Slice Groups for Post-Compression Region of Interest Encryption in H.264/AVC and Its Scalable Extension

Andreas Unterweger, Andreas Uhl

University of Salzburg
Department of Computer Sciences
Jakob-Haringer-Straße 2
Salzburg, Austria

Abstract

Encrypting regions of interest in H.264/AVC and SVC bit streams after compression is a challenging task due to drift. In this paper, we assess whether the use of slice groups makes this task easier and what its expense in terms of bit rate overhead is. We introduce the concept of all-grey base layers for SVC which simplify the encryption of regions of interest in surveillance camera applications while obeying all standard-imposed base layer restrictions. Furthermore, we show that the use of slice groups is possible with relatively low overhead for medium and high bit rates (below 5% in most of the tested configurations). This applies to H.264/AVC as well as SVC bit streams with two and three spatial layers, including those with the newly introduced all-grey base layers. Although we are able to contain spatial and inter-layer drift with our proposed encryption setup, temporal drift still remains an issue that cannot be solved by sole usage of slice groups.

Keywords: H.264/AVC, SVC, Slice groups, Selective encryption, Region of Interest, Overhead

1. Introduction

In video surveillance and other applications, there is often the need to disguise people's identities in order to protect their privacy. A common approach to achieve this is the selective encryption of people's faces (also called region-based selective [1] or Region of Interest (RoI) encryption), i.e. encrypting all picture areas which contain a face, while leaving all other picture areas untouched.
This allows for reversible de-identification, i.e., the disguise of identities with the possibility to restore them by undoing the encryption. Restoring is typically only possible with a correct key which is possessed, e.g., by law enforcement authorities in case suspects of a crime need to be identified. Although several techniques for reversible de-identification exist, RoI encryption is one of the most common ones in video surveillance.
While RoI encryption can be applied before (e.g., [2, 3, 4]), during (e.g., [5, 6, 7]) or after compression (e.g., [8, 9, 10]), each with its own advantages and disadvantages [11], most approaches proposed so far focus either on encryption before or during compression. Although this makes drift, i.e., the propagation of parts of encrypted picture areas into non-encrypted ones through spatial and temporal prediction, easier to manage, it does not allow using existing surveillance infrastructure whose input images and/or encoder cannot be modified.
Typically, surveillance cameras have compression hardware built in (as of 2014, Motion JPEG and H.264/AVC are very common) which reduces the bandwidth of the captured and transmitted video footage. Although this saves time and computational resources by not requiring additional encoding hardware, it makes modifications (like additional encryption) to the built-in compression hardware nearly impossible due to the often hard-wired encoder.
In order to be able to reuse this infrastructure notwithstanding, applying RoI encryption after compression has to be considered, reviving the drift issue. Therefore, in this paper, we try to assess the fitness of the slice group coding tool of H.264/AVC [12] and its scalable extension [13], also known as Scalable Video Coding (SVC), to allow selective encryption of picture areas and to contain drift.
For the sake of applicability, we consider a state-of-the-art video surveillance system which delivers H.264/AVC-compressed output. We assume that the surveillance system detects faces (or other regions of interest) using a built-in face detector. This is common in most state-of-the-art surveillance systems. Since the coordinates of the detected faces are available this way, we further assume that the surveillance camera places the detected faces in slice groups. The definition of slice group borders based on the detected RoI does not require any special additional coding standards to be implemented since it is supported

Email addresses: aunterweg@cosy.sbg.ac.at (Andreas Unterweger), uhl@cosy.sbg.ac.at (Andreas Uhl)

by both, H.264/AVC and SVC. Even if face detection functionality is not yet in place in a surveillance system, it can simply be added without requiring major system modifications, e.g., by using an additional black box implementation of a face detector, which does not affect the rest of system.
The main reason for using slice groups is their ability to contain drift to a certain extent, thereby simplifying RoI encryption. Note that slice groups have other uses as well, thereby extending the results of our investigations to scenarios which are not encryption-specific.
By evaluating the limitations and possibilites of slice group coding, we aim at determining whether or not the aforementioned setup simplifies the encryption process in terms of drift. Furthermore, we evaluate the overhead induced by this approach in order to determine whether or not it is of practical use, i.e., for example to be included into existing and/or future surveillance systems to simplify RoI encryption after compression.
So far, no practical post-compression RoI encryption approaches have been proposed for H.264/AVC. [14], which uses one RoI per slice to contain spatial drift, encrypts at bit stream level, but does not evaluate the slice-induced overhead. In addition, only "regular" slice shapes (without Flexible Macroblock Ordering (FMO), i.e., in macroblock scan order from top-left to bottom-right) have been evaluated. This is not adequate for the use case considered in our paper, which requires rectangular RoI.
Although [10] describes a simple encryption approach for MPEG-4 Part 2 which could be extended to H.264/AVC, it is of no practical use since it reencodes the bit stream using intra blocks to avoid drift, which makes it actually an in-compression encryption approach. Although this may be suitable in terms of (transcoding) complexity for the video surveillance use case, the overhead is too large. As reported by [5] who applied the scheme described in [10] to H.264/AVC, overheads exceed 100% is some cases, depending on the complexity of the transcoding operation. Full transcoding with prediction restriction decreases the overhead to about 1-6% [15, 5], but is undesirable due to its high complexity.
Related work on RoI encryption in SVC is sparse. Two approaches are proposed in [16] and [17], albeit without considering or compensating for the effects of drift, which is an important matter. [7] deals with drift by imposing restrictions on the encoding process in terms of a limited motion estimation range as well as interpolation and upsampling constraints. Besides the reported significant increase in bit rate, this method cannot be applied on a bit stream level without recompression. Similarly, [18] proposes separate RoI coding by restricting motion estimation and inter-layer prediction, albeit without the explicit intention to do so for the sake of encryption. However, all of these approaches are in-compression encryption methods and cannot be applied at bit stream level.
Apart from RoI-related experiments and analyses of SVC, the encryption of certain Network Abstraction Layer (NAL) units has been proposed in [19]. However, their proposed encryption approach yields bit streams which are no longer format compliant and can hence not be decoded anymore by a regular decoder. This is not desirable in surveillance applications as the background without the encrypted RoI should be visible and therefore decodable.
Furthermore, the extraction and quality optimization of RoI across multiple layers to lower the total bit rate has been analyzed in [20]. However, the paper mainly focusses on cropping RoI through slice data removal and modification. It is not at all encryption-related and does therefore not take drift into account.
Although slice groups have been used to deal with drift in a number of encryption approaches (e.g., [5, 15]), a detailed examination of its actual usefulness to contain different causes of drift has not been done so far. The overhead induced by some of the aforementioned encryption approaches has been analyzed, but this is not true for the general overhead introduced by slices groups which change from frame to frame to cover RoI. This is especially true for SVC.
A number of analyses on slice groups for H.264/AVC, including overhead measurements for moving RoI, have been performed in [21]. However, they do not actually encode the RoI completely independently, as opposed to our implementation. Thus, their implementation provides an approximation, but no exact slice-group-related results, which are presented in this paper.
This paper is structured as follows: In section 2, the key concepts of video coding with slice groups in H.264/AVC and SVC are described, followed by an analysis of their limitations in section 3. After evaluating several scenarios in terms of feasibility for video surveillance with encrypted RoI in section 4, we conclude our paper.
This paper extends our previous work [22] by slice group overhead results for (non-scalable) H.264/AVC bit streams as well as a dissection of the overhead components. Furthermore, a detailled analysis of drift for both, H.264/AVC and SVC is provided and an additional post-compression approach is proposed and evaluated to circumvent standard-imposed restrictions. In addition, more sequences of actual surveillance footage are used.

