Polishing: A Technique to Reduce Variations in Cached Layer-Encoded Video

Michael Zink, Oliver Heckmann, Jens Schmitt, Andreas Mauthe, Ralf Steinmetz
Multimedia Communications Lab
Merckstrasse 25
D-64283 Darmstadt, Germany
+49-6151-166886
zink@kom.tu-darmstadt.de

ABSTRACT

In this paper we introduce a technique called polishing to reduce variations in cached layer-encoded video by identifying those segments which can be removed without diminishing the perceptual quality. In fact, polishing even allows to increase the perceptual quality of a layer-encoded video. This can be used for perceptual quality improved playout from a cache towards a client as well as for perceptual quality aware cache replacement decisions. We devise an optimal scheme to polish layer-encoded videos but show that a simple heuristic achieves similar performance in simulations. By means of simulations we furthermore show how polishing can improve the quality of the cached content.

Keywords: Video Streaming, Layer Encoded Video, Caching.

1 INTRODUCTION

The combination of caching and adaptive streaming bears the disadvantage that a layer-encoded video can only be cached according to the available bandwidth of the path between server and cache if a congestion controlled transport mechanism is applied. The created copy of the video stored on the cache might, depending on the path's condition, contain a large amount of layer variations. Forwarding such a video object towards a client might be annoying for the watching user [1].

In this paper, we present a new technique, called polishing, in which a cache considers to send only a subset of the segments of a locally stored object in order to reduce layer variations in a video which is streamed towards a client. Investigations by performing a subjective assessment of layer variations in videos [1] have shown that it can be beneficial to omit the transmission of certain segments, especially if the amount of layer variations is reduced. (For the definition of segment and layer variations see Figure 4 and Figure 5.)

At first, this might sound rather strange since some information is not transmitted at all and, thus, the PSNR of the video is reduced. Yet, our investigations have shown that, despite reducing the PSNR, reducing layer variations can increase the perceived quality of a video, if done carefully. I.e., in the subjective assessment we previously performed, a layered video with a lower PSNR but less layer variations is given a higher perceptual quality than the same video with a higher PSNR and a larger amount of layer variations.

We specify an optimal polishing technique and compare this technique with a less complex heuristic by performing a series of simulations. In additional simulations we investigate the effect of using polishing for fine-grained cache replacement strategies.

2 POLISHING AND ITS APPLICATIONS

In this section, the fundamental ideas behind polishing are presented. We show how it can be applied to increase the quality when an incomplete video object is transmitted from the cache to the client and afterwards present how polishing can also be used for cache replacement.

2.1 Transport

As aforementioned, the amount of variations in a layer-encoded video that is stored on the cache can be reduced by omitting the transmission of certain segments from the cache to the client.

The challenge of polishing is to identify segments that should not be transmitted in order to increase the perceived quality of a video object at a client. Polishing is a new technique that determines those segments that should not be transmitted from a cache to a client with the goal to increase the perceived quality of a video at a client. After the caching process of a certain video object is finished the polishing algorithm (as described in Section 4) is executed. This algorithm identifies segments that should not be played out in subsequent streaming sessions from cache to clients. Note, that the identified segments are not removed from the cache. If this video object is requested, the transport mechanism of the cache will take into account the information gained by polishing and decide which data will be sent to the client and which not.

2.2 Fine-grained Cache Replacement

Since polishing identifies segments that are of less importance in relation to quality, the information gained by applying this technique can also be used for cache replacement strategies. Assuming that the storage space on a cache is exhausted and data has to be removed from the cache in order to allow the caching of a new object, segments identified by polishing can be deleted. A fine-grained replacement scheme based on segments can increase the efficiency of the cache as shown by [2]. If, in addition, popularity information is taken into account, e.g., segments of the least popular object are deleted first, the quality of the cached content is also based on its popularity.

