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

Michael Zink, Oliver Heckmann, Jens Schmitt, Ralf Steinmetz
KOM, Darmstadt University of Technology, Merckstrasse 25, 64283 Darmstadt, Germany

ABSTRACT

In this paper we present polishing, a novel technique to maximize the playback utility of a streamed layer-encoded video. Polishing reduces the amount of layer variations in a cached layer-encoded video before streaming it to a client if this increases the quality of the video. Polishing can also be used as a cache replacement strategy for removing the parts of layer-encoded videos on a cache that harm the quality least. This paper presents optimal algorithms for both applications and simulation results.

Keywords: Video Streaming, Layer-Encoded Video, Cache Replacement

1 INTRODUCTION

Caching and the use of layer-encoded video are two important techniques for scalable, adaptive video streaming [1]. Unfortunately, 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 viewer.

In this paper, a new technique, called polishing, is presented. With polishing a cache considers sending only a subset of the segments of a locally cached object in order to reduce layer variations to the client. Investigations by performing a subjective assessment of layer variations in videos [2] have shown that the subjective quality of a video can be significantly increased by omitting the transmission of certain segments, especially if that reduces the amount of layer variations. At first, this might sound rather strange since some information is not transmitted at all and, thus, the average PSNR of the video is reduced. Yet, the investigations in [2] have shown that, despite reducing the average PSNR, sensibly reducing layer variations can increase the perceived quality of a video, because layer variations are often annoying for the viewer.

The polishing technique presented in this paper can be useful in two cases. In the first case, a connection to the server that stores the original version of the video object cannot be established but a cached version of the video object already exists on the local cache. Assuming that not all missing segments are retransmitted so far polishing can be applied to increase the quality of the streamed video.

In the second case, the storage space at the local cache is exhausted and new video objects should be stored on the cache. Instead of removing complete, less popular objects from the cache polishing can be applied to remove a certain amount of segments from all videos depending on their popularity. Thus, new objects can be added to the cache's storage while only segments of certain objects are removed instead of complete objects, the quality of all videos suffers the least possible way.

In the next section we discuss the cases in which polishing is useful in more detail. Then we discuss related work in Section 3. In the fourth section we present an optimal polishing algorithm for video transport and simulation results. In the fifth section we discuss the use of polishing for cache replacement strategies, present an optimal algorithm and again some simulation results. We conclude with a summary and outlook.

2 APPLICATIONS FOR POLISHING

In this section, the fundamental ideas behind polishing are presented. It is shown how polishing can be applied to increase the quality when an incomplete video object is transmitted from the cache to the client. In addition, a cache replacement method based on polishing is presented.

2.1 Polishing for Transmission

As mentioned above, 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 necessarily removed from the cache. If this video object is requested from a client, 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.

The information gained by polishing can, e.g., be used to stream a polished version of a video object if retransmission scheduling [1] cannot be performed. This might be the case if the server or the link between the server and the cache is down or the server does not have additional capacity to allow retransmission in addition to already active streams. The general concept of polishing in such a case is shown in Figure 1.