2. H.264/AVC and SVC

The H.264 video coding standard allows for efficient compression of moving pictures by exploiting spatial and temporal redundancy. As a detailed description of H.264/AVC's features (as presented in [23]) is not within the scope of this paper, only the coding tools required herein are explained briefly.
In H.264/AVC-compliant bit streams, each coded picture is split into one or more slices, each of which consists of macroblocks of $16 \cdot 16$ luma samples and the corresponding chroma samples. Slices can be summarized to slice groups of specific forms (this is also known as FMO), depending on the so-called slice group map type. As RoI encryption

requires a background left-over, i.e., a region of the picture which does not belong to any encrypted region of interest, only slice group map types 2 (foreground slice groups with left-over background) and 6 (explicit slice group specification) will be considered, as only they allow this. Since slice group map type 6 is practically identical to slice group map type 2 in this use case, we will only consider slice group map type 2 henceforth.

To exploit spatial and temporal redundancy, H.264/AVC allows predicting samples of macroblocks from blocks around the one to be predicted in the same picture as well as from arbitrary blocks in previously coded pictures. In the former case, predictions over slice borders are forbidden, thereby allowing all slices to be decoded independently.

The scalable extension of H.264/AVC specified in its Annex G, also referred to as SVC, allows for multiple so-called layers within one bit stream, which can be accessed or extracted depending on the capabilities of the device decoding the stream. Each layer differs from the others either by frame rate (temporal scalability), resolution (spatial scalability) or quality (Signal-to-Noise Ratio (SNR) scalability). The bottom-most layer is referred to as base layer and coded in a way that is compatible with (non-scalable) H.264/AVC.

All layers but the base layer can exploit inter-layer redundancies by using coded information of lower layers for prediction. The basis of this prediction for spatial and SNR scalability can either be filtered intra-coded samples (inter-layer intra prediction), motion vectors (inter-layer motion prediction) or inter-coded difference signal samples (inter-layer residual prediction), with details for each prediction type to be found in [24]. In contrast, temporal scalability is achieved through hierarchical inter prediction as explained in detail in [13].

Figure 1 shows an example of a scalable bit stream with multiple layers. The base layer (temporal layer 0 (T0), spatial layer 0 (S0) and SNR layer 0 (Q0)) has the lowest possible frame rate, resolution and quality and is used to predict the first spatial enhancement layer (T0, S1, Q0; not labeled) which doubles both, picture width and height. This enhancement layer is further used to predict an enhancement layer of the same resolution, but a doubled frame rate (T1, S1, Q0) as well as an enhancement layer with higher quality (T0, S1, Q1; not labeled) and subsequently a doubled frame rate (T1, S1, Q1).

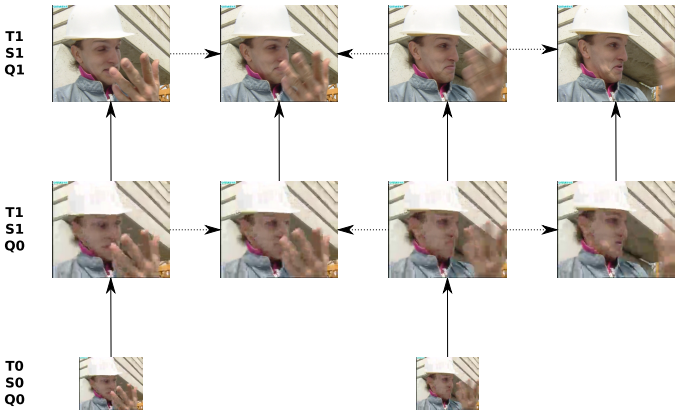

Figure 1: SVC with multiple layers: The base layer with half the frame rate and a quarter of the picture size can be used to predict the first spatial enhancement layer, which itself can be used to predict a second temporal and subsequently a third SNR enhancement layer. Adopted from [13]

3. Standard-imposed limitations

The H.264/AVC standard imposes restrictions on coding tools and parameter values by specifying profiles. As this paper discusses slice groups, we only consider profiles which allow the use of multiple slice groups in the first place. In this section, we investigate other relevant limitations imposed by those profiles.

For regular, i.e., non-scalable, H.264/AVC bit streams, only the Baseline and the Extended profile support slice groups. Although both allow using up to eight slice groups in total, one slice group is considered to be the background, i.e., the remainder of what the other seven slice groups encode.

Both, the Baseline and the Extended profile limit the available coding tools, most notably in that they only allow CAVLC entropy coding instead of CABAC. Moreover, the Baseline profile does not allow the use of B slices, i.e., only I and P slices can be used, as opposed to the Extended profile. Note that the lack of B slices is not a problem in surveillance scenarios where real-time transmission is expected, which would be delayed by the use of B frames [25]. The remaining profile constraints do not limit the use case described in this paper significantly and are therefore not described in detail.

For scalable bit streams, only the Scalable Baseline profile supports slice groups. Similar to the H.264/AVC Baseline profile, entropy coding is limited to CAVLC, the number of slice groups cannot exceed seven (plus background) and B slices are not allowed. Furthermore, the base layer may not contain more than one slice group.

This is a severe limitation in an encryption scenario because this means that the regions of interest cannot be in separate slice groups in the base layer. Thus, either a different drift compensation approach for the base layer is required or an alternative to slice groups in the base layer has to be found. As the former is hard to achieve, we consider three additional alternatives to slice groups in the base layer as depicted in Figure 2.

One possibility is to use extended spatial scalability, depicted on the left and in the middle of Figure 2, where the base layer only contains the region of interest and the enhancement layer adds the rest of the video frame. Due to the limitations of the Scalable Baseline profile, the width and height ratios between the base layer and the corresponding region of interest in the enhancement layer have to be either 1 (Figure 2, left), 1.5 (not depicted) or 2 (Fig-

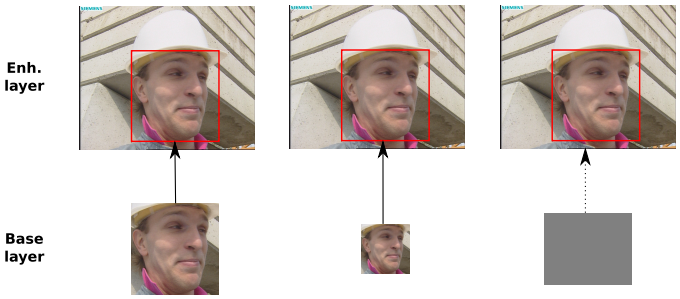

Figure 2: Alternatives to slice groups in the base layer: Left and middle: Extended spatial scalability; right: all-grey base layer

ure 2, middle).
However, this setup is only useful if there is exactly one region of interest. Since this would impose a severe practical limitation, it is not considered in the remainder of this paper. Alternatively, we propose adding a base layer which is all-grey ($Y = C_b = C_r = 128$) as shown in Figure 2, right. Since intra DC prediction and skip modes allow encoding such an artificial layer very compactly, its overhead is relatively small when using the maximum possible width and height ratios of 2, i.e., a base layer with half the width and height of the enhancement layer.
However, it effectively reduces the number of usable spatial layers, which is limited to three in the Scalable Baseline profile, by one. This allows for a maximum of two non-grey spatial layers for actual video content. Depending on the use case, these two remaining layers may be sufficient to provide spatial scalability.
Since the standard-imposed restrictions prevent encryption methods from easily encrypting the base layer (there are no slice groups allowed in order to contain the drift), the base layer would have to be treated separately for encryption in all practical scenarios, entailing different restrictions and drawbacks. There are two possibilities to put the grey base layer in place: a true post-compression approach and a constrained post-compression approach.
In the true post-compression approach, the input bit stream has a regular base layer. During encryption, it is replaced by a grey base layer at bit stream level, as shown in figure 3. If no inter-layer prediction is used (this reduces rate-distortion performance by about 1-2 dB [26]), no reencoding is necessary. The original base layer is irrecoverably lost in this case, which in-turn is expected to increase the rate-distortion performance. This allows for post-compression encryption at the cost of losing the original base layer.
Conversely, in the constrained post-compression approach, the grey base layer has to be put in place by the encoder. This can simply be done by using an all-grey image instead of a downscaled version of the corresponding high resolution image. It constrains the supported bit streams since a grey base layer is already required to be in the input file. This only allows for post-compression encryption if the encoder hardware can be configured to support a