2.3 Spectrum

In our previous work on retransmission scheduling (RS) [3] we have developed a metric for the quality of layer-encoded video and have verified the validity of this metric by a subjective assessment [1]. Here, we briefly introduce this metric, which is referred to as spectrum. The spectrum is used as a metric for the quality of a layer-encoded video before and after polishing. The definition of the spectrum is given in (1).

$$s(v) = \sum_{t=1}^{T} z_t \left(h_t - \frac{1}{\sum_{i=1}^{T} z_i} \left(\sum_{j=1}^{T} z_j h_j \right) \right)^2 \qquad (1)$$

h_t – number of layers in time slot t, $t=1, \ldots, T$

z_t – *indication of a step in time slot t* , $t=1, \ldots, T$

Figure 1 Definition of the spectrum

A small spectrum indicates a good quality while the quality worsens with an increasing spectrum. Although, the spectrum is a good metric for RS it has a one important disadvantage in combination with polishing. The spectrum becomes 0 for the case that no layer changes occur irrespective of how many layers the video object consists. This is uncritical in the case of RS where new segments are added to the video and achieving a spectrum of 0 always leads to a better quality than the one of the originally cached object. It is different in the case of polishing in which segments are discarded. If the decision to drop certain segments would be solely driven by the spectrum, polishing could lead to the fact that all segments of incomplete layers are discarded. This effect, which we denote as over polishing (see Figure 2), is undesirable because it decreases the quality in a drastic manner. In Section 4, we present a new algorithm for polishing that avoids the problem of *over polishing*.

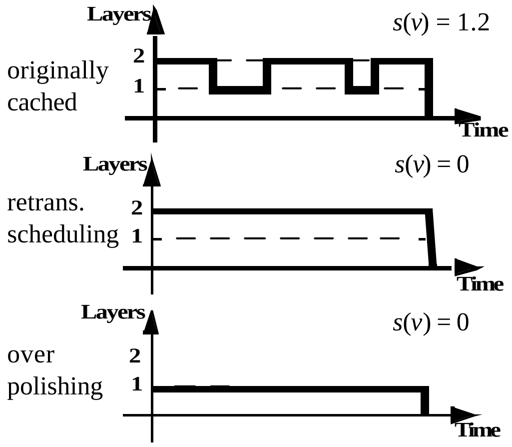

Figure 2 Polishing vs. retransmission scheduling

2.4 Example

Here, we give a simple example to demonstrate the effect of polishing. We assume that a layer-encoded video is stored in the cache as shown in Figure 3. The variations in the amount of layers is caused by a congestion controlled transmission between server and cache which results from the network conditions on the path between both (see Section 5 for details on how we simulated the congestion controlled transmission). Figure 3 also shows the layer-encoded video as it would be transmitted to the client after the polishing algorithm has been performed. The third shape in Figure 3 shows the result of a simple heuristic where only the highest (5th) layer is dropped. Further details about the optimal polishing algorithm are given in Section 4. Figure 3 shows a significant reduction in layer variations due to polishing. A reduced spectrum (28 compared to 193 of the originally cached object) indicates a better perceptual quality compared to the one of the originally cached object. The effect of over polishing has been avoided.

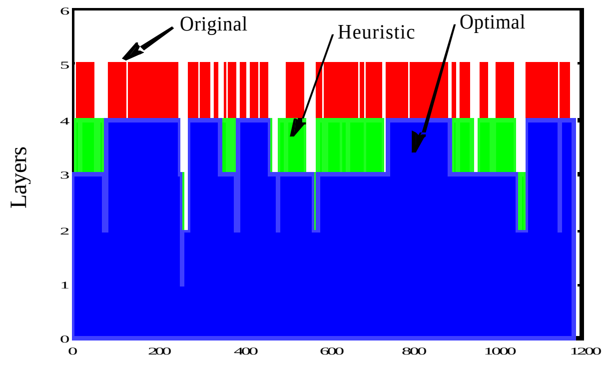

Figure 3 Comparison of originally cached and polished (heuristic and optimal) video object

3 RELATED WORK

There is much less work on the caching of layer-encoded video than work on caching of monolithic video (e.g., MPEG-1) objects. Existing work on caching layer-encoded video is only concerned about cache replacement and not the playout to the client. Polishing is, to the best of our knowledge, the first investigation on this topic.