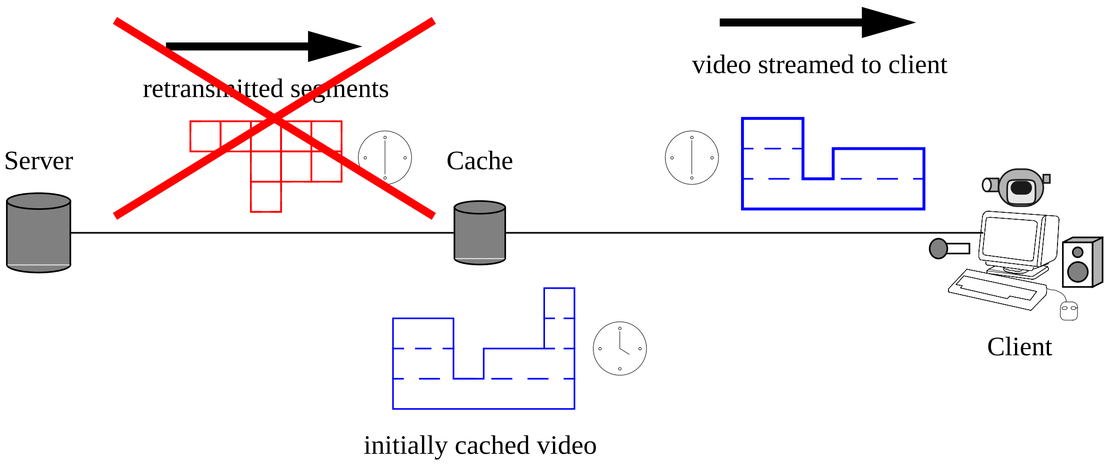


Figure 1: Polishing in the case of impossible retransmissions

2.2 Polishing for Fine-grained Cache Replacement

Since polishing identifies segments that are of less importance or even harmful for the quality of the video playback, 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 [3]. 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. Figure 2 shows the basic principle of cache replacement with the aid of polishing. In this example four objects are stored on the cache. After performing polishing, 15 segments are identified by the algorithm that can be removed from the cache and, thus, new storage space is created to store an additional object on the cache. The second alternative in Figure 2 shows the case in which popularity based caching is applied. The popularity of the cached object decreases from left to right influencing the amount of segments that are identified for removal by polishing, leading to the fact that objects with a higher popularity are cached in a better quality.

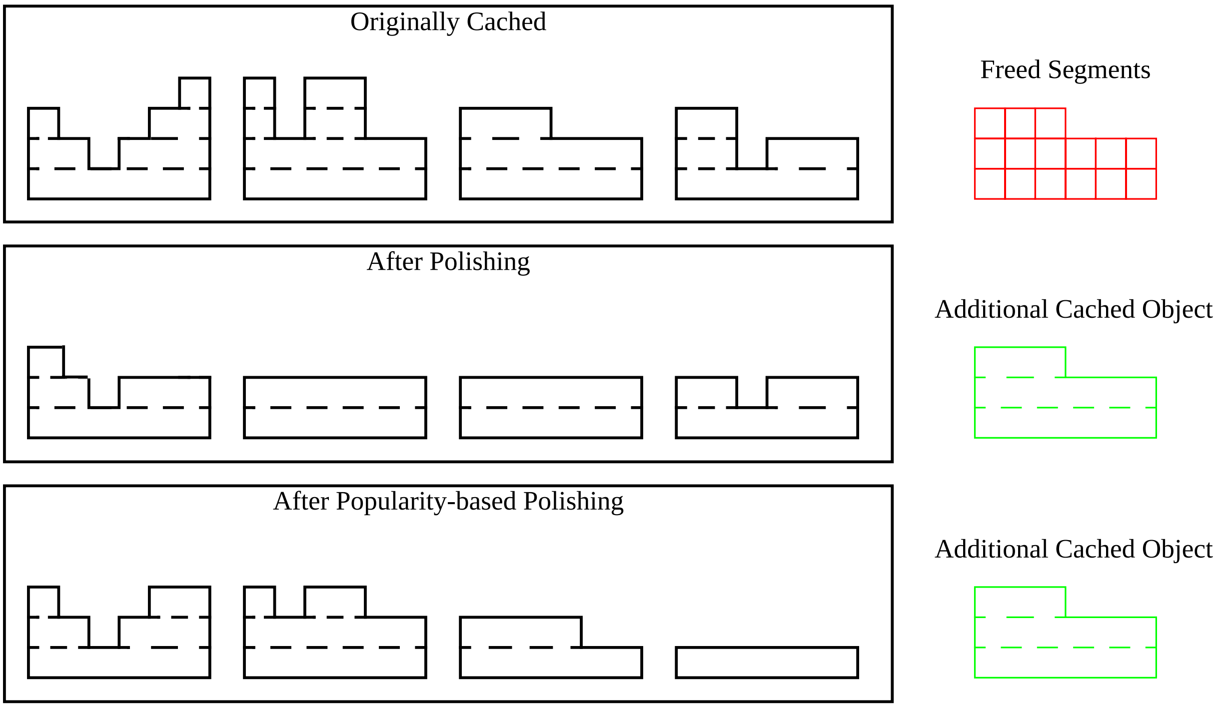


Figure 2: Cache replacement with the aid of polishing

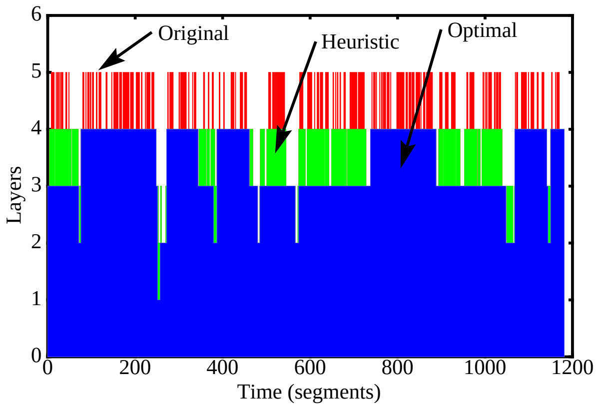


Figure 3: Comparison of originally cached and polished (heuristic and optimal) video object