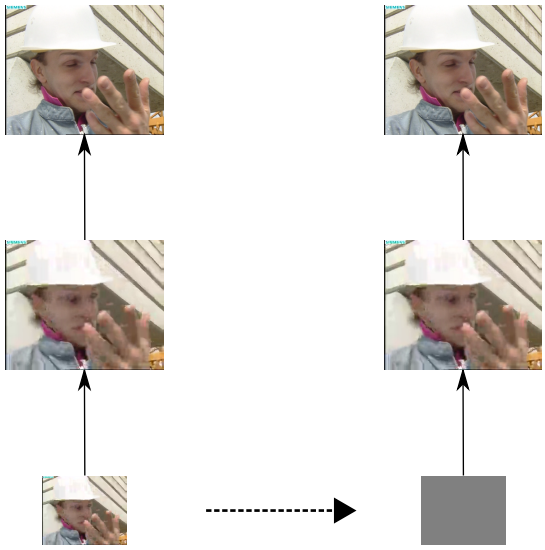

Figure 3: True post-compression encryption: An existing base layer is replaced by an all-grey base layer to circumvent standard-imposed restrictions for slice groups in the base layer

grey base layer. We consider both, the constrained and the true post-compression approaches in this paper and analyze their differences in detail in section 4.
Although there have been multiple proposals for region-of-interest support through slice groups in all layers [27, 28], the final version of the standard does not allow this. Similarly, the technique proposed in [29] to alternatively support regions of interests as enhancement layers is not supported. This paper limits the available options to the ones supported by the standard, i.e., the all-grey base layer introduced above as well a regular (i.e., full-content) base layer for comparison.
When the base layer is encrypted completely, for example, it is not usable by a decoder which only extracts and displays the base layer, yet a standard decoder would not be aware of this when receiving the bit stream. When using a grey base layer, as proposed, the situation is similar: A standard decoder only shows a grey picture, which is still format compliant. However, the overhead of using a grey base layer is expected to be significantly lower as compared to a completely encrypted base layer, which requires a separate encryption approach and additional drift prevention mechanisms in order to avoid inter-layer drift.
Despite the loss of one usable spatial layer, the grey base layer simplifies encryption by containing drift. Although the unavailability of slice groups in the base layer (see above) would normally make encryption harder (without the possibility of using slice groups to contain drift), the fact that the base layer is all grey does not require any encryption and does therefore not induce any drift.
Regarding further limitations imposed by the standard, we will focus on the combination of constrained intra prediction and constrained inter-layer prediction, which ensure single-loop decoding [30]. Since these two limitations severely limit the number of possibilities for prediction and

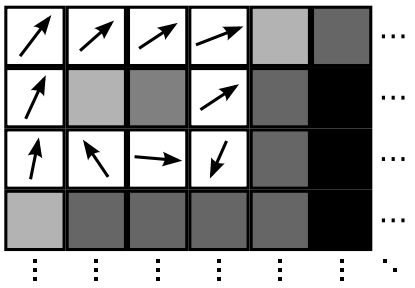

Figure 4: Constrained intra prediction: In a P slice, intra blocks may not use inter blocks for prediction. The grey level of the depicted intra blocks denotes the number of allowed intra modes

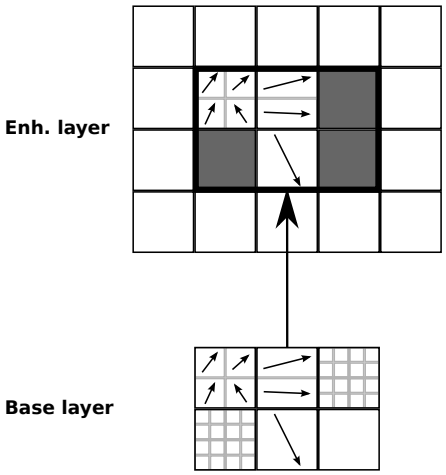

Figure 5: Constrained inter-layer prediction: Upsampled intra blocks (grey) must be reconstructed from base layer intra samples

therefore drift, they are crucial for the RoI encryption use case.

Constrained intra prediction limits the blocks which can be used for intra prediction. Figure 4 illustrates this in a P slice which contains inter (depicted by motion vectors) and intra (depicted by grey levels) macroblocks. Although the black intra blocks may use all possible intra prediction modes, the dark- and light-grey ones may not. For example, the light-grey macroblock at the top left may only use DC prediction since all other prediction directions would require predicting from one of the surrounding inter macroblocks. Note that constrained intra prediction reduces coding efficiency, especially for isolated intra macroblocks, i.e., intra macroblocks surrounded by inter macroblocks.

SVC enforces constrained intra prediction in all layers which are used for inter-layer prediction so that inter-layer predicted samples do not require additional motion compensation in the base layer. Additionally, constrained inter-layer prediction ensures that inter-layer-predicted intra samples are not used for intra prediction themselves, as illustrated in Figure 5.

Inter-layer prediction allows using information from the base layer in the enhancement layer. If blocks are upsampled through inter-layer intra prediction (grey blocks in Figure 5), the corresponding reference block in the base

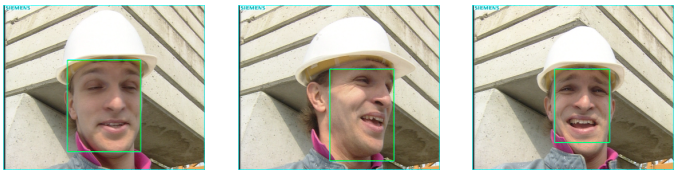

Figure 6: Moving slice groups: Frames 1, 11 and 21 of the *foreman* sequence with one moving foreground slice group around the face (green) and one background slice group (remainder, turquois)

Figure 7: Encrypted RoI: Frames 1, 11 and 21 of the *foreman* sequence. The RoI in this example is the actor's face. Note that the noise is symbolic to illustrate the combination of moving RoI and an arbitrary form of encryption

layer has to be an intra block as well. Constrained intra prediction in the base layer ensures that no additional motion compensation loop is required. Furthermore, if the enhancement layer is used for further inter layer prediction, the upsampled blocks may not be used for further intra prediction due to the constrained intra prediction requirement to avoid multi-loop decoding.

Note that constrained inter prediction [31] has also been proposed, but not incorporated into the final video coding standards. With constrained inter prediction, inter macroblocks must not depend on intra macroblocks from the same slice. This allows minimizing the dependencies between intra and inter data partitions when data partitioning is used. This is useful when the intra data (partition) is lost – the inter data can still be used.

4. Experimental evaluation

In this section, we describe our experimental setup and results. We refer to the term of "moving slice groups" for RoI herein since the position of RoI may change from frame to frame, thereby changing the slice group positions accordingly, as illustrated in Figure 6.

Recall that our use case is encryption, i.e., we assume that the moving slice groups will be encrypted at some point, as illustrated by example in figure 7. Note, however, that we do not propose a specific encryption algorithm – our results are independent of the employed encryption approach as long as the latter is format compliant. The noise in figure 7 is therefore only symbolic.

