified into spatial features and temporal features. Feature extraction is key in video classification. Simple and effective feature extraction algorithms are preferred.

Spatial feature extraction

Spatial feature extraction is to characterize spatial regions with different contents. A video frame may be partitioned into blocks of size $N\times N$. A commonly used spatial feature is the variance of each block averaged over the entire video sequence:

$$F_{\mathrm{S,Var}}=\frac{1}{KM}\sum_{k=1}^{K}\sum_{m=1}^{M}\left\{\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}[x_m^k(i,j)-\bar{x}_m^k]^2\right\}, \quad (1)$$

where $x_m^k(i,j)$ denotes the pixel value in location (i,j) of the mth block in the kth frame, $\bar{x}_m^k$ denotes the mean of the pixel values of the mth block in the kth frame, M denotes the number of blocks per frame, and K denotes the number of frames under investigation from the video sequence.

Also, each block can be classified into classes of flat-area, texture, fine-texture, and edge based on the Texture Masking Energy (TME) of the block, as was done in (Tan *et al*., 1996):

$$T_{\mathrm{E},m}^{k}=\left\{\sum_{i=1}^{N}\sum_{j=1}^{N}{}_{(i,j)\neq(1,1)}[\hat{H}(i,j)]^2[x_m^k(i,j)]^2\right\}^{1/2},$$

where $x_m^k(i,j)$ denotes the DCT coefficient in location (i,j) of the mth block in the kth frame. The function $\hat{H}(f)$ is the HVS relative sensitivity function with respect to the spatial frequency f, where f is related to the spatial position (i,j) as follows:

$$f(\text{cycles/degree})=\frac{\sqrt{i^2+j^2}}{2N}(\text{cycles/pixel})\times f_s(\text{pixels/degree}),$$

and we may choose f_s=32. $\hat{H}(f)$ can then be obtained as $\hat{H}(f)=|A(f)|H(f)$, where

$$H(f)=(0.31+0.69f)\exp(-0.29f),$$

$$A(f)=\left\{\frac{1}{4}+\frac{1}{\pi^2}\left[\ln\left(\frac{2\pi f}{\delta}+\sqrt{\frac{4\pi^2f^2}{\delta^2}+1}\right)\right]^2\right\}^{1/2},$$

where δ=11.636 degree^{-1}. Note that $T_{\mathrm{E},m}^{k}$ is a sum of the energy of the mth block in the kth frame weighted by the reciprocal of the square of the sensitivity function $\hat{H}(f)$. This implies that the TME provides a metric measuring the insensitivity, or equivalently, the capability of the block being resistant to noise. We may further divide each $N\times N$ block into four $(N/2)\times(N/2)$ sub-blocks, and obtain the TME for each sub-block $T_{\mathrm{E},m}^{k}(h)$, h=1, 2, 3, 4 in a similar way. Since the TME represents the insensitivities of a block and its sub-portions to noise, we use both $T_{\mathrm{E},m}^{k}$ and $T_{\mathrm{E},m}^{k}(h)$ to classify the mth $N\times N$ block in the kth frame into one of the four major categories: texture, fine-texture, edge, and flat-area.

(1) Blocks classified as flat-area have a smooth appearance or there is not much deviation in the pixel values from the mean;

(2) Blocks classified as texture contain coarser texture areas in a video frame;

(3) Blocks classified as fine-texture are coarser than the flat regions but have finer texture then the blocks classified in the texture category;

(4) Blocks classified as edges mainly contain edge pixels in a video frame.

The normalized numbers of blocks for different categories in a video frame can be averaged over different frames and features are hence obtained to characterize the spatial content in a video sequence. Let S^k(flat-area), S^k(texture), S^k(fine-texture), and S^k(edge), denote the number of blocks classified as flat-area, texture, fine-texture, and edge in the kth frame respectively, then four spatial features can be obtained as:

$$F_{\mathrm{S,flat}}=\frac{1}{K}\sum_{k=1}^{K}\frac{S^k(\text{flat-area})}{M},\tag{2}$$

$$F_{\mathrm{S,texture}}=\frac{1}{K}\sum_{k=1}^{K}\frac{S^k(\text{texture})}{M},\tag{3}$$

$$F_{\mathrm{S,fine}}=\frac{1}{K}\sum_{k=1}^{K}\frac{S^k(\text{fine-texture})}{M},\tag{4}$$

$$F_{\mathrm{S,edge}}=\frac{1}{K}\sum_{k=1}^{K}\frac{S^k(\text{edge})}{M}.\tag{5}$$