Rejaie et al. [4] were the first that presented an approach for the caching of layer-encoded video. The video is streamed in a congestion controlled manner (using the RAP protocol [5]) from a server through a proxy into a client. Missing segments on a cache, caused by losses and rate adaptation, are prefetched in a demand driven fashion to improve the quality of a cached video. A cache replacement algorithm is presented that works on a fine-grained level, which allows the dropping of single segments of a layer. Simulations reveal that the quality of a cached video is directly related to its popularity. In contrast to our approach, where removable segments are identified by polishing, segments are dropped in a very simple manner. Each layer of a video is regarded separately. Beginning at the top layer, for each single layer segments are removed from end to beginning, while in our case the whole video is regarded for the removal of segments.

Similar investigation to [4] have been performed by Paknikar et al. [6], with the difference, that only complete layers can be dropped. In addition, their approach consists of a cluster of caches which is managed by a broker and is designed for a high-speed local area network.

An analytical investigation was performed by Kangasharju et al. [7]. Their main goal was to gain better insights on the effects of cache space and link bandwidth on the cache performance. In contrast to [4] only complete layers can be stored or removed from the cache in order to keep the problem mathematically tractable. Congestion control on the link between the server and the proxy and the proxy and the client is not assumed.

A prototype implementation of an adaptive multimedia cache is presented in [2]. The authors modified the Squid web proxy cache in order to be able to perform the caching of layer-encoded video. Based on this prototype initial experiments were performed. These preliminary results tend to confirm the simulative results from [4].

Quality based caching [8] is an additional approach for partial caching which assumes that metadata information about the quality of a scalable video is available. E.g., the metadata would provide that removing the top layer of a 5 layer video would reduce the quality of the video by 20%. It is not clear how this information can be obtained.

4 OPTIMAL POLISHING

To polish a layer-encoded video means to minimize the spectrum while at the same time maximizing the number of segments played out to the client and could thus be regarded as a multi-objective optimization problem. Two characteristics of that optimization problem make it hard to be treated directly: on the one hand, the two competing optimization goals and, on the other hand, the non-linear quadratic form of the spectrum. Therefore, we decided to resort to a substitute metric for the spectrum, namely layer variations, and a utility based approach where we introduced parameters for the relative weighting between the two competing goals of polishing.

Polishing - which under theses prerequisites means maximizing the playback utility of a video - can be formulated as the mixed integer programming problem [9] given in Figure 4.

The two parameters u_l and p describe the utility of the video playout. u_l is the utility for receiving layer l (and all lower layers) in one period , obviously, the more layers are played back the higher the utility. p describes the utility loss for a layer change. By including u_l into the optimization process the over polishing effect described in Section 2.3 is avoided. p prohibits quality loss by changing the playback layer too often.

The variables h_t contain the highest layer of the polished video at time t, it can never be higher than the highest cached layer (see constraint (5)). The binary variable z_t is needed to account for layer changes in the target function. z_t is forced to one by constraints (3) and (4) when the highest layer of the polished video changes. Binary variable b_{tl} stores whether a layer l is included in the polished video in period t or not, constraint (6) expresses its relationship with the highest layer h_t.

This problem can be solved with standard techniques like Branch and Bound and the Simplex algorithm [9]. We used the commercial mathematical programming solver Ilog CPLEX [10] to solve the problem.

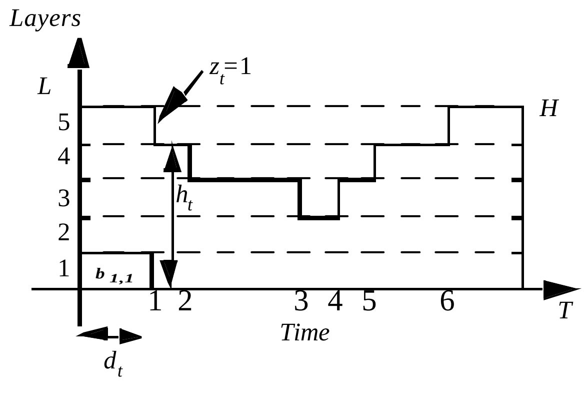

Figure 4 Cached layer-encoded video

5 SIMULATIONS

To verify if polishing is a valid approach and to obtain further information on the influence of the utility factors u_l and p we decided to perform a series of simulations. An additional goal was also to investigate how a simple heuristic performs in comparison to the optimal polishing algorithm. This heuristic simply drops one ore more adjacent layers, beginning from the top.