4.1. Setup

In order to evaluate the effect of slice-group-based RoI for encryption, we added support for moving slice groups to both, the H.264/AVC (*JM*) and SVC (*JSVM*) reference software, since they do not support this by

themselves.
Although the *JM* supports slice group coding in principle, it only does so with one set of coordinates for all frames. Therefore, in our modification, before encoding each frame, the corresponding RoI coordinates are loaded and all data structures containing the slice group information are adapted accordingly. Since the slice groups' position and size are signaled by a Picture Parameter Set (PPS) preceding the corresponding picture, the *ResendPPS* parameter is enabled so that one PPS is inserted before each frame. Note that the PPS data structure needs to be modified as well, albeit before the PPS is written to the output.
In the *JSVM*, slice group coding is implemented partially, but not used. Therefore, it is enabled separately for all spatial layers but the base layer which does not support slice group coding (see section 3). In addition, in each layer, the RoI coordinates are calculated depending on the picture size and the corresponding slice group settings are adapted accordingly. In order to signal the slice groups, one additional PPS per frame and enhancement layer is required. In contrast to the *JM* with its *ResendPPS* parameter, this requires inserting one PPS per frame per enhancement layer by modifying the source code accordingly.
We use a total of six test sequences depicted in figure 8: three common test video sequences (*akiyo*, *foreman* and *crew*, each 300 frames long and in Common Intermediate Format (CIF) resolution) as well as three surveillance video sequences where the camera that captured them is static and people move by (*hall* with 300 frames in CIF resolution, *ice* with 240 frames in 4CIF resolution and *visor_1246522137645_new_4_camera2* (abbreviated *visor* henceforth) from the VISOR data set (`http://www.openvisor.org/video_details.asp?idvideo=323`) with 1019 frames in Quarter Video Graphics Array (QVGA) resolution). All video sequences have 30 frames per second, except the *visor* sequence, which has only 10 frames per second. In addition, the *visor* sequence was converted from the Red Green Blue (RGB) to the YCbCr color space with 4:2:0 subsampling using *ffmpeg*.
The three common video sequences differ in terms of face count and motion, representing both, typical and extreme cases for evaluation. *akiyo* has one RoI and very little motion, while *foreman* has a significant amount of motion. Both have only one RoI. Conversely, the *crew* sequence has a significant amount of motion and a changing number (between 2 and 11) of RoI.
The three surveillance video sequences have no global motion, as mentioned above. *hall* has little local motion and between no and 2 RoI. Conversely, *ice* has a significant amount of motion and between 2 and 7 RoI. In contrast, *visor* has jerky motion due to the low frame rate and no RoI most of the time. The short time intervals in which there are RoI visible, there are between 1 and 7 RoI.
All faces were segmented manually by enclosing them in rectangles. The corresponding coordinates were rounded to the nearest macroblock border. Since a maximum of seven slice groups (RoI) are supported in both, H.264/AVC and SVC (see section 3), only the first top-left-most faces are considered, i.e., placed in a separate slice group. This only affects the *crew* sequence, which has more than seven RoI, but does not impact our results. Since we do not actually encrypt the RoI, but only assess the overhead induced by slice groups, the smaller RoI will give an upper bound of the overhead for actual implementations which will likely combine some of the RoI to reduce the number of slice groups to seven.

Figure 8: Video sequences used for testing (from top-left to bottom-right): Frame 100 of *foreman*, *akiyo*, *crew*, *hall*, *ice* and *visor*

4.2. Overhead (H.264/AVC)

In the case of H.264/AVC, we distinguish various typical Group Of Pictures (GOP) structures: $I*$ (i.e., only I frames), $I(PPP)*$ (i.e., one I frame, followed by groups of three P frames), $I(bP)*$ (i.e., one I frame, followed by groups with one non-reference B and one P frame each) and $I(BBBBBBBP)*$ (i.e., one I frame, followed by groups of seven B frames and one P frame each, where the B frames are coded hierarchically). Note that GOP structures with B frames require the use of the Extended profile (see section 3).
We encode the test sequences with a constant Quantization

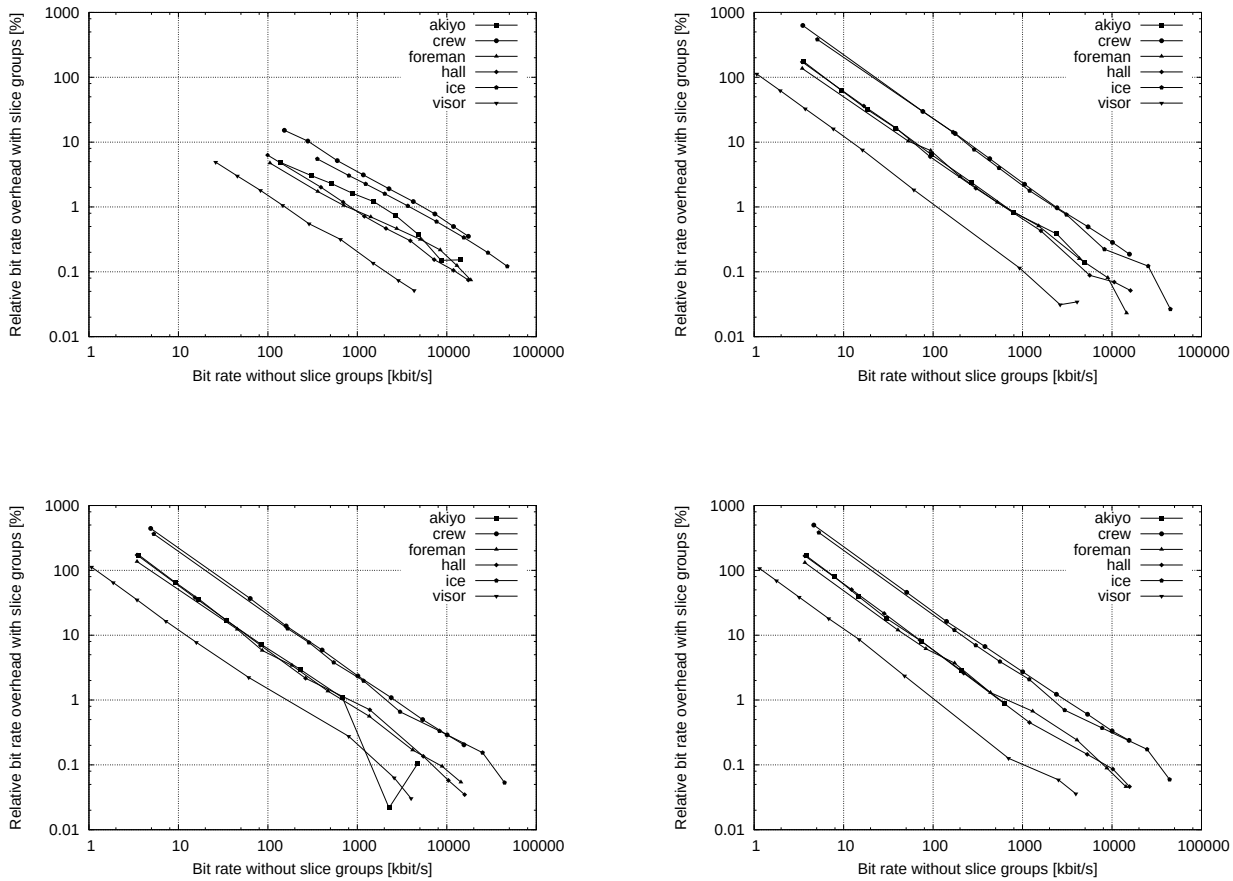

Figure 9: Overhead with slice group coding for different GOP structures: $I*$ (top-left), $I(PPP)*$ (top-right), $I(bP)*$ (bottom-left), $I(BBBBBBBP)*$ (bottom-right)

Parameter (QP) for all frame types and default settings. Using QPs between 3 and 51 with a step size of 6 to double the quantizer step size with each run allows covering the whole QP range (since the results of this paper may be useful for other applications as well, we did deliberately not restrict the QP range to typical surveillance video settings). Each QP-sequence combination is encoded with and without slice groups. Since the difference in terms of distortion between the encoded sequences with and without slice groups is very small (< 0.1 dB), we approximate the overhead introduced by slice group coding by comparing the corresponding bit rates directly.

Figure 9 shows the overhead for the different GOP structures and sequences. In order to make comparisons between the overheads of different GOP structures easier, figure 10 depicts the overhead of the *crew* sequence in detail for all tested GOP structures.

