8 Screen Content Coding for HEVC

Contents

8 Screen Contents Coding for HEVC .......................................................... 1
8.1 Introduction to SCC ........................................................................... 3
8.2 Screen Content coding tools ............................................................... 6
  8.2.1 Intra Block Copy ........................................................................ 6
  8.2.2 Palette mode .............................................................................. 8
    8.2.2.1 Palette derivation ................................................................. 9
    8.2.2.2 Coding the palette entries .................................................. 10
    8.2.2.3 Coding the palette indices .................................................. 10
  8.2.3 Adaptive color transform (ACT) .................................................. 11
    8.2.3.1 Color space conversion ..................................................... 12
    8.2.3.2 Encoder optimization ......................................................... 13
  8.2.4 Adaptive motion vector resolution .............................................. 13
8.3 Lossless and visually lossless coding algorithms ............................... 14
  8.3.1 Residual DPCM .......................................................................... 14
  8.3.2 Sample-based weighted prediction with directional template matching .. 16
  8.3.3 Sample-based angular intra-prediction ....................................... 17
  8.3.4 Sample-based angular intra-prediction with edge prediction ............ 18
8.4 Fast coding algorithms ....................................................................... 19
  8.4.1 Adaptive motion compensation precision ................................... 19
  8.4.2 Fast intra coding ........................................................................ 20
8.5 Visual quality assessment ................................................................... 22
  8.5.1 Screen image quality assessment .............................................. 22
  8.5.2 Objective quality assessment ..................................................... 23
  8.5.3 Subjective quality assessment ..................................................... 24
8.6 Other SCC algorithms ...................................................................... 25
  8.6.1 Segmentation ............................................................................. 25
  8.6.2 Rate control .............................................................................. 26
Acronyms

APSIPA  Asia-Pacific Signal and Information Processing Association Annual Summit and Conference
ACT  Adaptive Color Transform
BV  Block Vector
BMSB  Broadband Multimedia Systems and Broadcasting
BVP  Block vector prediction
CBF  Coded block flag
CfP  Call for Proposals
DiOR  Digital operating room
DSCQS  Double Stimulus Continuous Quality Scale
DSIS  Double Stimulus Impairment Scale
DIQA  Document image quality assessment
IBC  Intra Block Copy
LSC  Layer segmentation based coding
NIQA  Natural image quality assessment
P2SM  Pseudo 2D String Matching
PCoIP  PC over IP
PCS  Picture Coding Symposium
SC  Stimulus Comparison, Screen content
SCM  Screen Content Coding Test Model
SCADA  Supervisory control and data acquisition
SDM  Structural Degradation Model
SDSCE  Simultaneous Double Stimulus Continuous Evaluation
SIP  Simple Intra Prediction
SIQA  Screen image quality assessment
SIQM  Structure Induced Quality Metric
SPEC  Shape primitive extraction and coding
SS  Single Stimulus
SSCQE  Single Stimulus Continuous Quality Evaluation
SSIM  Structural Similarity
SWP  Sample-based weighted prediction
TSF  Transform skip flag
TSM  Transform skip mode
VDI  Virtual desktop infrastructure
WSIs  Whole slide images
ABSTRACT

Screen content (SC) video is generated by computer program and displayed on screen without any signal noise. The SC picture usually contains a lot of discontinuous textures and edges. Patch of SC picture is often identical in many regions in the picture. Based on these differences from camera-captured video, it has been developed as an extension of HEVC. We discussed, here, the four screen content coding tools, lossless and lossy coding performance, fast algorithms, quality assessment, and other algorithms developed recently.