2.3 Example

We now give a simple example to demonstrate the effect of polishing. It is assumed 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 4.2 for details on how the congestion controlled transmission was simulated). Figure 3 also shows the layer-encoded video as it would be transmitted to the client after our optimal 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. The example also shows the storage space that could be freed on a cache based on the information that is gained by polishing. In comparison, for the simple heuristic, that simply drops the top layer, the amount of storage space that can be freed due to optimal polishing is higher while at the same time the remaining quality of the video is better. A simulative investigation on polishing as a mechanism for cache replacement is given in Section 5.

3 RELATED WORK

In our previous work on retransmission scheduling (RS) [1] we have developed a metric for the quality of layer-encoded video and have verified the validity of this metric by a subjective assessment [4]. 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, $t = 1,..., T$
z_t - indication of a step in time slot t, $z_t \in \{0,1\}$, $t = 1,..., T$

Figure 4: Definition of the spectrum

A small spectrum indicates a good quality while the quality worsens with an increasing spectrum. Looking at the example of Figure 3 the spectrum of the polished video is reduced significantly (28 compared to 193 of the originally cached object) indicating a better perceptual quality compared to the one of the originally cached object.

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*, is undesirable because it decreases the quality in a drastic manner. The effect of *over-polishing* lead us to this work. In *Section 4*, we present a new algorithm for polishing that avoids the problem of *over-polishing.*

The work on polishing presented in this paper is, to the best of our knowledge, the first investigation on this topic. There are however some works related to cache replacement strategies for layer-encoded video. Rejaie et al. [5] introduce a fine-grained cache replacement mechanism that allows the deletion of single segments. Each layer of a video is regarded separately. Beginning at the top layer of a video, 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. Thus, in the case of Rejaie's approach segments of the top layer of the cached video are removed until none of the segments of this layer is left. If more space on the cache is needed, this process will be continued on the next lower layer. In the case of polishing segments from all available layers can be removed independently. The approach presented by Paknikar et al. [6] allows only the removal of complete layers which is somewhat similar to the heuristic presented in Section 4.2. Also in the analytical investigation performed by Kangasharju et al. [7] only complete layers can be dropped.

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%. The authors leave open how this necessary metadata can be obtained.

4 POLISHING FOR TRANSMISSION

4.1 Optimal Algorithm

Polishing for the transmission of a single video maximizes the playback utility of a video. The resulting optimization problem can be formulated as the mixed integer programming problem [9] given in Figure 5.

Indices:

$l = 1, \ldots, L$ -layer of the video

$t = 1, \ldots, T$ -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 - sufficiently large number ($H \geq L$)

d_t - length of period (in seconds)

u_l - utility of receiving layer l for one second (if video is played back for 1 sec. on layer 3 it generates a utility 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 video 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

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

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

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

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

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

$$z_t \in \{0, 1\} \qquad \forall t = 2, \ldots, T \quad (8)$$

Figure 5: Optimization model

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 t. 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 3 is avoided. p prohibits quality loss by changing the playback layer too often.

The variable h_t contains 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. The 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 exactly with standard techniques like Branch and Bound and the Simplex algorithm [9]. We used the commercial mathematical programming solver Ilog CPLEX [10] to find the optimal solution for the problem.

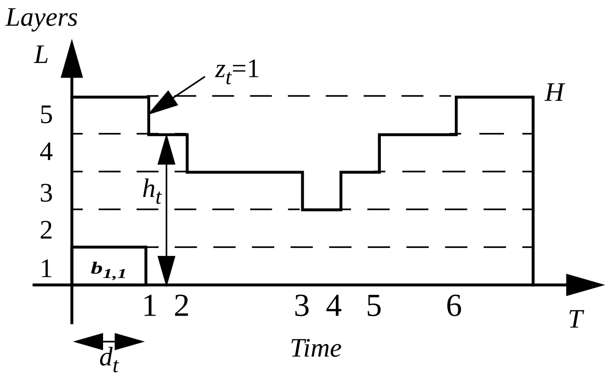


Figure 6: Cached layer-encoded video

4.2 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 a series of simulations were performed. 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 completely, beginning from the top.