It is obvious that the *crew* and the *ice* sequence (depicted by circles and pentagons in figure 9, respectively) exhibit the highest overhead in nearly all scenarios, since they require the highest number of slice groups. Conversely, the *visor* sequence exhibits the lowest overhead, since it only

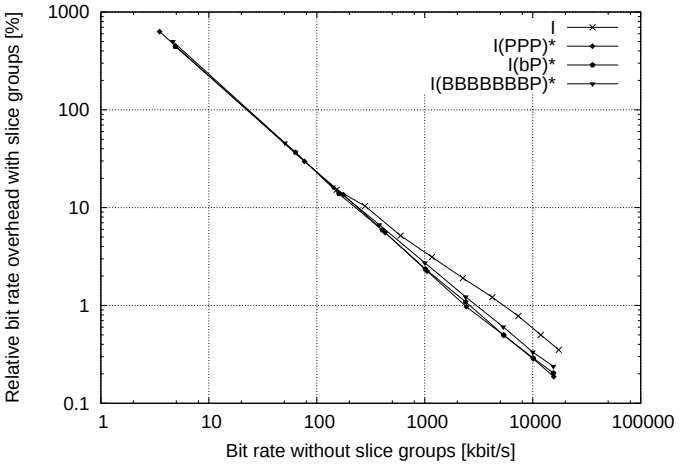

Figure 10: Slice group overhead comparison for the *crew* sequence with different GOP structures

requires slice groups for some short parts of the sequence and has a lower frame rate. The *akiyo*, *foreman* and *hall* sequences require about the same number of slice groups in total, so their overhead lies between the two corner cases with nearly none and nearly always the maximum number of slice groups.

Unsurprisingly, the $I*$ GOP structure (depicted by crosses in figure 10) exhibits the lowest overhead percentages. As it uses no inter prediction at all, the lowest possible bit rate is relatively high (see figure 9, top-left). Although this overhead difference compared to other GOP structures is notable for very high bit rates, it becomes insignificantly small for bit rates above 1000 kbit/s (see figure 10). The other GOP structures behave very similarly in terms of relative overhead, thus making the GOP structure choice practically irrelevant for low to medium bit rates. For high bit rates, it is irrelevant as long as the GOP does not consist of I frames only.

In general, the overhead for all GOP structures decreases with the bit rate, i.e., it increases with the QP. For low bit rates, slice group coding adds an unacceptable overhead of up to several hundred per cent. Conversely, for bit rates which are higher than 1000 kbit/s, all sequences but *crew* and *ice* exhibit a small overhead of approximately 1% or less.

The overhead of the *crew* sequence is approximately five times higher than the overhead of the other sequences over a large QP range, i.e., for nearly all bit rates. This is due to the use of the maximum number of slice groups in nearly all frames throughout the sequence and shows that the number of slice groups significantly influences the bit rate overhead.

For very high bit rates, i.e., very low QP, the overhead of the *akiyo* sequence with slice groups fluctuates, resulting in non-depicted data points for some bit rates due to the corresponding very small negative values which cannot be depicted using logarithmic axes. The fluctuations are due to the fact that *akiyo* requires a relatively low bit rate compared to the other sequences. Thus, in the high bit rate range, the slice group borders which prevent intra prediction only affect the number of quantized non-zero coefficients minimally, so that the overhead becomes very low. Depending on the actual coefficients, this impacts further intra prediction, making the very small overhead a nearly random value due to the high impact of the very small changes in the coefficients.

Note that this prediction-border-related overhead is only one part of the total overhead. The overheads depicted above can be split into two components: Firstly, there is a constant overhead for the additional PPS which are required to signal the position and size of the slice groups for each frame. Secondly, the additional prediction borders induced by the slice groups decrease coding efficiency, resulting in an overhead when using a constant QP.

Table 1 shows the first component and the absolute total overhead for the *crew* sequence for the $I(BBBBBBBP)*$ GOP structure (since the GOP structure does not impact the overhead significantly, as shown above, this can be considered to be representative for this sequence). Without slice groups, there is only one PPS of 9 bytes required. Conversely, when slice groups are used, 7657 bytes are required for all 300 PPS – one per frame, with different sizes each, depending on the number of RoI.

QP	File size diff.	Relative PPS size diff.
3	46249	16.53%
9	41751	18.31%
15	40170	19.03%
21	36613	20.88%
27	34221	22.34%
33	31716	24.11%
39	28773	26.58%
45	28995	26.37%
51	28980	26.39%

Table 1: Absolute overhead in bytes for the *crew* sequence with $I(BBBBBBBP)*$ GOP structure. The rightmost column denotes the relative amount of PPS bytes of the corresponding total absolute overhead

Although the number of PPS bytes required for signalling remains constant ($7657 - 9 = 7648$ bytes), their relative amount increases with increasing QP. Most notably, for very low QP, i.e., very high quality, it only accounts for less than a fifth of the total absolute overhead. The remainder of the overhead is, as described above, due the second overhead component, i.e., the slice-group induced prediction borders. It can be seen that the PPS-related constant overhead does not exceed 27% of the total absolute overhead.

4.3. Overhead (SVC)

In the case of SVC, we use the GOP size of the default JSVM configuration, i.e., four. Since GOP structures with B frames are not allowed in combination with slice groups (see section 3), we use P frames instead. Thus, an $(IPPP)*$ GOP structure, i.e., a repeated sequence of one I frame and followed by three P frames, is used.

We encode the test sequences with a constant QP for both frame types and default settings with two and three dyadic spatial layers. The base layer is all grey (see section 3), although we test "classical" base layers (with the actual down-sized input video) as well for comparison. Inter-layer prediction is set to adaptive to allow for optimal coding efficiency.

In this section, we consider the constrained post-compression approach, in which the grey base layer is already put into place by the encoder, as described in section 3. An analysis of the differences between this approach and the true post-compression approach is provided in section 4.4.

Note that we use 4CIF versions of *crew* and *foreman* for these measurements since CIF sequences with three spatial layers would yield impractically small base layers. Since we were unable to obtain a 4CIF version of *akiyo*, we omit-

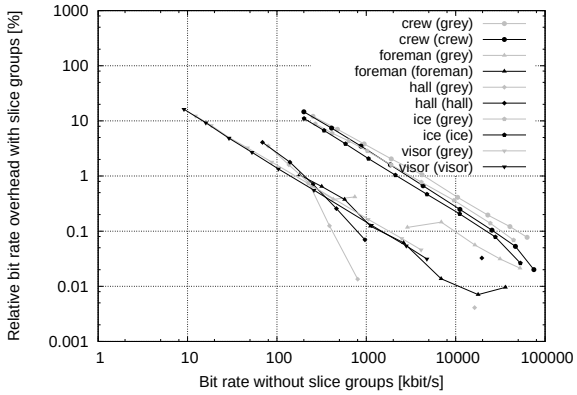

Figure 11: Overhead with slice group coding for different sequences when using two dyadic spatial layers. The names in parentheses denote the used base layer sequence.

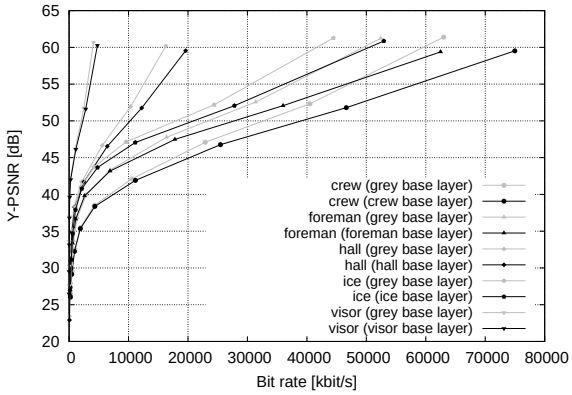

Figure 12: Rate-distortion plot for SVC with two dyadic spatial layers and slice groups. Different base layers (depicted in grey and black) result in significantly different enhancement layer Y-PSNR.

ted it from this test set. Note, however, that we kept the *visor* sequence in the test set due to its relevance as the only low-frame-rate surveillance video.

