# CONTENT-BASED UTILITY FUCNTION PREDICTION FOR REAL-TIME MPEG-4 VIDEO TRANSCODING

*Yong Wang*[*], *Jae-Gon Kim*[#], *Shih-Fu Chang*[*]

[*]Department of Electrical Engineering
Columbia University, New York, USA

[#]Electronics and Telecommunications Research Institute,
Daejeon, Korea

## ABSTRACT

Utility function based transcoding is an efficient systematic solution for choosing optimal media transcoding operation to meet dynamic resource constraints (such as bandwidth). However, to date the real-time generation of utility function is not feasible due to computational complexity. In this paper we present a content-based utility function prediction framework for real-time MPEG-4 video transcoding. We develop a statistical approach combining real-time compressed-domain feature extraction, content-based pattern classification and regression. Our extensive experiment results demonstrate that the proposed method achieves very promising prediction accuracy -- up to 89% in choosing the optimal transcoding operation with the highest quality from multiple alternatives meeting the same target bitrate.

## 1. INTRODUCTION

Media transcoding is a process adapting original media into a new version and meanwhile matching the resource (e.g., bandwidth) constraint or user's preference. Many adaptation methods exist for adjusting the bit rate of compressed video streams. For example, requantization of DCT coefficients, frame dropping (FD), DCT coefficients dropping (CD), resolution reduction are commonly used. However, most existing works concentrate on optimization of pre-selected adaptation operations, rather than systematic solutions in choosing the optimal operations. In [8,10], the rate-distortion (R-D) characteristics were used to find optimal combinations of the frame rate and spatial quality of video. However, real-time generation of the R-D information was not addressed. Some works [9] use approximate analytical models for generating R-D information. But the accuracy still requires improvement and the analytical models are codec specific.

In [1][2], we presented a new utility function based framework that extends the conventional R-D model, to support flexible considerations of diverse types of adaptations, resources and utilities. The usage of utility function enables selection of the optimal operator among multiple options meeting given resource constraints or user preferences. Though utility function provides many benefits in guiding the design of media transcoding, the generation of utility function through exhaustive simulations is, unfortunately, time consuming and thus to date not applicable in real-time applications. To solve this problem, we propose a real-time utility function prediction method that explores the strong correlations between content features and the utility function characteristics associated with a video.

One related work involving utility function prediction is [3], where subjective utility functions for MPEG-4 streams are estimated based on the visual features by using pattern classification methods. One problem of this method is its limited accuracy due to rough prediction using only the representative utility function of each class. It does not consider the variance of utility function within the same video class.

In this paper we propose a novel utility function prediction model combining real-time compressed-domain feature extraction, pattern classification, and regression. The basic idea is to consider this problem from a standpoint of statistic regression, using the automatically extracted content features as input and the utility function as the target. At the same time pattern classification is employed to enhance the regression accuracy. Our extensive MPEG-4 transcoding experiment results show a very promising accuracy (up to 89%) in choosing the optimal operation from multiple competing options. To the best of our knowledge, this is the first system that can predict the utility function in real-time and achieve a satisfactory accuracy in finding the optimal transcoding operator.

The rest of this paper is organized as follows: in section 2, we describe the framework of utility function based transcoding and the statistical approach to utility function prediction; the experiment setup and results are presented in section 3; the collusion is given in section 4.

## 2. UTILITY FUNCTION BASED TRANSCODING AND PREDICTION

### 2.1 Utility Function Based Transcoding

Utility function (UF) is an efficient tool for describing the relationship between the utility of video and the required resource. A general description of UF can be found in [4], where relationships among diverse types of resources (bandwidth, power, display etc) and utilities (objective or subjective quality, user satisfaction etc.) can be modeled. Figure 1 depicts the conceptual Adaptation-Resource-Utility spaces (left) and the UF for a special case (right) when we only consider bandwidth for resource and video objective quality for utility. Each point in the UF stands for one specific transcoding operator, which is defined by the specific transcoding methods (e.g., CD and FD) and the set of parameters in specifying each method (e.g., the level of CD and FD).

To illustrate our method without losing generality, in this paper we consider a specific case involving only two types of adaptations -- Frame Dropping (FD) and AC DCT Coefficient Dropping (CD) and their combinations. FD adapts the source stream by skipping frames, while CD transcodes the source stream by truncating the high frequency DCT coefficients. FD-CD is important in practical wireless mobile applications. The FD-CD methods have the benefits of implementation simplicity and computational efficiency. Both operations can be efficiently implemented in the compressed domain without full decoding of

the compressed streams. Specifications about FD-CD, such as algorithm, implementation and special issues can be found in [1][2].