Temporal feature extraction

Temporal features can be obtained using block-based motion estimation. An example of temporal feature extraction is to characterize the motion vector associated with a macroblock of size $N\times N$ as well as the amount of prediction error. Each macroblock in a video frame can be classified into one of the following four categories: zero-motion, low-prediction-error, medium-prediction-error, and high-prediction-error.

Blocks classified as zero-motion are those with zero motion vectors and small-enough prediction errors. The prediction error, which can be either the sum of absolute difference (SAD) or the mean square error (MSE), is compared to a threshold to decide whether the prediction error is sufficiently small.

Blocks classified as low, medium, and high-prediction-error are those with non-zero valued motion vectors. The motion prediction error is compared to two thresholds, where blocks with prediction error smaller than the lower threshold are classified as low-prediction-error, blocks with prediction error falling between the two thresholds are classified as medium-prediction-error, and blocks with prediction error larger than the higher threshold are classified as high-prediction-error.

The normalized numbers of blocks for different

categories in a video frame that can be averaged over different frames and features are hence obtained to characterize the temporal content in a video sequence. Let S^k(zero), S^k(low), S^k(medium), and S^k(high) denote the number of blocks classified as zero-motion, low-prediction-error, medium-prediction-error, and high-prediction-error in the kth frame respectively, then four temporal features can be obtained as

$$F_{\mathrm{T,zero}}=\frac{1}{K}\sum_{k=1}^{K}\frac{S^k(\mathrm{zero})}{M}, \tag{6}$$

$$F_{\mathrm{T,low}}=\frac{1}{K}\sum_{k=1}^{K}\frac{S^k(\mathrm{low})}{M}, \tag{7}$$

$$F_{\mathrm{T,medium}}=\frac{1}{K}\sum_{k=1}^{K}\frac{S^k(\mathrm{medium})}{M}, \tag{8}$$

$$F_{\mathrm{T,high}}=\frac{1}{K}\sum_{k=1}^{K}\frac{S^k(\mathrm{high})}{M}. \tag{9}$$

Different features obtained from Eqs.(1)~(9) can be combined to create new features. For example, a feature can be obtained by adding $F_{\mathrm{S,texture}}$ in Eq.(3) to $F_{\mathrm{S,edge}}$ in Eq.(5), since blocks classified as texture and edge usually play a more significant role in quality judgement.

Another issue for video feature extraction worthy to mention is the complexity of the feature extraction method. A simpler method is obviously preferred. Therefore, low complexity should be retained in the method design. For example, block-based motion estimation should better be maintained only at the macroblock level and a smaller search window is preferred. Also, a video sequence usually contains hundreds of frames and a portion of the video sequence may be used for feature extraction, instead of evaluating the entire sequence. It is better to go through the entire sequence first and extract the key frames that include scene changes. Hence, a selection of video frames is accordingly determined for feature extraction.

VIDEO QUALITY PREDICTION USING VIDEO CLASSIFICATION

As shown in Fig.1, pattern matching is implemented after features are extracted for a test video sequence. Pattern matching is used to match the features extracted from the test sequence to the features extracted from the sample video sequences in the database. The closest matches found in the database to the test sequence are used to predict the quality of the sequence under test.