As in the H.264/AVC experiments (see section 4.2), each QP-sequence combination is encoded with and without slice groups. Again, the difference in terms of distortion between the encoded sequences with and without slice groups is very small (< 0.15 dB), so we approximate the overhead introduced by slice group coding by comparing the corresponding bit rates directly.

As depicted in Figure 11, in the case of two spatial layers, the overhead shows a similar dependency on the bit rate as in the H.264/AVC case (see section 4.2). While very low bit rates result in infeasibly large overhead, medium and high bit rates exhibit moderate to low overhead.

The *crew* and *ice* sequences exhibit the highest overhead when using slice groups due to the large number of RoI, as in the H.264/AVC case (see section 4.2). The *foreman* and *hall* sequences profit from scalability more than the other sequences, resulting in very small negative overhead values ($< 0.1\%$ absolute). Note that these values cannot be depicted properly due to the logarithmic Y axis.

Using an all-grey base layer does not affect the overhead significantly due to the use of slice groups. Compared to the classical base layer configuration, however, an all-grey base layer allows using slice-group-based encryption for SVC in the first place, since slice groups cannot be used in the base layer (see section 3).

Figure 12 shows a rate-distortion plot for the two-layer case with slice groups, where the Y-PSNR values are those of the enhancement layer. The plot allows comparing the all-grey base layer with a classical base layer. It is obvious that the all-grey base layer results in significantly better rate-distortion performance (up to 5 dB) for medium and high bit rates.

Since an all-grey base layer greatly improves rate-distortion performance avoiding the need for additional drift compensation due to encryption in the base layer, it can be considered a better solution than a classical base layer for this use case. As the overhead due to slice groups is similar in both, the all-grey and the classical base layer scenario (see above), this is also true for other potential use cases in which the base layer does not have to be the downsampled input sequence.

Note that an all-grey base layer in a scenario with two spatial layers defies the purpose of scalable video coding, since one of the two layers becomes unusable for content. However, it allows establishing a baseline for comparison in terms of overhead and allows assessing the usefulness of the concept. In order for all-grey base layers to be practically useful, a scenario with three spatial layers has to be considered so that two spatial layers remain for actual content.

When increasing the number of spatial layers to the maximum of three (see section 3), the overhead due to slice groups increases, as depicted in Figure 13. The overall overhead is significantly higher than in the two-layer case (see Figure 11) for low to medium bit rates. This is due to the fact that slice groups introduce prediction borders which reduce coding efficiency and the three-layer case (with two enhancement layers with slice groups) uses double the amount of slice groups than the two-layer case (with one enhancement layer with slice groups). However, for high bit rates, the overhead is still relatively small and therefore practically negligible for most use cases.

Compared to the two-layer case, the all-grey base layer configuration in the three-layer case allows for an overhead which is approximately as low as the overhead in the classical base layer configuration. Although the all-grey base layer configuration exhibits a higher overhead for medium-to-high bit rates, the actual overhead is only insignificantly higher.

However, in the three-layer case the rate-distortion performance improvement of the all-grey base layer is only very

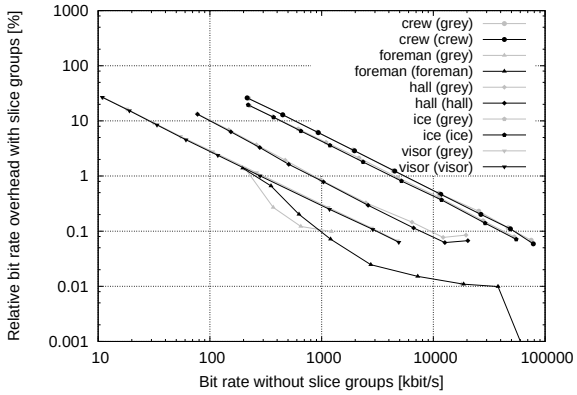

Figure 13: Overhead with slice group coding for different sequences when using three dyadic spatial layers. The names in parentheses denote the used base layer sequence.

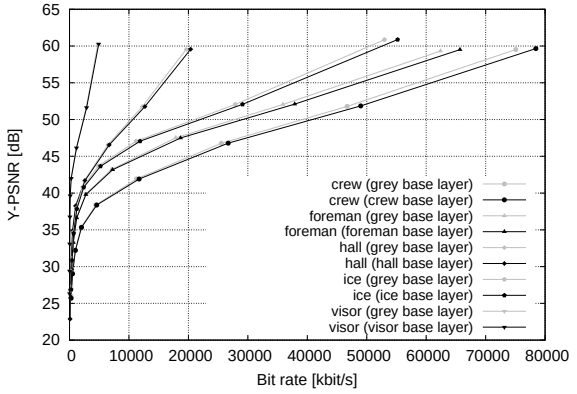

Figure 14: Rate-distortion plot for SVC with three dyadic spatial layers and slice groups. Different base layers (depicted in grey and black) result in similar enhancement layer Y-PSNR.

small, as depicted in Figure 14. Although there are still differences of up to 1 dB between an all-grey and a classical base layer in terms of enhancement layer Y-PSNR, the performance improvement is nowhere near the improvements of the two-layer case (see above).
This is mainly due to the fact that there are two enhancement layers, which use most of the bit rate and the fact that the first enhancement layer can be used to predict parts of the second one through inter-layer prediction. This makes the three-layer case with an all-grey base layer similar to a two-layer case with an additional all-grey bit stream, which is very likely not used at all for inter-layer prediction. However, an all-grey base layer still has advantages compared to a classical base layer for the use case in this paper, since base layer encryption cannot rely on slice groups due to base layer limitations (see above). Thus, an all-grey base layer is still to be preferred over a classical base layer in the three-layer case.

4.4. True post-compression approach performance

Since the previous section dealt with the performance of the constrained post-compression approach, this section aims at highlighting the performance differences of the true post-compression approach, in which a regular base layer is replaced by a grey base layer during encryption.
As described in detail in section 3, the true post-compression approach requires a "regular" base layer in an SVC bit stream which does not use inter-layer prediction. During encryption, the original base layer is removed (which can be done safely since no inter-layer-prediction-related dependencies can yield drift) and replaced by an all-grey base layer.
Both, the constraint of not allowing inter-layer prediction and the replacement of the base layer, change the rate-distortion performance significantly. Although the base layer constraint is known to result in a decrease of about 1-2 dB [26], the use of an all-grey base layer has been shown to increase rate-distortion performance significantly when using two spatial layers with inter-layer prediction in section 4.3.
Thus, it is necessary to evaluate the overall change in rate-distortion performance in this section. We do this by evaluating several differently coded versions of the *crew* sequence in 4CIF resolution with the same basic encoding parameters as in section 4.3.
Figure 15 shows the results for two dyadic spatial layers. The imposed constraint (no inter-layer prediction) on the base layer (dotted black line) decreases rate-distortion performance by about 1 dB, as expected, compared to SVC with inter-layer prediction (solid black line). However, the replacement of the base layer by an all-grey base layer (grey line) increases the performance significantly, yielding even higher Y-PSNR values than SVC with inter-layer prediction. The difference is small for low bit rates, but reaches up to 5 dB for very high bit rates.
Conversely, figure 16 shows the results for three dyadic spatial layers, where the differences between the different configurations practically vanish for most bit rates. Even though SVC with inter-layer prediction is slightly superior to the grey base layer without inter-layer prediction for the true post-compression approach, the difference is only about 0.5 dB.
Note that in both, figure 15 and 16, the performance of the true post-compression approach (grey line) is equal to the performance of the constrained post-compression approach described in section 4.3. In summary, both approaches outperform SVC with inter-layer prediction in terms of rate-distortion performance when using two spatial layers and are only marginally inferior when using three spatial layers. This makes them adequate alternatives which simplify encryption at the expense of one lost, i.e., grey, spatial layer. It also justifies the restriction to disallow inter-layer prediction in the base layer for the true post-compression approach.