The simulations are performed in the following manner: 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 [1]. 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. The polishing algorithm was implemented using the mathematical programming solver Ilog CPLEX [10]. 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: $u_1 = 1$, $u_2 = 1$, $u_3 = 1$, $u_4 = 1$, $u_5 = 1$ and $p = 8$.

4.2.1 Investigation of the Utility Parameters

To obtain better insights in the influence of the parameters u_l and p a series of simulations with varying values for those parameters were performed. The results of this simulation are presented in Figure 7 and Figure 8. 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 7 shows the results for three different simulations and the spectrum for two versions of the heuristic. In the first version the top (Heu(1)) layer is dropped and in the second the two top (Heu(2)) layers are dropped. Then the average spectrum and average amount of segments of all 100 resulting objects are calculated. As can be derived from both figures, the heuristic has the disadvantage that it is static while, in the case of optimal 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, the different classes of simulation were performed with $u_l = 1$, $u_l = 1/l$, and $u_l = 1/l^2$ respectively.

---

1. We have to admit that the parameter choice is rather arbitrary. However, simulations with other values showed no significant impact on our results.

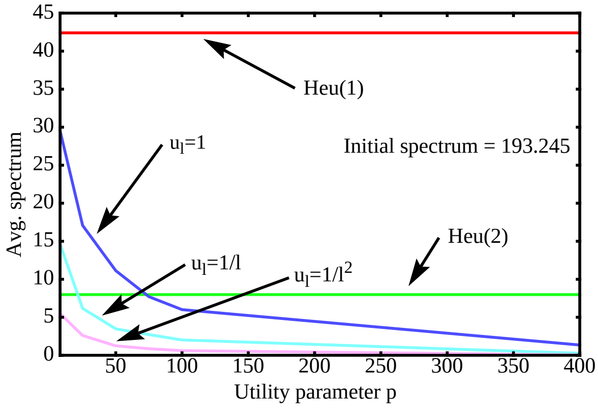


Figure 7: Avg. Spectrum

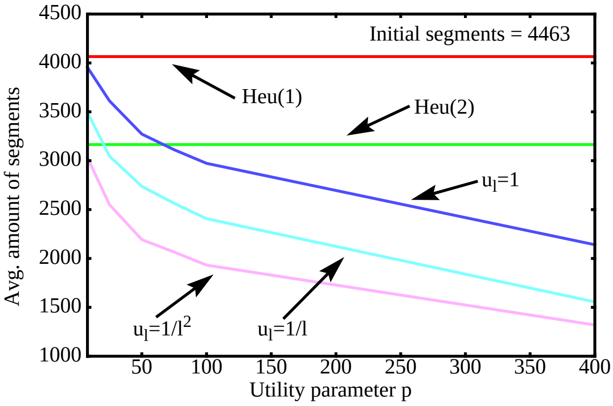


Figure 8: Avg. amount of segments per video object

In this specific simulation the heuristic turns out to be an alternative compared to the optimal polishing. A closer look at Heu(2) shows that the spectrum is significantly reduced while the amount of segments is reduced by a third. This result can also be achieved by applying the optimal polishing algorithm but at the prize of a higher computational effort.

4.2.2 Simulations with TFRC Traces

An additional series of simulations was performed with the difference that the initial videos used as input for the polishing simulation were generated in a different way. In this case TFRC traces generated by a ns2 simulation were used to generate the initial video. The bandwidth information generated by the ns2 simulation is used to determine how many layers of a video can be transmitted during a certain period. This complies with a cached layer-encoded video that has been transmitted via TFRC. A detailed description on this simulation environment is given in [11].

Figure 9 depicts the resulting average spectrum obtained by the application of the heuristic and the optimal polishing algorithm. Compared to the results of the simulation where the videos are generated randomly the results of optimal polishing are similar. Comparing the results for Heu(2) (Figure 7 and Figure 9) with each other shows that the reduction in the spectrum is not as high with the TFRC based simulation. This difference is caused by the fact that TFRC also starts with a slow-start (identical to TCP) and, for a short while until the occurrence of a loss event, the sending rate can become quite high. Thus, reducing the top or the two top layer might affect only a small portion of the video. Due to the slow-start chracteristics the maximum amount of layers might occur only once during the streaming of the video. In simulations presented in the following section it is shown that the static behavior of the heuristic can reduce its efficiency.

The difference in the average amount of segments for the random (Figure 8) and TFRC-based (Figure 10) simulation is caused by the fact that the TFRC traces are shorter in duration. The length of such a trace is equivalent to 400 time units while in the case of random based simulation the layer-encoded video has a length of approximately 1200 time units.