The simulations are performed in the following manner:

Indices:

$l=1, ..., L$ Layer of the video

$t=1, ..., T$ Period t

Parameters:

h_t^{cached}	number of the highest layer that is cached for period t, all lower layers are cached in period t, too.
H	sufficientlz large number (H>L)
d_t	length of period (in seconds)
u_l	utility of receiving layer l for one second (if video is playd back for one sec. On layer 3 it generates a ultility of $u_1+u_2+u_3$)
p	utility loss for a change in the number of layers that are played back

Variables:

h_t	the layer the video is played back in period t
z_t	binary variable, one if a layer change occurs at the beginning of period t, zero otherwise
b_{tl}	binary variable, one if is played back in period t at layer l or higher, zero otherwise

Optimization problem:

$$Max\left(\sum_{l,t} u_l d_t b_{tl} - \sum_t p z_t\right) \quad (2)$$

subject to

$$h_t - h_{t-1} \le H z_t \qquad \forall t = 2,...,T \quad (3)$$

$$h_{t-1} - h_t \le H z_t \qquad \forall t = 2,...,T \quad (4)$$

$$0 \le h_t \le h_t^{cached} \qquad \forall t = 1,...,T \quad (5)$$

$$l b_{tl} \le h_t \qquad \forall t = 1,...,T \qquad \forall l = 1,...,L \quad (6)$$

$$b_{tl} \in \{0,1\} \qquad \forall t = 1,...,T \qquad \forall l = 1,...,L \quad (7)$$

$$z_t \in \{0,1\} \qquad \forall t = 1,...,T \quad (8)$$

For each simulation an instance of a layered video on the proxy cache is randomly generated. Here, we modeled such a layer-encoded video instance as a simple finite birth-death process since it is the result of the congestion-controlled video transmission which restricts state transitions to direct neighbor states. {0,..., L} is the state space and birth and death rate are chosen equal as 1-1/SQRT(3) (for all states) which results in a mean length of 3 time units for periods with stable quality level . We use a discrete simulation time where one unit of time corresponds to the transmission time of a single segment. In Figure 3 (Original), an example video instance generated in this way is given. On each instance of a layer-encoded video created as described above our polishing algorithm is performed. Before and after polishing the spectrum of the video is calculated in order to obtain information about the quality change. An example of such a simulation step is shown in Figure 3 (before and after polishing) with the following set of parameters: $u_l=1$, $u_l=1$, $u_l=1$, $u_l=1$, $u_l=1$ and $p=8$.

5.1 Utility Parameters

To obtain better insights in the influence of the parameters u_l and p we ran a series of simulations. The results of this simulation are presented in Figure 6 and Figure 7. For each parameter set 100 video objects were randomly created and polished as described above. The average spectrum and the average total amount of segments were calculated before and after running the polishing algorithm. Figure 6 shows the results for three different simulations and the spectrum for two versions of the heuristic.