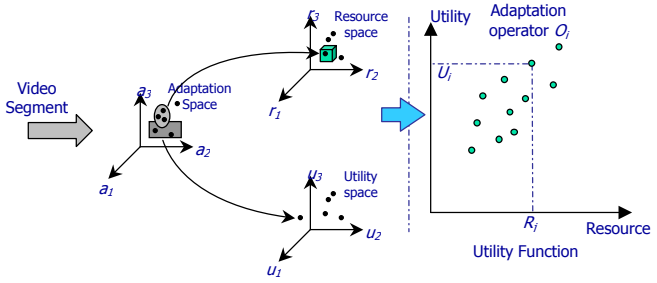



Figure 1: the adaptation-resource-utility conceptual framework and the associated utility function

A typical utility function is shown in figure 2. Given an adaptation operator $O_i$, its corresponding resource and utility value are denoted as $R_i$ and $U_i$. An operator specifies the exact amount of CD and FD combined in meeting the reduced bit rate. This ($O_i$, $R_i$, $U_i$) triplet, called anchor node, is a point in the utility function. We group points with the same FD amount (e.g., dropping one B frame between every two I or P frames in MPEG stream) to the same curve. We use linear approximation to describe other adaptation operators by connecting two adjacent anchor nodes sharing the same FD because interpolation can be done for CD but not for FD. Given a resource point $R$, all of the possible operators meeting the same resource constraint, such as $O_A$ and $O_B$ in figure 2 can be found. The research challenge and opportunity here is to automatically select the optimal operator based on some criterion (e.g., maximal utility value) from the competing operators.

We simplify the representation of the utility function by using the linearized envelope of each curve as the dash lines shown in figure 2. Thus the utility function can be denoted as

$$\vec{F}^{UF} = (f_1^{UF}, f_2^{UF}, ..., f_n^{UF}) = (r_1, r_2, ..., r_{n/2}, u_1, u_2, ..., u_{n/2})$$

where $r_i$ and $u_i$ are resource and utility value for $i^{th}$ end anchor node. Although higher-order representations can be incorporated, our experiment results discussed later verify that this linear approximation is sufficient.

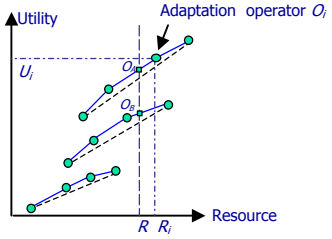



Figure 2: Example of UF considering only FD and CD.

The characteristics of UF depend on the type of video and the specific video coding techniques. Given the same video codec, there exist strong correlations between video content features and UF behaviors. For example, videos with similar visual features, e.g., motion, visual complexity, scene composition etc, tend to exhibit similar UF characteristics. Such correlations have been verified in our earlier works in classifying the UF of video segments and will be used again as the basic assumption in this paper. Note to take into account the dynamic nature of video, we decompose the video into short video segments with consistent features and apply UF prediction for each segment separately.

### 2.2 Utility Function Prediction

The issue of utility function prediction can be formalized as: given the content feature $\vec{F}^{CF}$ of one specific video clip, how can we find a suitable model mapping from the content feature space into the utility function space, so that $\vec{F}^{UF} = G(\vec{F}^{CF})$, where $\vec{F}^{UF} = (f_1^{UF}, f_2^{UF}, ..., f_n^{UF})$ is the $n$-dimension utility function vector and $f_i^{UF}$ is $i^{\text{th}}$ component, and similarly $\vec{F}^{CF} = (f_1^{CF}, f_2^{CF}, ..., f_m^{CF})$ is the $m$-dimension content feature vector and $f_j^{CF}$ is $j^{th}$ component. $G$ is the mapping we are hunting for. This is a typical multivariate regression problem. For each $f_i^{UF}$ in $\vec{F}^{UF}$, we are trying to find a mapping $g_i$, so that

$$f_i^{UF} = g_i(\vec{F}^{CF}) = g_i(f_1^{CF}, f_2^{CF}, .., f_m^{CF}) \quad (1)$$

Using Taylor expansions, we have:

$$\begin{aligned} f_i^{UF} &= g_i(\vec{F}^{CF}) = g_i(f_1^{CF}, f_2^{CF}, .., f_m^{CF}) \\ &= g_i(\vec{F}_0^{CF}) + (\vec{F}^{CF} - \vec{F}_0^{CF}) \nabla g_i(\vec{F}_0^{CF})^T + O((\vec{F}^{CF} - \vec{F}_0^{CF})^2) \end{aligned} \quad (2)$$