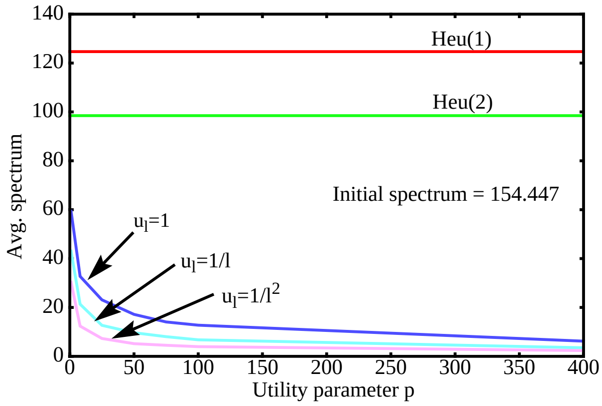


Figure 9: Avg. spectrum (TFRC based)

Nevertheless, the behavior of the optimal polishing against u_l and p is almost similar for both simulation types. The average amount of segments decreases with a increasing p and a decreasing u_l .

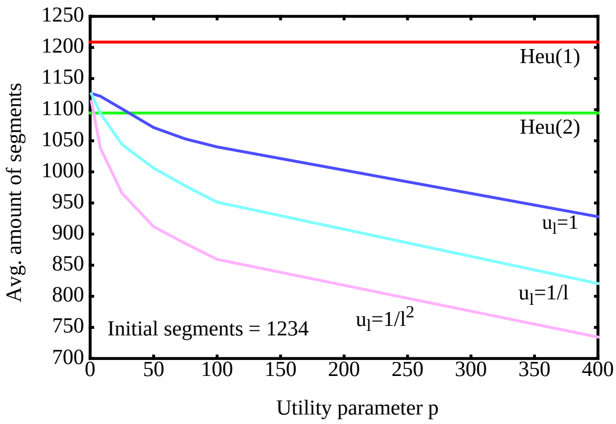
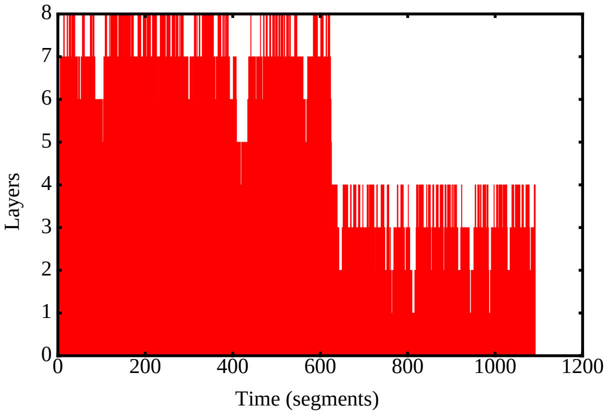


Figure 10: Avg. amount of segments per video object (TFRC based)

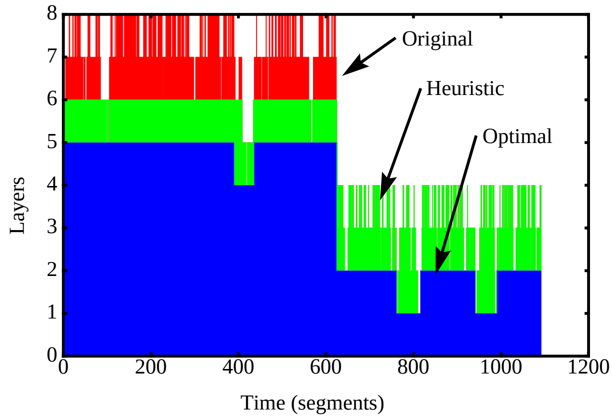
4.3 Layer-Encoded Video with Different Quality Regions

An example for a video object, which consists of two different quality regions, is shown in Figure 11. There can be several reasons that can cause the creation of such a video object on the cache. One possibility is a limited bandwidth between server and cache caused by competing traffic. In the given example the transmission of the competing traffic started after half of the video is already streamed from the server to the cache. The occurrence of two major quality regions in a cached video is rather arbitrary, since several of such *regions* with different quality levels can occur depending on the situation on the path between server and cache. Nevertheless, the chosen example is sufficient to demonstrate the drawback of the heuristic presented above. As can be seen in Figure 11 (b), the disadvantage of the heuristic is the fact that only the region with the better quality (higher amount of layers) is polished. In this example the 2 top layers are dropped. This effect does not occur with optimal polishing where both regions are polished. Applying the heuristic might be annoying for the viewer. The already existing quality decrease between the two regions is even intensified by the high amount of quality changes in the second half of the video. With optimal polishing, quality variations are reduced in both regions and, thus, the quality decrease between the two regions is not that intense. Comparing the two resulting spectra (see Table 1) of the second region demonstrates the effect mentioned above. The