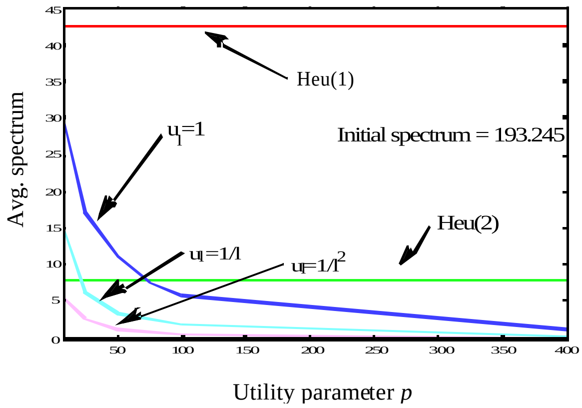

Figure 6 Spectrum

In the first version the top (Heu(1)) layer is dropped and in the second the two top (Heu(2)) layers are dropped. Then we calculated the average spectrum and average amount of segments of all 100 resulting objects. As can be derived from both figures the heuristic has the disadvantage that it is static while, in the case of polishing, the selection of the parameters u_l and p influences spectrum and amount of segments of the polished video object. On the other hand, the heuristic is simple and can be applied with little computational effort and the obtained results are fairly close to the ones of the optimal polishing.

In the case of polishing, we performed the different classes of simulation with $u_l=1$, $u_l=1/l$, and $u_l=1/l^2$ respectively.

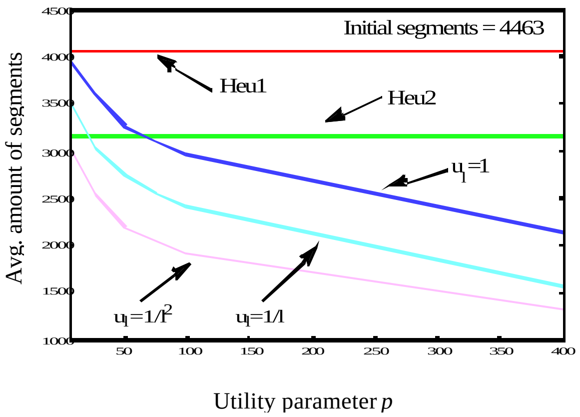

Figure 7 Avg. amount of segments per video object

5.2 Replacement

In another set of simulations we investigate how polishing can influence the cache replacement. This should be seen as an initial step to show the general applicability of polishing for cache replacement. Further investigations have to show, if a relation between popularity and quality of the cached video can be achieved by the application of polishing.

At the beginning of the simulation we assumed that the cache was initially filled with 50 unpolished video objects, consuming all of the cache's storage space. Then, 50 additional objects should be incrementally, i.e., one per time slot, stored on the cache. To generate the additional storage space for these objects in the first step the already cached objects are polished. If the space gained by polishing or in the case that all objects are already polished, a cached object will be removed in FIFO manner. This simulation was performed for the parameter set *p=50* and $u_l=1$. In an additional simulation (no-polishing), objects are not polished but simply removed. We also simulated the cache replacements for the two versions of the heuristic. Figure 8 depicts the spectrum for each of the simulations, while the average amount of segments per object is shown in Figure 9.

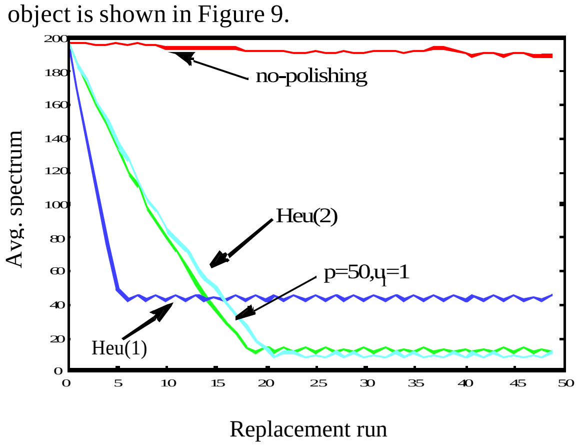

Figure 8Spectrum for cache replacement

An interesting result is revealed by the comparison of both graphs for the case *p=50* and $u_l=1$, where a high reduction of the spectrum results in a moderate reduction of the average amount of segments per cached object. Compared to a cache replacement which does not incorporate polishing the total amount of video objects stored on the cache is higher in the polishing simulations (see Table 1). Thus, integrating polishing into the cache replacement method is beneficial since a higher amount of video objects can be cached and layer changes are reduced (smaller spectrum). Based on the parameters that can be chosen for the polishing method, the behavior of the cache replacement method can be influenced. E.g., a cache operator can control, with the aid of this parameters, if more objects in a lower quality or vice versa should be cached. Also in this simulation, the results of the heuristic are close to the optimal polishing result.