Keywords

Adaptive color transform, Adaptive motion vector resolution, Intra block copy, Palette coding, Residual DPCM, Sample-based prediction, Screen contents, Screen image quality assessment, String matching, Template matching, Transform skip

8.1 Introduction to SCC

High efficiency video coding (HEVC) discussed in Chapter xxx is the latest video compression standard of the Joint Collaborative Team on Video Coding (JCT-VC), which was established by the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG) [1]. It demonstrates a substantial bit-rate reduction over the existing H.264/AVC standard [2]. Several extensions and profiles of HEVC have been developed according to application areas and objects to be coded. However, both the HEVC and the H.264/AVC focused on compressing camera-captured video sequences, mainly consisting of human objects and complex textures (Figure 8.1). Although several different types of test sequences were used during the development of these standards, the camera-captured sequences exhibited common characteristics such as the presence of sensor noise and an abundance of translational motion. Thus, video compression exploits both temporal and spatial redundancies. A frame which is compressed by exploiting the spatial redundancies is termed as intra frame and the frames which are compressed by exploiting the temporal redundancies are termed as inter frames. The compression of a inter frame requires a reference frame which will be used to exploit the temporal redundancies. The inter frame is usually of two types namely a P frame and B frame. The P frame make use of one already encoded/decoded frame which may appear before or after the current picture in the display order i.e. a past or a future frame as its reference,
whereas the B frame makes use of two already encoded/decoded frames, one of which is a past and the other being the future frame as its reference frames, thus, providing higher compression but also higher encoding time as it has to use a future frame for encoding [3]. Furthermore, conventional video coders, in general, remove high-frequency components for compression purposes, since the human visual sensitivity is not so high in high frequencies.

![Figure 8.1. Image example from camera captured video content [4].](image)

Recently, however, there has been a proliferation of applications that display more than just camera-captured content. These applications include displays that combine camera-captured and computer graphics, wireless displays, tablets as second display, control rooms with high resolution display wall, digital operating room (DiOR), virtual desktop infrastructure (VDI), screen/desktop sharing and collaboration, cloud computing and gaming, factory automation display, supervisory control and data acquisition (SCADA) display, automotive/navigation display, PC over IP (PCoIP), ultra-thin client, remote sensing, etc. [5], [6]. The type of video content used in these applications can contain a significant amount of stationary or moving computer graphics and text, along with camera-captured content, as shown in Figure 8.2. However, unlike camera-captured content, screen content frequently contains no sensor noise, and such content may have large uniformly flat areas, repeated patterns, highly saturated or a limited number of different colors, and numerically identical blocks or regions among a sequence of pictures. These characteristics, if properly managed, can offer opportunities for significant improvements in compression efficiency over a coding system designed primarily for camera-captured natural content. Unlike natural images/video, screen contents may not be very smooth. They usually have totally different statistics. For text or graphics contents, it is much sharper and with high contrast [7]. Because of the high contrast, any little artifact caused by removing high frequency components in conventional video coders may be perceived by users. Thus, all coding techniques supported by HEVC RExt and additional coding tools such as intra block coping, palette coding, and adaptive color space transform are required to compress screen content. Features of screen contents are summarized as:
- Sharp content: Screen content usually includes sharp edges, such as in graphic or animation content. To help encoding sharp content, transform skip has been designed for screen content [8].
- Large motion: For example, when browsing a web page, a large motion exists when scrolling the page. Thus, new motion estimation algorithms to handle the large motions for screen content may be required.
- Artificial motion: For example, when fading in or fading out, the conventional motion model may not be easy to handle it.
- Repeating patterns: For example, the compound images may contain the same letter or objects many times. To utilize the correlation among repeating patterns, Intra Block Copy (IBC) has been developed.

![Images of screen content: (a) slide editing, (b) animation, (c) video with text overlay, (d) mobile display](image)

Figure 8.2. Images of screen content: (a) slide editing, (b) animation, (c) video with text overlay, (d) mobile display [4].

Joint Call for Proposals (CfP) was released in Jan. 2014 with the target of developing extensions of the HEVC standard including specific tools for screen content coding [9]. The use cases and requirements of the CfP are described in [5] and common conditions for the proposals are found in [10]. These documents identified three types of screen content: mixed content, text and graphics with motion, and animation. Up to visually lossless coding performance was requested for RGB and YCbCr, 4:4:4 formats having 8 or 10 bits per color component. After seven responses to the CfP were evaluated at the JCT-VC meeting [11], several core experiments (CEs) were defined including intra block copying extensions, line-based intra copy, palette mode, string matching for sample coding, and cross-component prediction and adaptive color transforms. As results of evaluating the outcome of the CEs and related proposals, the HEVC Screen Content Coding Test Model 6 [12] and Draft Text 5 [13] were published in Oct. 2015. All the documents are available in [14]. Test sequences [15], reference software [16] and manual [17] are also available.
8.2 Screen Content coding tools

HEVC-SCC is based on the HEVC framework while several new modules/tools are added as shown in Figure 8.3, including intra block copy (IBC), palette coding, adaptive color transform, and adaptive motion vector resolution.