The closest match is obtained using the k-Nearest Neighbor algorithm (k-NN). The weighted Euclidean distances between the features of the test sequence and each of the sequences in the database is calculated to measure the closeness of the test sequence to the sequences in the database. The best match can be obtained as follows, where the weights can be obtained by training:

$$i\_best=\arg\min_i\{D_i\}=\arg\min_i\left(\sum_{j=1}^{n}w_j\left\|f_j^{(i)}-f_j^{(\mathrm{test})}\right\|\right). \tag{10}$$

In the above equation, n denotes the number of features selected to construct the feature space, $[f_1^{(i)}, f_2^{(i)}, ..., f_n^{(i)}]^{\mathrm{T}}$ denotes the feature vector extracted from the ith sample video sequence in the database, $[f_1^{(\mathrm{test})}, f_2^{(\mathrm{test})}, ..., f_n^{(\mathrm{test})}]^{\mathrm{T}}$ denotes the feature vector extracted from the test video sequence, D_i denotes the weighted distance between the ith sample video sequence and the test sequence in the feature space, and $[\omega_1, \omega_2, \ldots, \omega_n]^{\mathrm{T}}$ denotes the weights, where $\sum_{j=1}^{n}\omega_j=1$. It is expected that the test sequence behaves in a similar manner as the matched sequences. The Euclidean distances are arranged in an ascending order so that k closest neighbors are hence obtained. The quality score of the test sequence is predicted from the quality scores of these k sample video sequences in the database.

One issue that needs to be addressed is the number of closest neighbors used to predict the quality of the test sequence (k). We may choose a fixed number of neighbors (k) or use an adaptive scheme based on the closeness of the sequences from the database to the test sequence in terms of the Euclidean distance.

The prediction of the quality scores is obtained from the nearest neighbors identified. We take a weighted sum of the quality scores corresponding to the nearest neighbors. The weights are inversely proportional to the distances from the test sequence:

$$Q = \sum_{i=1}^{k} t_i Q_i \Big/ \sum_{i=1}^{k} t_i, \quad (11)$$

where Q is the predicted value, k is the number of closest matches identified, Q_i's are the quality scores of the sequences from the database corresponding to the closest matches identified, and t_i's are the weights, which are inversely proportional to the Euclidean distance between the test sequence and the closest sequences identified using pattern matching.

Derivation of weights for the weighted Euclidean distance

In order to facilitate the k-NN search strategy to identify the closest matches for the test sequence to the sample sequences in the database, we need to determine an optimum set of weights as in Eq.(10). We developed a method to obtain the optimum set of weights over different compression bitrates for the prediction of video quality scores as follows.

Step 1: Initial guess for different compression bitrates.

We randomly choose 75% of our video sequences as training sequences and the remaining as test. We first pursue the optimal weights for the lowest bitrate scenario, i.e. 64 kbps, with some random initial guess such that the average absolute error between the true quality scores and the predicted quality scores is minimized. For the same training set and test set we pursue the optimal weights for subsequent bitrate scenarios with the initial guess as the optimal weights from the previous bitrate scenario. A nonlinear constraint is added to guarantee that the correlation between the optimal weights and the initial guess or the optimal weights for the previous bitrate scenario is above 0.85. This may lead to sub-optimal solutions but will help obtain a better correlation.

Step 2: Refinement of the weights to the optimum.

We use a leave-one-out procedure and pursue the optimal weights for each test sequence using the results derived from the first step. We also add a constraint such that the optimal weights should have a correlation greater than 0.85 with the initial guess. This guarantees that the variance between the weights for all the sequences is small. We choose the mean of the weights for all the test sequences as the final weights.

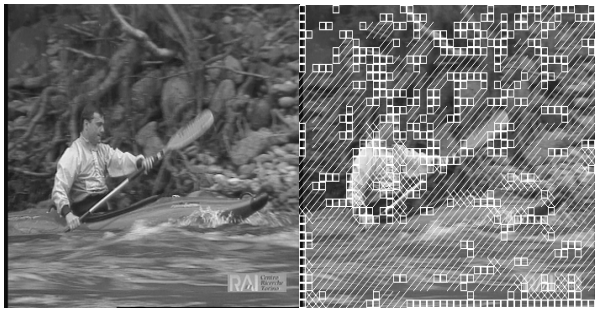
EXPERIMENTAL RESULTS

All the video sequences in our experiments are in the format of YUV 4:2:0 CIF.

Feature extraction

1. Spatial feature extraction

The spatial features that were extracted from one frame of Canoa and Football sequences are shown in Fig.2. Each 8×8 block is classified into one of the four categories: Flat, Fine-texture, Texture and Edge blocks.