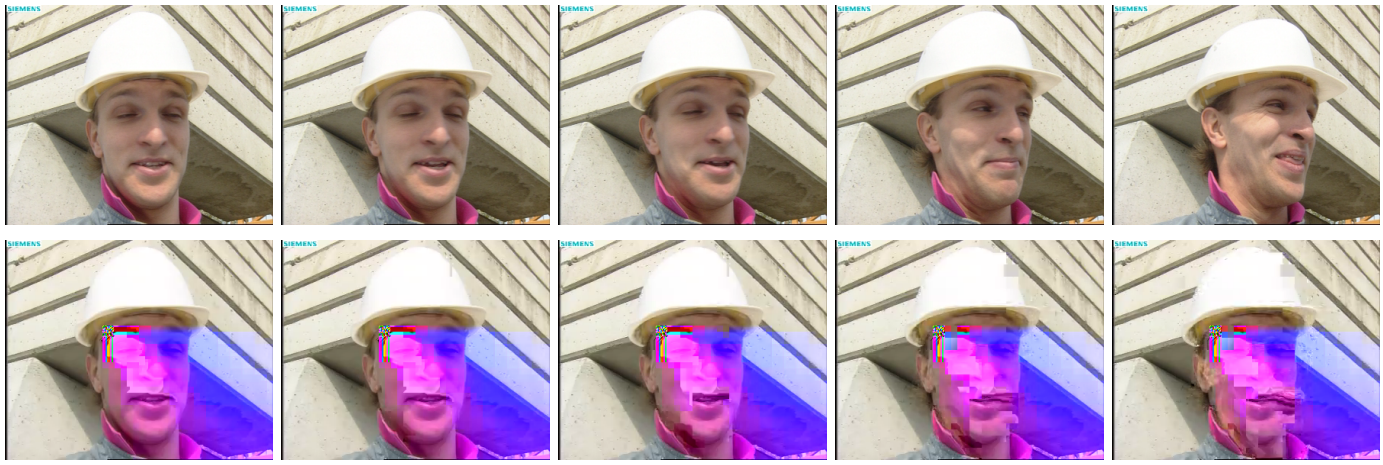

Figure 18: Example for temporal drift: The first, second, third, fifth and tenth frame (from left to right) of the *foreman* sequence where one block in the first frame (left-most) has been modified (as in figure 17). The top row shows the original frames, whereas the bottom row shows the frames with temporal drift (second from the left to right-most).

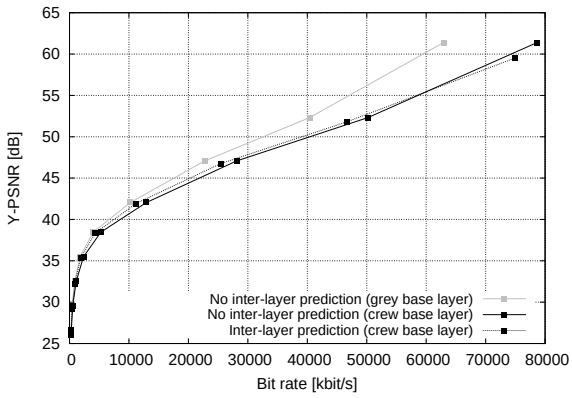

Figure 15: Rate-distortion plot for SVC with two dyadic spatial layers and slice groups with different coding configurations to illustrate the performance differences of the true post-compression approach.

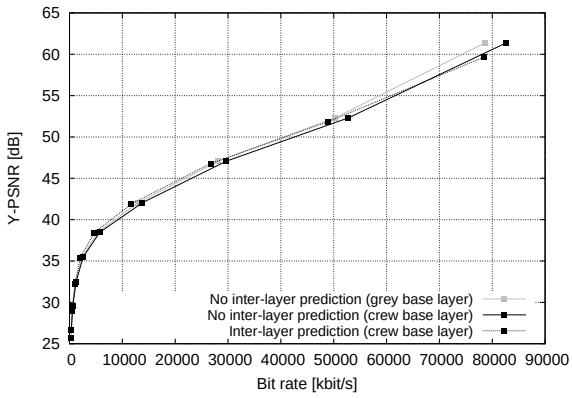

Figure 16: Rate-distortion plot for SVC with three dyadic spatial layers and slice groups with different coding configurations to illustrate the performance differences of the true post-compression approach.

Figure 17: Example for spatial drift due to one changed macroblock in the first frame of the *foreman* sequence: original frame (left) versus modified frame with drift (right).

4.5. Drift

In H.264/AVC, two types of drift can occur – spatial and temporal drift due to intra and inter prediction, respectively. Due to the interdependency of macroblocks through the respective forms of prediction, changes to one macroblock influence one or more other macroblocks. Figures 17 and 18 illustrate this by example. In SVC, inter-layer drift, i.e., the propagation of errors between a base and an enhancement layer, may occur in addition.

Slice groups are able to contain spatial drift as they form a prediction border for intra prediction. This simplifies the encryption of slice groups in an I frame since the blocks outside the RoI, i.e., those which are contained in the slice group which forms the background, cannot use encrypted data for prediction.

For SVC, this applies to the co-located I frames in the higher layers as well, since each block in them is either coded using intra prediction or inter-layer intra prediction, which must use co-located base layer intra samples (see section 3). In addition, the slice group coordinates and size scale along with the spatial layer in our use case, which keeps encoded data inside the RoI due to the co-location property. Thus, encoding each RoI as one slice group prevents spatial drift and inter-layer drift for all I

frames.
However, slice groups are not able to contain temporal drift, since motion vectors may cross slice group boundaries. In H.264/AVC, this means that P and B frames are very likely to exhibit drift. Within each inter frame itself, however, slice groups contain spatial drift due to imposed intra prediction border as well as the limitation of motion vector predictors.
This is also true for P frames in SVC. However, in higher layers, inter-layer drift may occur. Although constrained intra prediction limits the spatial propagation of inter-layer-predicted samples, inter-layer motion and residual prediction may upscale drift-induced errors from the base layer which have been created through temporal drift. This means that inter-layer drift in this use case is only a consequence of temporal drift and can be prevented, if temporal drift can be eliminated.
In summary, when using slice groups in the use case described herein, spatial drift is contained in I, P and B frames for H.264/AVC, and in I and P frames for SVC. However, temporal drift cannot be contained in P and B frames for H.264/AVC, and P frames for SVC. Inter-layer drift in SVC can be contained in I frames in the described use case due to the co-location of slice groups in all layers, but cannot be contained in P frames as it is a consequence of temporal drift.

5. Future work

Although this paper shows that slice groups help containing drift in H.264/AVC and SVC, post-compression encryption approaches which make use of this have yet to be developed. Since the problem of temporal drift remains, this is a challenging task and remains future work.
In addition, the detailed effects of SNR scalability have to be studied. Although SNR scalability can be considered as a special case of spatial scalability where width and height remain the same, the overhead of slice groups in SNR layers may be significantly lower due to the more restricted inter-layer prediction mechanisms. This would make SVC encryption yet more feasible, since SNR layers are identical to spatial layers in terms of drift as analyzed in this paper.

6. Conclusion

We showed the impact of slice group coding on post-compression encryption for a typical surveillance use case. We analyzed the slice-group-induced bit rate overhead as well as the usefulness of slice groups for the containment of drift. For medium and high bit rates, H.264/AVC as well as SVC configurations with two and three layers can be used to reduce drift with slice groups with relatively low overhead. In contrast, for low bit rates, the overhead is too large for practical use. Furthermore, we introduced the concept of all-grey base layers which simplifies encryption significantly in the two- and three-layer case of SVC, albeit at the cost of losing one spatial scalability layer. Finally, we showed that the containment of drift in SVC can be reduced to the containment of temporal drift in H.264/AVC for this surveillance use case.

7. Acknowledgments

This work is supported by FFG Bridge project 832082.