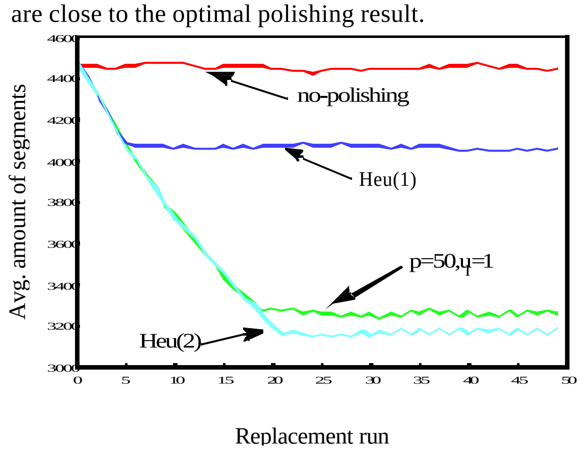

Figure 9 Avg. amount of segments per object for cache replacement

Simulation	No-polish	P=50, ul=1	Heur(1)	Heur(2)
Objects	50	68	55	70

6 Conclusions

In this paper, we have presented a new technique to reduce the variations in layer-encoded video for the playout from the cache to the client. Simulative investigations of polishing indicate the general applicability and the benefits of this technique which can also be used for cache replacement.

In future work, we will refine the simulations on cache replacement by introducing a model for popularity based user requests. Afterwards we plan to integrate polishing in our experimental streaming platform. Additionally, it might be interesting to investigate if polishing can also be applied at the client's buffer, since additional variations can be introduced on the path between cache and client.

REFERENCES

[1] M. Zink, J. B. Schmitt, and R. Steinmetz. Layer Encoded Video in Scalable Adaptive Streaming, 2003. Accepted for publication in IEEE Transactions on Multimedia.

[2] R. Rejaie and J. Kangasharju. Mocha: A Quality Adaptive Multimedia Proxy Cache for Internet Streaming. In *Proceedings of the 11th International Workshop on Network and Operating System Support for Digital Audio and Video, Port Jefferson, New York, USA*, pages 3–10, June 2001.

[3] M. Zink, J. Schmitt, and R. Steinmetz. Retransmission Scheduling in Layered Video Caches, April 2002. Proceedings of the International Conference on Communications 2002, New York, New York, USA.

[4] R. Rejaie, H. Yu, M. Handley, and D. Estrin. Multimedia Proxy Caching for Quality Adaptive Streaming Applications in the Internet. In *Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies 2000 (INFOCOM'00), Tel-Aviv, Israel*, pages 980–989, March 2000.

[5] R. Rejaie, M. Handley, and D. Estrin. RAP: An End-to-End Rate-based Congestion Control Mechanism for Realt ime Streams in the Internet. In *Proceedings of the Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies 1999, New York, NY, USA*, pages 395–399, March 1999.

[6] S. Paknikar, M. Kankanhalli, K. Ramakrishnan, S. Srinivasan, and L. H. Ngoh. A Caching and Streaming Framework for Multimedia. In *Proceedings of the ACM Multimedia Conference 2000, Los Angeles, CA, USA*, pages 13–20, October 2000.

[7] J. Kangasharju, F. Hartanto, M. Reisslein, and K. W. Ross. Distributing Layered Encoded Video through Caches. *IEEE Transactions on Computers*, 51(6):622–636, June 2002.

[8] S. Podlipnig and L. Böszörmenyi. Replacement strategies for quality based video caching. In *International Conference on Multimedia and Expo, Lausanne, Switzerland*, pages 49–52, August 2002.

[9] F. S. Hillier and G. J. Lieberman. *Operations Research*. McGraw-Hill, 1995.

[10] Ilog Cplex. Mathematical programming optimizer. http:// www.ilog.com/products/cplex