8.2.1 Intra Block Copy

HEVC-SCC introduces a new CU mode in addition to the conventional intra and inter modes, referred to as intra block copy (IBC). The IBC mode performs like an inter mode prediction but the prediction units (PU) of IBC coded coding units (CU) predict reconstructed blocks in the same picture, taking the advantage of exploiting the repeated patterns that may appear in screen content. Similar to inter mode, IBC uses block vectors to locate the predictor block [12].

Since screen content is likely to have similar or repeated patterns on a screen, such spatial redundancy can be removed by quite different way from the conventional intra prediction schemes. The most significant difference is the distance and shapes from neighboring objects [18]. Removable spatial redundancy in the conventional intra prediction schemes refers to the similarity between the boundary

---

Figure 8.3. Encoding block diagram for screen content coding [6] © IEEE 2016.
pixels of the block to be coded and the adjacent pixels located spatially within one pixel. However, the removable spatial redundancy in IBC mode refers to the similarity between the area in the reconstructed picture and the block to be coded. A target 2D block/object is predicted from a reconstructed 2D block/object that is more than one pixel distant from it using the motion or location information (motion vector) from the reference block/object.

![Figure 8.4. Intra block copy prediction in the current picture](image) © IEEE 2016.

Figure 8.4 shows the concept of IBC and the block vector (BV), which is conceptually similar to the motion vector (MV) of inter prediction. In terms of the accuracy of the vectors, MV has in usual quarter-pel accuracy to ensure improved prediction accuracy, whereas BV is enough to be in integer-pel accuracy. This is because of the characteristics of the screen content in IBC mode. For example, objects in computer graphics are generated pixel by pixel and repeated patterns are found in integer-pel accuracy. Block compensation, which is similar to motion compensation, of IBC is conducted on the reconstructed area of the current frame not previously coded, or decoded frames. In addition, the BV should be sent to the decoder side, but it is derived by prediction to reduce the amount of data in a manner similar to the motion vector. Prediction may be independent from the MV prediction method or in the same way as the MV prediction method.

The global BV search is performed for 8×8 and 16×16 blocks. Search area is a portion of the reconstructed current picture before loop filtering, as depicted in Figure 8.4. Additionally, when slices/tiles are used, the search area is further restricted to be within the current slice/tile. For 16×16 blocks, only a one-dimensional search is conducted over the entire picture. This means that the search is performed horizontally or vertically. For 8×8 blocks, a hash-based search is used to speed up the full picture search. The bit-length of the hash table entry is 16. Each node in the hash table records the position of each block vector candidate in the picture. With the hash table, only the block vector candidates having the same hash entry value as that of the current block are examined [19].

The hash entries for the current and reference blocks are calculated using the original pixel values.
The 16-bit hash entry $H$ is calculated as

$$H = \text{MSB}(\text{DC}_0, 3) \ll 13 + \text{MSB}(\text{DC}_1, 3) \ll 10 + \text{MSB}(\text{DC}_2, 3)$$

$$\ll 7 + \text{MSB}(\text{DC}_3, 3) \ll 4 + \text{MSB}(\text{Grad}, 4)$$  \hspace{1cm} (8.1)

where $\text{MSB}(X,n)$ represents the $n$ MSB of $X$, DC0, DC1, DC2, DC3 denote the DC values of the four $4\times4$ sub-blocks of the $8\times8$ block, and Grad denotes the gradient of the $8\times8$ block. Operator ‘$\ll$’ represents arithmetic left shift.