Ignoring high orders, this mapping can be considered as a classic linear regression problem. The first-order approximation is effective if the content feature space can be divided into some small areas centered at $\vec{F}_{0k}^{CF}$, and the UF at each center point can be accurately predicted. Suppose we form $K$ such subareas $C_k$, $k$=1,2…$K$. For each $\vec{F}^{UF} \in C_k$,

$$\begin{aligned} &\vec{F}^{UF} = (f_1^{UF}, f_2^{UF}, ..., f_n^{UF}) = (\vec{F}^{CF}\ 1) \cdot \begin{pmatrix} \vec{g}_{1k}^T & \vec{g}_{2k}^T \cdots \vec{g}_{nk}^T \\ c_{1k} & c_{2k} \cdots c_{nk} \end{pmatrix} = \vec{F} \cdot G_k \\ &c_{ik} = (g_{ik}(\vec{F}_{0k}^{CF}) - \vec{F}_{0k}^{CF} \cdot \nabla g_{ik}(\vec{F}_{0k}^{CF})^T) \\ &\vec{g}_{ik} = \nabla g_{ik}(\vec{F}_{0k}^{CF}) \quad i = 1,2,...n \end{aligned} \quad (3)$$

Thus the problem can be modeled as a $K$-segment piecewise linear regression problem. We adopt machine-learning approach to find these subsets. Firstly by unsupervised clustering, we divide the utility function space into some subspaces. Then by classification based on the content features, we can label an incoming video clip into one subspace. At last the utility function is predicted using the piecewise linear regression model.

Figure 3 shows the system architecture of our proposed framework. The upper part is the overall structure. For each video stream, the content features are extracted and the utility function prediction is applied. The adaptation engine reshapes the stream according to the predicted utility function. The lower part is a zoomed view into the utility function prediction model. It can be roughly categorized as the offline training routing and the online processing routing, while the former is in charge of the unsupervised clustering, classification learning and regression learning, and the later undergoes online classification and linear regression. For the offline training routing, firstly we build up a media pool using training video clips. The utility function and content features of each clip in the pool is calculated in advance. Then the content features are used to

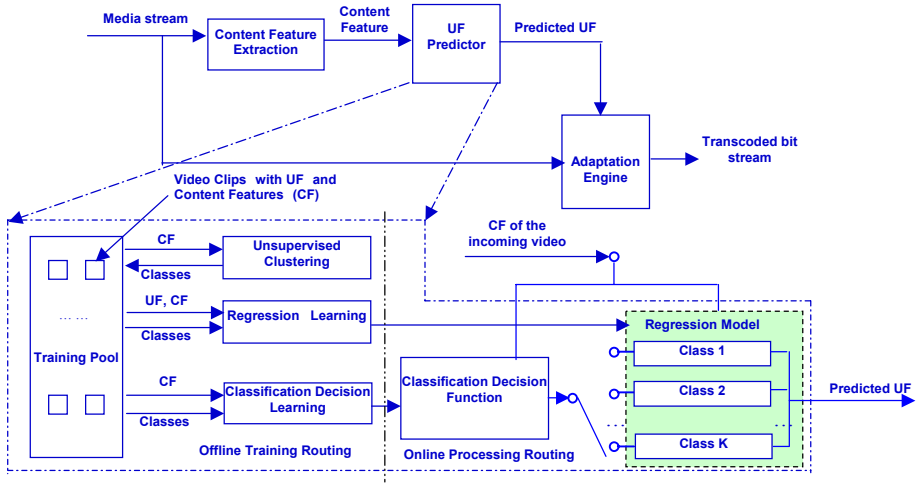



Figure 3: System architecture of the proposed framework

obtain the clustering result and each clip will be mapped to one of the clusters accordingly. Given the labeled instances in the pool, the classification decision can be trained using the content features. And further the linear regression conversion matrix can be trained using utility functions and content features. Please note for each class, there is its specific regression model. For the online processing routing, the content features of the incoming video will firstly be used by the classification function. Then according to the classification result, corresponding regression model for the selected class will be activated and the predicted utility function is obtained.