References

[1] Y. Ou, C. Sur, K. H. Rhee, Region-based selective encryption for medical imaging, in: Proceedings of the International Conference on Frontiers in Algorithmics (FAW'07), Lecture Notes in Computer Science, Springer-Verlag, Lanzhou, China, 2007, pp. 62–73.

[2] T. E. Boult, PICO: Privacy through invertible cryptographic obscuration, in: IEEE/NFS Workshop on Computer Vision for Interactive and Intelligent Environments, Lexington, KY, USA, 2005, pp. 27–38.

[3] P. Carrillo, H. Kalva, S. Magliveras, Compression Independent Reversible Encryption for Privacy in Video Surveillance, EURASIP Journal on Information Security 2009 (2009) 1–13.

[4] F. Dufaux, T. Ebrahimi, A framework for the validation of privacy protection solutions in video surveillance, in: Proceedings of the IEEE International Conference on Multimedia & Expo, ICME '10, IEEE, Singapore, 2010, pp. 66–71.

[5] L. Tong, F. Dai, Y. Zhang, J. Li, Prediction restricted H.264/AVC video scrambling for privacy protection, Electronic Letters 46 (1) (2010) 47–49. doi:10.1049/el.2010.2068.

[6] Z. Shahid, M. Chaumont, W. Puech, Selective and scalable encryption of enhancement layers for dyadic scalable H.264/AVC by scrambling of scan patterns, in: 16th IEEE International Conference on Image Processing, Cairo, Egypt, 2009, pp. 1273–1276.

[7] Y. Kim, S. Yin, T. Bae, Y. Ro, A selective video encryption for the region of interest in scalable video coding, in: Proceedings of the TENCON 2007 - IEEE Region 10 Conference, Taipei, Taiwan, 2007, pp. 1–4.

[8] T.-L. Wu, S. F. Wu, Selective encryption and watermarking of MPEG video (extended abstract), in: H. R. Arabnia (Ed.), Proceedings of the International Conference on Image Science, Systems, and Technology, CISST '97, Las Vegas, USA, 1997.

[9] F. Dufaux, T. Ebrahimi, Video surveillance using JPEG 2000, in: Proceedings of the SPIE Applications of Digital Image Processing XXVII, Vol. 5588, 2004, pp. 268–275.

[10] F. Dufaux, T. Ebrahimi, Scrambling for privacy protection in video surveillance systems, IEEE Transactions on Circuits and Systems for Video Technology 18 (8) (2008) 1168–1174. doi:10.1109/TCSVT.2008.928225.

[11] A. Massoudi, F. Lefèbvre, C. D. Vleeschouwer, B. Macq, J.-J. Quisquater, Overview on selective encryption of image and video, challenges and perspectives, EURASIP Journal on Information Security 2008 (Article ID 179290) (2008) doi:10.1155/2008/179290, 18 pages.

[12] ITU-T H.264, Advanced video coding for generic audiovisual services, `http://www.itu.int/rec/T-REC-H.264-200711-I/en` (Nov. 2007).

[13] H. Schwarz, D. Marpe, T. Wiegand, Overview of the scalable H.264/MPEG4-AVC extension, in: Proceedings of the IEEE International Conference on Image Processing, ICIP '06, IEEE, Atlanta, GA, USA, 2006, pp. 161–164.

[14] R. Iqbal, S. Shirmohammadi, A. E. Saddik, J. Zhao, Compressed-domain video processing for adaptation, encryption, and authentication, IEEE Multimedia 15 (2) (2008) 38–50.

[15] F. Dufaux, T. Ebrahimi, H.264/AVC video scrambling for privacy protection, in: Proceedings of the IEEE International Conference on Image Processing, ICIP '08, IEEE, San Diego, CA, USA, 2008, pp. 47–49.

[16] H. Sohn, E. Anzaku, W. D. Neve, Y. M. Ro, K. Plataniotis, Privacy protection in video surveillance systems using scalable video coding, in: Proceedings of the Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, Genova, Italy, 2009, pp. 424–429.

[17] Y. Kim, S. Jin, Y. Ro, Scalable Security and Conditional Access Control for Multiple Regions of Interest in Scalable Video Coding, in: Y. Shi, H.-J. Kim, S. Katzenbeisser (Eds.), International Workshop on Digital Watermarking 2007 (IWDW 2007), Vol. 5041, Springer Berlin / Heidelberg, 2008, pp. 71–86.

[18] S. S. F. Shah, E. A. Edirisinghe, Evolving Roi Coding in H.264 SVC, in: VISAPP 2008: Proceedings of the Third International Conference on Computer Vision Theory and Applications – Volume 1, 2008, pp. 13–19.

[19] C. Li, X. Zhou, Y. Zhong, NAL level encryption for scalable video coding, in: Advances in Multimedia Information Processing, PCM'08, Springer-Verlag, 2008, pp. 496–505. doi:10.1007/978-3-540-89796-5.

[20] D. Grois, E. Kaminsky, O. Hadar, Roi adaptive scalable video coding for limited bandwidth wireless networks, in: 2010 IFIP Wireless Days (WD), 2010, pp. 1–5.

[21] Y. Dhondt, S. Mys, K. Vermeirsch, R. Van de Walle, Constrained Inter Prediction: Removing Dependencies Between Different Data Partitions, in: J. Blanc-Talon, W. Philips, D. Popescu, P. Scheunders (Eds.), Advanced Concepts for Intelligent Vision Systems, Vol. 4678 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2007, pp. 720–731.

[22] A. Unterweger, A. Uhl, Slice Groups for Post-Compression Region of Interest Encryption in SVC, in: IH&MMSec'14: Proceedings of the 2014 ACM Information Hiding and Multimedia Security Workshop, ACM, Salzburg, Austria, 2014, pp. 15–22.

[23] T. Wiegand, G. J. Sullivan, G. Bjontegaard, A. Luthra, Overview of the H.264/AVC video coding standard, IEEE Transactions on Circuits and Systems for Video Technology 13 (7) (2003) 560–576.

[24] C. A. Segall, G. J. Sullivan, Spatial scalability within the H.264/AVC scalable video coding extension, IEEE Transactions on Circuits and Systems for Video Technology 17 (9) (2007) 1121–1135. doi:10.1109/TCSVT.2007.906824.

[25] A. Leontaris, P. Cosman, Compression Efficiency and Delay Tradeoffs for Hierarchical B-Pictures and Pulsed-Quality Frames, IEEE Transactions on Image Processing 16 (7) (2007) 1726–1740.

[26] H. Schwarz, D. Marpe, T. Wiegand, Overview of the scalable video coding extension of the H.264/AVC standard, IEEE Transactions on Circuits and Systems for Video Technology 17 (9) (2007) 1103–1120. doi:10.1109/TCSVT.2007.905532.

[27] T. M. Bae, T. C. Thang, D. Y. Kim, J. W. K. Yong Man Ro, J. G. Kim, Multiple Region-of-Interest Support in Scalable Video Coding, ETRI Journal 28 (2) (2006) 239–242.

[28] T. C. Thang, T. M. Bae, Y. J. Jung, Y. M. Ro, J.-G. Kim, H. Choi, J.-W. Hong, Spatial Scalability of Multiple ROIs in Surveillance Video, http://wftp3.itu.int/av-arch/jvt-site/2005_04_Busan/JVT-O037.doc (Jan. 2005).

[29] J.-H. Lee, C. Yoo, Scalable ROI algorithm for H.264/SVC-based video streaming, in: 2011 IEEE International Conference on Consumer Electronics (ICCE), 2011, pp. 201–202.

[30] H. Schwarz, T. Hinz, D. Marpe, T. Wiegand, Constrained inter-layer prediction for single-loop decoding in spatial scalability, in: IEEE International Conference on Image Processing (ICIP) 2005, Vol. 2, 2005, pp. II–870–873.

[31] P. Lambert, W. D. Neve, Y. Dhondt, R. V. de Walle, Flexible macroblock ordering in H.264/AVC, Journal of Visual Communication and Image Representation 17 (2) (2006) 358–375.