Originally cached layer-encoded video

(a)



Originally cached and polished layer-encoded video

(b)

Figure 11: Polishing for two-staged layer-encoded video

spectrum in the second region is significantly higher for the heuristic than for optimal polishing, while for the first region both spectra are identical.

	Region 1	Region 2
Heuristic	6	86.81
Optimal Polishing	6	1.2

Table 1: Spectrum per region

Also for the case of the layer-encoded video consisting of two quality regions a series of simulations were performed. The simulation environment is identical to the one described in Section 4.2 with a slightly modified creation process for the initially cached video that results in quality variations as shown in Figure 11 (a). Similar to the simulations described in Section 4.2.1, 100 video objects are randomly created and the average spectrum and total amount of segments are calculated. Figure 12 shows the average spectrum for the optimal polishing depending on the parameters u_l and p. The resulting spectra for the three variations of the heuristic are always larger than the ones for optimal polishing.

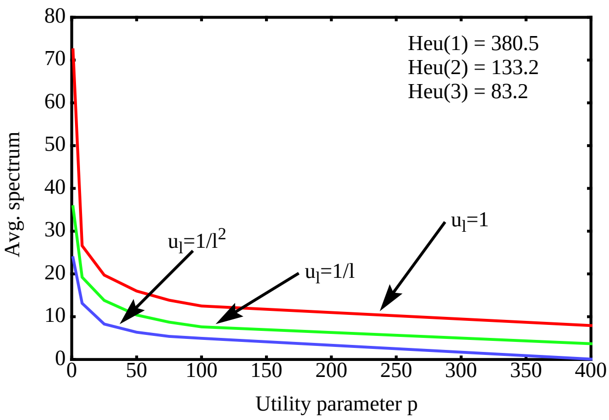


Figure 12: Avg. Spectrum

The comparison of the average amount of segments per video object (see Figure 13) shows that the three variations of the heuristic (Heu(1), Heu(2), and Heu(3)) result in higher values than the optimal polishing. An interesting case is the one for $u_l = 1$ and $p < 50$ where the spectrum for the optimal polishing is lower than the one of Heu(3) but the amount of segments is larger. This is a different result to the one presented in Section 4.2.1 where the spectrum for the optimal polishing is always higher than the one for Heu(2), if the amount of segments is larger for optimal polishing than for Heu(2). The result of this simulation shows the advantage of the optimal polishing compared to the heuristic in the case that the cached video consists of regions with varying quality levels. This advantage becomes even more obvious if one imagines cached video objects which consist of more than two regions with different quality levels. The polishing effect (i.e. reducing the amount of layer variations) would probably only occur in the region with the highest quality level, while the remaining regions would remain unpolished.

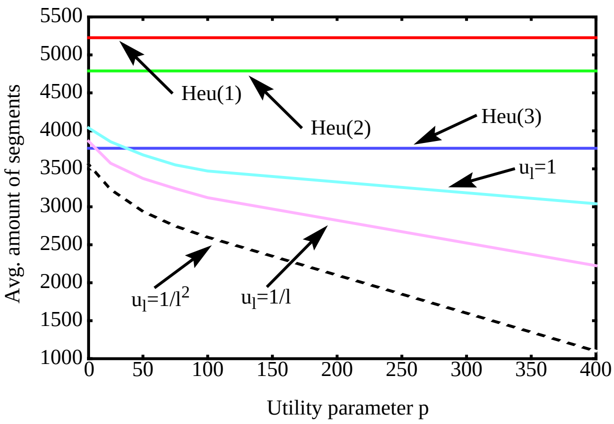


Figure 13: Avg. amount of segments per video object

5 POLISHING FOR POPULARITY-BASED CACHE REPLACEMENT

5.1 Optimization Model

Next we investigate whether optimal polishing can be applied to polish cached videos based on their popularity. That means, less segments are deleted from popular video objects while the amount of deleted segments increases for less popular objects. Thus, the quality of the cached object is directly related to its popularity. In contrast to the aforementioned approaches polishing is not performed on individual videos, but the complete content of the cache is regarded and polished according to the popularity of each single object and the amount of space that should be freed. This problem can be, similar to the problem presented in Section 4, formulated as mixed integer problem as shown in Figure 14.