### 2.3 Content Feature Extraction

Content features are extracted based on one-second clip so that they are more or less consistent. We also make sure there is no shot transition in the clip. Based on our previous study in [3], we choose the following content features: 1) Average motion intensity; 2) Motion variance; 3) Average ratio of macroblocks with non-zero motion vector; 4) Average I frame AC DCT coefficient energy; 5) Average P frame AC DCT coefficient energy; 6) Average quantization step; 7) Average PSNR if available (this feature is available in the server where the original video is located, or in the adaptation engine where the encode log information is available). These content features characterize both the spatial texture complexity and temporal motion intensity information. They are extracted directly from the compressed format of the input stream and its metadata.

### 2.4 Algorithm Description

*2.4.1 Unsupervised Clustering*

The purpose of unsupervised clustering is to distinguish the whole content feature hyperspace into several small subspaces. We employ the K-Harmonic Mean (KHM) algorithm introduced in [5], which is a variance of the classic K-mean method with a *p*-order harmonic mean kernelled distance evaluation:

$$D(\{x_i\}_{i=1}^{N},\{c_l\}_{l=1}^{K})=\frac{1}{N}\sum_{i=1}^{N}\frac{K}{\sum_{l=1}^{K}\left(\|x_i-c_l\|^{p}\right)^{-1}} \qquad (4)$$

where $x_i$ is the feature vector and $c_l$ is the cluster center. $N$ is the number of observations and $K$ is the number of clusters. Compared with K-Mean, KHM has the advantage of initialization insensitivity. One general problem of unsupervised clustering is to decide the number of the clusters. Clustering with K too large will lose generality and result in overfitting, while too small K will lead to too much bias. In our experiment we empirically set $K$=16 which yields satisfied performance. It is an interesting research topic how to choose a suitable cluster number dynamically for different practical domains.

*2.4.2 Supervised Classification*

The purpose of supervised classification is to label the incoming video into one specific class and afterwards select corresponding regression model to predict the utility function. We employ the multiple-class DAGSVM algorithm presented in [6] with minor modification to handle the ambiguous region issue. In DAGSVM the classifier is constructed by using Decision Directed Acyclic Graph.

*2.4.3 Piecewise Linear Regression*

We use the least square error (LSE) algorithm to train the piecewise linear regression conversion matrix for each class. For an *m-dimension* content feature and *n*-dimension utility function vector, each matrix is with *(m+1)* x *n* dimension. (See Eq. (3)).

## 3. EXPERIMENT RESULTS

In our experiment, we select three movies to setup our video pool. The details of the video pool are summarized in table 1 and the algorithm in table 2. Our proposed algorithm will be tested by a formal method of cross-validation separating instances in the pool as training and testing data.

The utility function is constructed as follows: based on the MPEG GOP structure, we adopt four FD operators: no FD, one B frame dropping (in each sub GOP), two B frames dropping and keeping I frame only. For each FD, we adopt six CD levels: from 0% to 50% with 10% incremental step. Therefore there are totally 24 anchor nodes and four curves in each utility function. We implement our CD using Lagrange optimization algorithm based on the work in [7] with some modifications. Specifically,

instead of using macroblock-level truncation (i.e., each macroblock has a unique truncation point shared by all four luminance and two chrominance blocks), we use a block-level search to find individual truncation points for each block. Our experiment on standard sequences shows 0.5dB to 2dB improvement by using this finer level of optimization.

Evaluation of the proposed prediction method can be based on various performance metrics. First, errors in predicting the UF can be got based on $L$-2 distance metrics.

$$D(\{x_i\}_{i=1}^{N}) = \frac{1}{N}\sum_{i=1}^{N}\left\|F_i^{UF} - \widetilde{F}_i^{UF}\right\|^2 \qquad (5)$$

where $F_i^{UF}$ is the actual utility function and $\widetilde{F}_i^{UF}$ is the predicted one. $N$ is the size of the test set. Ideally this value is zero and we will see how close we can go toward this ideality. Figure 4(1) shows the prediction errors from three methods: our proposed method; the proposed method without regression and an alternative approach using clustering in the UF space and without regression, which is adopted in [3]. The experiment has been run for 10 times and the averages are attached afterwards. The proposed method achieves the most accurate result.

Secondly, we compare the accuracy in selecting the optimal operator given various target bit rates. Five bandwidths are selected: 1.2Mbps, 1.0Mbps, 800kbps, 480kbps and 320kbps. The original input video rate is 1.5 Mbps. Our proposed method is compared with two others: one is using clustering in the CF space; another is using a probability-based selection criterion, i.e., the most frequently used operator given a target bitrate in the training pool will be selected. This method is reasonable if we only have prior knowledge and don't want to use content features dynamically computed from new video to update UF and the choice of the optimal operation. Figure 4(2) lists the comparison result. Our method exhibits significantly higher accuracy (up to 89%).