In addition to full block vector search, some fast search and early termination methods are employed in the HEVC-SCC. The fast IBC search is performed after evaluating the RD cost of inter mode, if the residual of inter prediction is not zero. The SAD-based RD costs of using a set of block vector predictors are calculated. The set includes the five spatial neighboring block vectors as utilized in inter merge mode (as shown in Figure 8.5) and the last two coded block vectors. In addition, the derived block vectors of the blocks pointed to by each of the aforementioned block vector predictors are also included. This fast search is performed before the evaluation of intra prediction mode. It is applied only to $2N\times2N$ partition of various CU sizes.

![Figure 8.5. Candidates of block vector predictor [20] © IEEE 2015.](image)

### 8.2.2 Palette mode

For screen content, it is observed that for many blocks, a limited number of distinct color values may exist. In this case, the set of color values is referred to as the palette. Thus, palette mode enumerates those color values and then for each sample, sends an index to indicate to which color it belongs. In special cases it is also possible to indicate a sample that is outside the palette by signalling an escape symbol followed by component values as illustrated in Figure 8.6. Palette mode can improve coding efficiency when the prediction does not work due to low redundancy and when the number of pixel values for the block is small [18] [21]. According to the results by Xiu, et al, [22], coding gain of the palette-based coding increases up to 9.0% in the average BD-rate for lossy coding and up to 6.1% for lossless coding mode.
8.2.2.1 Palette derivation

In the SCM-7.0 software [16], for the derivation of the palette for lossy coding, a modified k-means clustering algorithm is used. The first sample of the block is added to the palette. Then, for each subsequent sample from the block, the SAD from each of the current palette entries is calculated. If the distortion for each of the components is less than a threshold value for the palette entry corresponding to the minimum SAD, the sample is added to the cluster belonging to the palette entry. Otherwise, the sample is added as a new palette entry. When the number of samples mapped to a cluster exceeds a threshold, a centroid for that cluster is calculated and becomes the palette entry corresponding to that cluster.

In the next step, the clusters are sorted in a decreasing order of frequency. Then, the palette entry corresponding to each entry is updated. Normally, the cluster centroid is used as the palette entry. But a rate-distortion analysis is performed to analyze whether any entry from the palette predictor may be more suitable to be used as the updated palette entry instead of the centroid when the cost of coding the palette entries is taken into account. This process is continued till all the clusters are processed or the maximum palette size is reached. Finally, if a cluster has only a single sample and the corresponding palette entry is not in the palette predictor, the sample is converted to an escape symbol. Additionally, duplicate palette entries are removed and their clusters are merged.

For lossless coding, a different derivation process is used. A histogram of the samples in the CU is calculated. The histogram is sorted in a decreasing order of frequency. Then, starting with the most frequent histogram entry, each entry is added to the palette. Histogram entries that occur only once are converted to escape symbols if they are not a part of the palette predictor.

After palette derivation, each sample in the block is assigned the index of the nearest (in SAD) palette entry. Then, the samples are assigned to ‘INDEX’ or ‘COPY ABOVE’ mode. For each sample
for which either 'INDEX' or 'COPY_ABOVE' mode is possible, the run for each mode is determined. Then, the cost (in terms of average bits per sample position) of coding the mode, the run and possibly the index value (for 'INDEX' mode) is calculated. The mode for which the cost is lower is selected. The decision is greedy in the sense that future runs and their costs are not taken into account.

### 8.2.2.2 Coding the palette entries

For coding of the palette entries, a palette predictor is maintained. The maximum size of the palette as well as the palette predictor is signaled in the sequence parameter set (SPS). In SCM 4, a palette_predictor_initializer_present_flag is introduced in the PPS. When this flag is 1, entries for initializing the palette predictor are signaled in the bitstream. The palette predictor is initialized at the beginning of each CTU row, each slice and each tile. Depending on the value of the palette_predictor_initializer_present_flag, the palette predictor is reset to 0 or initialized using the palette predictor initializer entries signaled in the picture parameter set (PPS). In SCM 5, palette predictor initialization at the SPS level was introduced to save PPS bits when a number of PPS palette predictor initializers shared common entries. In SCM 6, a palette predictor initializer of size 0 was enabled to allow explicit disabling of the palette predictor initialization at the PPS level.