We maximize the total utility of all videos weighted by their popularity. Compared to the model presented in Section 4 two additional parameters are introduced. The popularity of each video object w_v and the total capacity of the cache's storage that the already cached objects can consume K^{max}. The latter allows to determine how much storage space should be freed on the cache to allow the caching of new video objects.

5.2 Simulation Results

For the simulation, we specified the amount of cache space that should become available for the caching of new data $K^{total} - K^{max}$. In addition, each video object is assigned a certain popularity w_v. Figure 15 shows the originally cached and the resulting polished video object for the video with the highest (a) and the lowest (b) popularity on the cache. In this case, ten video objects are stored on the cache and 25% of the total cache space is freed by polishing the cached videos according to their popularity. For the case of the most popular video object 20% of the original segments are deleted, while for the least popular file 31% of the original segments are removed from the cache.

This simulation was performed 20 times. For each single simulation the initial cache state was randomly generated. Table 2 shows the amount of segments (in percent) that were removed from each of the ten cached video objects. Objects which are shaded equally were assigned the same popularity value. The popularity is the highest for objects 1, 2, and 3 while it is the lowest for objects 7, 8, 9, and 10. The popularity for 4, 5, and 6 lies in between the other two groups. The results of this simulation show that with the extended polishing algorithm a very fine granular cache replacement can be achieved. With this algorithm, it is possible to free cache space for new content while data from already cached content is

Indices:

$v = 1, \ldots, V$ -number of video object

$t = 1, \ldots, T$ -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 - sufficiently large number ($H \geq L$)

K^{max} maximum capacity available for already cached video objects

w_v - popularity of the video object

u_l - utility of receiving layer l for one second (if video is played back for 1 sec. on layer 3 it generates a utility 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 video is played back in period t at layer l or higher, zero otherwise

Optimization problem:

$$Max \sum_v \left(\sum_l \sum_t w_l u_{lt} b_{vtl} - \sum_t p z_{vt} \right) \tag{9}$$

subject to

$$h_{vt} - h_{vt-1} \leq H z_{vt} \qquad \forall t = 2, \ldots, T \ \forall v = 1, \ldots, V \tag{10}$$

$$h_{vt-1} - h_{vt} \leq H z_{vt} \qquad \forall t = 2, \ldots, T \ \forall v = 1, \ldots, V \tag{11}$$

$$0 \leq h_{vt} \leq h_{vt}^{cached} \qquad \forall t = 1, \ldots, T \ \forall v = 1, \ldots, V \tag{12}$$

$$l b_{vtl} \leq h_{vt} \qquad \forall t = 1, \ldots, T \ \forall v = 1, \ldots, V \ \forall l = 1, \ldots, L \tag{13}$$

$$b_{vtl} \in \{0, 1\} \qquad \forall t = 1, \ldots, T \ \forall v = 1, \ldots, V \ \forall l = 1, \ldots, L \tag{14}$$

$$z_{vt} \in \{0, 1\} \qquad \forall t = 2, \ldots, T \ \forall v = 1, \ldots, V \tag{15}$$

$$\sum_v \sum_t b_t h_{vt} \leq K^{max} \qquad \forall t = 1, \ldots, T \ \forall v = 1, \ldots, V \tag{16}$$

Figure 14: Cache replacement optimization model

removed according to its popularity. In this specific example 25% of the caches' storage space is available for the caching of new content while none of the cached objects had to be removed completely.

6 SUMMARY

In this paper, a new technique called polishing is presented which can be either applied for the streaming of data from the cache to the client or as a cache replacement mechanism. Polishing makes use of the fact that a reduction in layer variations can be achieved by not transmitting certain segments, although this means that some of the data available at the cache is omitted from the client. Optimal polishing maximizes the playback utility of a video, it is modeled as a mixed integer programming problem in this paper. Since optimal polishing is dependent from several parameters a simulative