## 4. CONCLUSIONS

Utility function based transcoding framework is an efficient solution for real-time media adaptation. However, to date generation of utility function is either time-consuming or limited to analytical approximation applicable to specific codecs. In this paper we present a general content-based utility function prediction model using automatic content feature extraction, and regression over clustering and classification. Our experiment results demonstrate very promising results in prediction accuracy over different types of video content.

## 5. ACKNOWLEDGEMENT

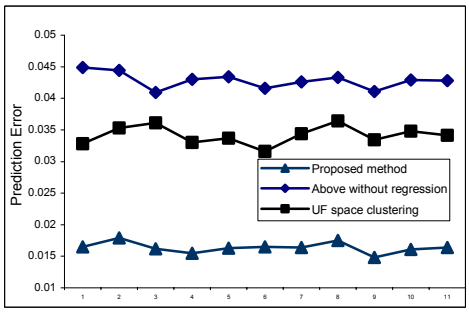
This work has been supported by Electronics and Telecommunications Research Institute of Korea.

Table 1: Media pool summarization

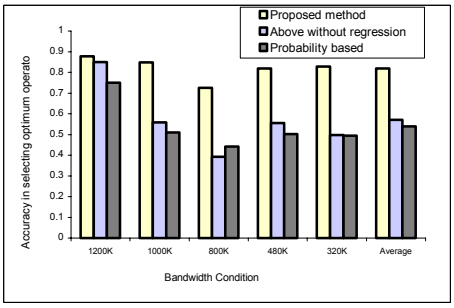
| | |
|---|---|
| Video source (Totally 2066 clips) | *1. A Beautiful Mind*<br>*2. Crouch Tiger Hidden Dragon*<br>3. *Taxi II* |
| Clip length | 1 seconds |
| Image size | 352 x 240 pixels |
| Video format | 30fps, MPEG4 with TM5 rate control |
| GOP structure | N=15, M=3 |

Table 2: Algorithm specification

| | |
|---|---|
| Clustering | KHM with $p$=2.5, $K$=16 |
| Classification | DAGSVM: $C$=100, Kernel=RBF with $\gamma$= 0.5 |
| Regression | Trained by LSE algorithm |




(1)




(2)

Figure 4: Performance Evaluation of Utility Function (1) L2 distance (2) accuracy in choosing the optimal operator

## Reference

[1] J.-G. Kim, Y. Wang, S.-F. Chang, K. Kang, J. Kim, "Description of utility function based optimum transcoding", ISO/IEC JTC1/SC29/WG11 M8319 Fairfax May 2002.

[2] J.-G. Kim, Y. Wang, S.F. Chang, "Content-Adaptive Utility-Based Video Adaptation", to be appeared in IEEE ICME-2003.

[3] P. Bocheck, Y. Nakajima and S.-F. Chang, "Real-time Estimation of Subjective Utility Functions for MPEG-4 Video Objects", PV'99, New York, USA, April 26-27, 1999.

[4] S.F. Chang, "Optimal Video Adaptation and Skimming Using a Utility-Based Framework", IWDC-2002, Capri Island, Italy, Sept. 2002.

[5] B. Zhang, "Generalized K-Harmonic Means-boosting in unsupervised learning", HP Lab Technical Report, Oct 2000.

[6] J.C. Platt,, N. Cristianini, and J. Shawe-Taylor, "Large Margin DAGs for Multiclass Classification", NIPS 12, pp. 547-553, MIT Press, 2000.

[7] A. Eleftheriadis, *Dynamic Rate Shaping of Compressed Digital Video*, Doctoral Dissertation, Graduate School of Arts and Sciences, Columbia University, June 1995.

[8] A. Vetro, Y. Wang, and H. Sun, "Rate-Distortion Optimized Video Coding Considering Frameskip," IEEE ICIP-2001. Thessaloniki, Greece, Oct. 2001.

[9] J.-J. Chen and H.-M. Hang, "Source model for transform video coder and its application— Part II: Variable frame rate coding," *IEEE Trans. Circuits Syst. Video Techno.* Vol. 7, pp. 187-298, Apr. 1997.

[10] E.C. Reed and J.S. Lim, "Optimal multidimensional bit-rate control for video communications," *IEEE Trans. Image Processing*, vol. 11, no. 8, pp. 873-885, Aug. 2002.