(a)

(b)

Fig.2 Spatial feature extraction using block classification (Flat: no mark; Fine-texture: diagonal lines; Texture: crosses; Edges: square blocks). (a) Canoa; (b) Football

2. Temporal feature extraction

As seen from Table 1, Canoa has many blocks that result in high motion prediction errors and thus coding Canoa is relatively more difficult. In contrast, News has a larger percentage of blocks with zero motion vectors and hence is easy to code. Football is of high motion, however, the relative number of blocks with high prediction errors is smaller than that of Canoa.

Table 1 Temporal feature extraction using block classification (MV: motion vector; PE: prediction error)

Sequence	Zero *MV*	Low *PE*	Medium *PE*	High *PE*
Canoa	0	0	0.187991	0.812009
F1owergarden	0.207351	0.118967	0.538721	0.134961
Football	0.256173	0.015432	0.317621	0.410774
News	0.907127	0.004209	0.069585	0.019080

3. Feature selection

We restricted the number of features to 6 for the *k*-NN search strategy. The features that were chosen are: (1) Number of low pass or flat blocks; (2) Total number of blocks that have texture and the number of blocks that have edges; (3) Number of blocks with zero motion vectors; (4) Number of blocks with low prediction error; (5) Number of blocks with medium prediction error; and (6) Number of blocks with high prediction error.

VQM scores prediction

VQM, known as Video Quality Metrics, is an objective video quality metric developed by NTIA/ITS that has demonstrated a good correlation with the subjective quality scores in many scenarios (5,446,492, 1995; 5,596,364, 1997; 09/431,160, 1999). We obtained the VQM quality scores of various video sequences using NTIA/ITS'S VQM software at different bitrates. The scores are in the range of 0~1 where 1 represents the highest impairment and 0 indicates no impairment. The weights for different features were obtained using the algorithm described in Section 4.1, and are shown in Table 2.

Table 2 Weights at different bitrates for VQM score prediction (MV: motion vector; PE: prediction error)

Rate (kbps)	Flat regions	Zero *MV*	Low *PE*	Medium *PE*	High *PE*	Texture +Edge
64	0.0643	0.6231	0.1920	0.0635	0.0247	0.0324
128	0.1052	0.7346	0.0857	0.0197	0.0273	0.0276
256	0.0851	0.5028	0.2639	0.0632	0.0465	0.0384
512	0.0622	0.0779	0.6521	0.0780	0.0578	0.0721
768	0.0673	0.0427	0.7214	0.0777	0.0500	0.0409

It can be observed from Table 2 that temporal features "Zero *MV*" and " Low prediction error" are the most important. We then examine the reliability of our proposed approach for video quality prediction at different coding bitrates. We adopt the leave-one-out strategy in our experiments to measure the prediction error. That is, each sequence was excluded from the sample database and selected as a test sequence. The prediction of the eliminated sequence was obtained using the quality scores of the remaining sample sequences in the database. The absolute percentage error incurred in the prediction is shown in Table 3 for sequences coded at various bitrates.

We defined the prediction error as the absolute difference between the original scores obtained using the VQM software and the scores predicted using our approach. It is observed that a larger prediction error is resulted for lower bitrate scenarios as compared to higher bitrates. The prediction error significantly reduces as we move from lower bitrates to higher bitrate scenarios. Table 4 shows the correlation between the original quality scores calculated using the VQM software and the predicted scores across different bitrates. We see a higher correlation for the higher bitrate scenarios as compared to the lower bitrates. We argue that the high prediction error introduced at lower bitrates may be caused by the inconsistency in calculating the quality scores by the VQM software. We know for a fact that the sequence News is easier to code as compared to the sequence Football, owing to its low spatial and temporal activities. The quality scores as calculated by the VQM software for News and Football at 64 kbps however are 0.6619 and 0.4657 respectively, suggesting that News has worse quality as compared to Football. At higher bitrates in contrast, the scores for News are lower than that of Football, suggesting that News has a better quality. Subjective testing always implies that News demonstrates a better quality than Football at a specific coding bitrate. Hence we argue that the inconsistency of the VQM software results in a lower correlation in the lower bitrate scenarios.