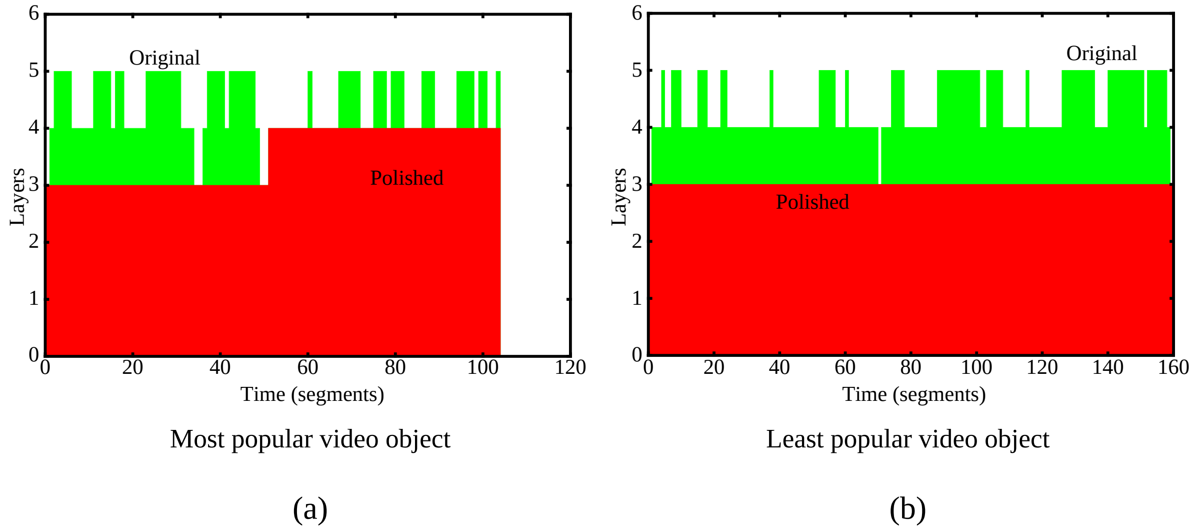


Figure 15: Polishing for most and least popular video object

Video Object	1	2	3	4	5	6	7	8	9	10
Average amount of removed segments in %	18	14	18	26	26	25	32	31	32	31
Avg. spectrum of polished object	5.6	4.3	5.3	3.1	4.0	5.4	2.2	2.0	2.8	2.4
Avg. spectrum of unpolished object	18.1	19.3	19.9	23.3	32.2	26.7	23.8	21.3	23.9	25.8

Table 2: Average amount of removed segments

investigation is performed to gain better insights in the influence of these parameters. In this simulation, the results of optimal polishing are compared with a simple heuristic that drops certain layers in complete. The results show that the heuristic can, in specific cases, achieves similar results as optimal polishing with less computational effort. Yet, simulations with layer-encoded video that consists of different quality regions demonstrate the drawback of the heuristic.

In addition, we showed how polishing can be used for cache replacement. The optimal polishing mixed integer problem is extended to vary the intensity of polishing based on the popularity of the video object and to operate on a set of videos instead of a single one. The results of this simulation show that by applying the extended optimal polishing storage space on the cache can be freed while the amount of segments is reduced according to the popularity of the single objects. To abstract, polishing is a good way to increase the quality of a streamed video, if retransmission from the cache cannot be performed and a good cache replacement mechanism.

7 REFERENCES

[1] M. Zink, J. Schmitt, and R. Steinmetz. Retransmission Scheduling in Layered Video Caches. In *Proceedings of the International Conference on Communications 2002 (ICC'02), New York, NY, USA*, pages 2474–2478, April 2002. ISBN 0-7803-7401-0.

[2] M. Zink, O. Künzel, J. B. Schmitt, and R. Steinmetz. Subjective Impression of Variations in Layer Encoded Videos. In *Proceedings of the 11th IEEE/IFIP International Workshop on Quality of Service (IWQoS'03), Monterey, CA, USA*, pages 134–154, June 2003. ISBN 3-540-40281-0.

[3] 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 (NOSSDAV'01), Port Jefferson, New York, USA*, pages 3–10, June 2001.

[4] M. Zink, J. Schmitt, and R. Steinmetz. Layer Encoded Video in Scalable Adaptive Streaming. *IEEE Transactions on Multimedia (accepted for publication)*.

[5] 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.

[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. ISBN 3-486-23987-2.

[10] Ilog Cplex: Mathematical Programming Optimizer. http://www.ilog.com/products/cplex/.

[11] M. Zink, C. Griwodz, J. Schmitt, and R. Steinmetz. Exploiting the Fair Share to Smoothly Transport Layered Encoded Video into Proxy Caches. In *Proceedings of SPIE/ACM Conference on Multimedia Computing and Networking (MMCN), San Jose, CA, USA*, pages 61–72. SPIE, January 2002. ISBN 0-8194-4413-8.