Table 7 (Column 2) shows the correlation between the original and predicted scores over different bitrates across various sequences. The high correlation suggests that the predicted scores for a specific sequence follow the same trend over different bitrates.

Prediction of PSNR's

Similar to Section 5.2, we predicted the *PSNR* values for various video sequences at different bitrates and obtained the results as shown in Table 5,

Table 6, and Table 7 (Column 3). It is observed that using our approach, the predicted *PSNR* values are well correlated to the true values regardless of the variety across encoding bitrates and video content, which is different from the case for VQM scores prediction discussed in Section 5.2.

CONCLUSION

We have proposed a video quality prediction approach using video classification. We extract a variety of spatial and temporal features from an arbitrary video sequence and group different video

Table 3 Prediction error for VQM scores at different bitrates (*p*: kbps)

Sequence	*p*=64		*p*=128		*p*=256		*p*=512		*p*=768	
	Original	Predicted	Original	Predicted	Original	Predicted	Original	Predicted	Original	Predicted
Akiyo	0.4992	0.6402	0.3195	0.4048	0.1591	0.2478	0.1051	0.1519	0.0755	0.1143
Bridge-close	0.6133	0.5771	0.4676	0.5611	0.3718	0.4679	0.3718	0.2745	0.2503	0.2472
Bridge-far	0.2155	0.6438	0.2030	0.5173	0.1917	0.3832	0.1572	0.2566	0.1317	0.2265
Canoa	0.4567	0.4774	0.4521	0.4789	0.4536	0.4715	0.3900	0.3658	0.2856	0.3071
Coastguard	0.6482	0.5340	0.6536	0.4662	0.5971	0.4090	0.3555	0.3270	0.2664	0.2531
Container	0.5227	0.6490	0.3874	0.4319	0.2945	0.2744	0.1945	0.1494	0.1568	0.0998
Crew	0.6361	0.5102	0.6331	0.3818	0.5941	0.2700	0.3950	0.1656	0.2847	0.1900
F1	0.4503	0.5629	0.4477	0.5239	0.4394	0.4061	0.3423	0.3112	0.2619	0.2064
Flower	0.4106	0.5094	0.4139	0.4635	0.3863	0.4750	0.2187	0.3419	0.1586	0.2416
Football	0.4657	0.4891	0.4607	0.4968	0.4708	0.4548	0.3908	0.3810	0.2911	0.3079
Foreman	0.5364	0.5674	0.5286	0.4771	0.3464	0.4373	0.1931	0.2602	0.1375	0.2546
Hall	0.5378	0.4223	0.3507	0.4115	0.2697	0.4183	0.1822	0.2149	0.1474	0.1777
Highway	0.7396	0.4261	0.5615	0.3610	0.3765	0.2601	0.2898	0.1898	0.2390	0.1727
Husky	0.3181	0.4658	0.3258	0.4792	0.3216	0.4774	0.3073	0.4024	0.2953	0.3168
Ice	0.4497	0.6531	0.4236	0.4688	0.2269	0.2549	0.1142	0.1662	0.0749	0.1391
Irene	0.6456	0.5403	0.5113	0.4149	0.2975	0.2588	0.1841	0.1479	0.1367	0.1033
Mobile	0.4856	0.4436	0.4853	0.4644	0.4825	0.4385	0.4204	0.3834	0.3328	0.3076
Mother	0.6626	0.5729	0.3908	0.3762	0.2351	0.2254	0.1415	0.1365	0.1017	0.1069
News	0.6619	0.5721	0.4544	0.3892	0.2890	0.2801	0.1535	0.1802	0.1024	0.1234
Paris	0.6352	0.6054	0.5840	0.4733	0.4171	0.4003	0.2684	0.1736	0.1975	0.1231
Rugby	0.4305	0.5004	0.4245	0.5070	0.4351	0.4942	0.4028	0.3859	0.3298	0.2950
Silent	0.7106	0.6184	0.4904	0.5060	0.2714	0.3123	0.1711	0.2132	0.1181	0.1240
Soccer	0.5037	0.4647	0.5004	0.4597	0.4719	0.4552	0.3126	0.3204	0.2349	0.2502
Tempete	0.5828	0.4759	0.5871	0.4497	0.5252	0.4500	0.3377	0.3159	0.2518	0.2945

Table 4 Correlation coefficients for VQM prediction at different bitrates

Bitrate (kbps)	64	128	256	512	768
Correlation coefficient	−0.424	−0.1617	0.47299	0.69967	0.7723

Table 5 Weights at different bitrates for PSNR prediction

Rate (kbps)	Flat regions	Zero *MV*	Low *PE*	Medium *PE*	High *PE*	Texture +Edge
64	0.0643	0.6231	0.1920	0.0635	0.0247	0.0324
128	0.1052	0.7346	0.0857	0.0197	0.0273	0.0276
256	0.0851	0.5028	0.2639	0.0632	0.0465	0.0384
512	0.0622	0.0779	0.6521	0.0780	0.0578	0.0721
768	0.0673	0.0427	0.7214	0.0777	0.0500	0.0409

Table 6 Correlation coefficients for PSNR prediction at different bitrates

Rate (kbps)	64	128	256	512	768
Correlation	0.83	0.829	0.9022	0.911426	0.90337

Table 7 Correlation coefficients for VQM prediction and PSNR prediction across different contents

Sequence	*VQM* prediction	*PSNR* prediction
Akiyo	0.9976	0.9970
Bridge-close	0.8437	0.9650
Bridge-far	0.9371	0.9744
Canoa	0.9656	0.9213
Coastguard	0.9456	0.9503
Container	0.9982	0.9990
Crew	0.8128	0.8939
F1	0.9359	0.9878
Flower	0.9761	0.9863
Football	0.9610	0.9923
Foreman	0.9589	0.9938
Hall	0.7923	0.8844
Highway	0.9924	0.9463
Husky	0.9834	0.9296
Ice	0.9621	0.9532
Irene	0.9987	0.9948
Mobile	0.9889	0.5871
Mother	0.9978	0.9947
News	0.9983	0.9986
Paris	0.9807	0.8312
Rugby	0.9507	0.8420
Silent	0.9866	0.9745
Soccer	0.9970	0.9819
Tempete	0.9819	0.9786

sequences with regard to these features. We exploit the *k*-NN strategy and predict the quality score of a test video sequence from those of its closest neighbors in the feature space. The significance of our approach is that it can be deployed to obtain the quality score for a compressed video without the availability of the original video sequence, as long as we know the group the sequence belongs to. Also, our approach can be used for the prediction of video qualities at different bitrates without actually coding the sequences. This is specially useful for video streaming with QoS. The coding bitrate can be identified first according to the video content present in a video sequence so that the coding parameters can be tuned to achieve the ideal decoded video quality.

References

Cheng, H., Lubin, J., 2005. Reference-Free Objective Quality Metrics for MPEG Coded Video. Proceedings of SPIE International Conference on Human Vision and Electronic Imaging X, **5666**:160-167.

ITU-T H.264, 2003. Advanced Video Coding for Generic Audiovisual Services.

Patent Application No. 09/431,160, 1999. In-service Video Quality Measurement System Using an Arbitrary Bandwidth Ancillary Data Channel. Http://www.its.bldrdoc.gov/n3/video/vqmdownload_US.htm.

Patent Number 5,446,492, 1995. A Perception-based Video Quality Measurement System.

Patent Number 5,596,364, 1997. Perception-based Audio-visual Synchronization Measurement System.

Tan, S.H., Pang, K.K., Ngan, K.N., 1996. Classified perceptual coding with adaptive quantization. *IEEE Trans. on Circuits and Systems for Video Technology*, **6**(4):375-388. [doi:10.1109/76.510930]

Wang, Z., Sheikh, H.R., Bovik, A.C., 2003. Objective video quality assessment. *In*: Furht, B., Marqure, O. (Eds.), The Handbook of Video Databases: Design and Applications. CRC Press, p.1041-1078. Available from http://www.cns.nyu.edu/~zwang/files/publications.html.

Zhang, Q., Zhu, W.W., Zhang, Y.Q., 2005. End-to-end QoS for video delivery over wireless Internet. *Proceedings of the IEEE*, **93**(1):123-134. [doi:10.1109/JPROC.2004.839603]