For each entry in the palette predictor, a reuse flag is signaled to indicate whether it is part of the current palette. This is illustrated in Figure 8.7. The reuse flags are sent using run-length coding of zeros. After this, the number of new palette entries is signaled using exponential Golomb code of order 0. Finally, the component values for the new palette entries are signaled.

![Figure 8.7. Use of palette predictor to signal palette entries](image)

### 8.2.2.3 Coding the palette indices

The palette indices are coded using three main palette sample modes: INDEX mode, COPY_ABOVE
mode, and ESCAPE mode as illustrated in Figure 8.8. In the INDEX mode, run-length coding is conducted to explicitly signal the color index value, and the mode index, color index, and run-length are coded. In the COPY_ABOVE mode, which copies the color index of the row above, the mode index and run-length are coded. Finally, in the ESCAPE mode, which uses the pixel value as it is, the mode index and the quantized pixel value are coded. When escape symbol is part of the run in 'INDEX' or 'COPY_ABOVE' mode, the escape component values are signalled for each escape symbol.

![Diagram](image.png)

Figure 8.8. Coding the palette indices [21] © IEEE 2014.

### 8.2.3 Adaptive color transform (ACT)

Conventional natural content is usually captured in RGB color format. Since there is strong correlation among different color components, a color space conversion is required to remove inter-component redundancy. However, for screen content, there may exist many image blocks containing different features having very saturated colors, which leads to less correlation among color components. For those blocks, coding directly in the RGB color space may be more effective. ACT enables the adaptive selection of color-space conversion for each block. To keep the complexity as low as possible, the color-space conversion process is applied to the residual signal as shown in Figure 8.9 and after the intra- or inter-prediction process, the prediction residuals are selected to perform forward color-space transform as shown in Figure 8.10.
8.2.3.1 Color space conversion

To handle different characteristics of image blocks in screen content, a RGB-to-YC_oC_g conversion [24] was investigated to use it for forward and backward lossy and lossless coding.

Forward transform for lossy coding (non-normative):

\[
\begin{bmatrix}
Y \\
C_o \\
C_g
\end{bmatrix} = \begin{bmatrix}
1 & 2 & 1 \\
2 & 0 & -2 \\
-1 & 2 & -1
\end{bmatrix} \begin{bmatrix}
R \\
G/4 \\
B
\end{bmatrix} \tag{8.2}
\]

Forward transform for lossless coding (non-normative):

\[
\begin{align*}
C_o & = R - B \\
t & = B + (C_o \gg 1) \\
C_g & = (G-t) \\
Y & = t + (C_g \gg 1)
\end{align*} \tag{8.3}
\]

Backward transform (normative):
8.2.3.2 Encoder optimization

In the ACT mode, encoder complexity increases double because the mode searching is performed in both the original color space and the converted color space. To avoid this, following fast methods are applied:

- For intra coding mode, the best luma and chroma modes are decided once and shared between the two color spaces.
- For IBC and inter modes, block vector search or motion estimation is performed only once. The block vectors and motion vectors are shared between the two color spaces.

8.2.4 Adaptive motion vector resolution

For natural video content, the motion vector of an object is not necessarily exactly aligned to the integer sample positions. Motion compensation is, therefore, not limited to using integer sample positions, i.e. fractional motion compensation is more efficient to increase compression ratio. Computer-generated screen content video, however, is often generated with knowledge of the sample positions, resulting in motion that is discrete or precisely aligned with sample positions in the picture. For this kind of video, integer motion vectors may be sufficient for representing the motion. Savings in bit-rate can be achieved by not signalling the fractional portion of the motion vectors.

Adaptive MV resolution allows the MVs of an entire picture to be signalled in either quarter-pel precision (same as HEVC version 1) or integer-pel precision. Hash based motion statistics are kept and checked in order to properly decide the appropriate MV resolution for the current picture without relying on multi-pass encoding. To decide the MV precision of one picture, blocks are classified into the following categories:

- C: number of blocks matching with collocated block
- S: number of blocks not matching with collocated block but belong to smooth region. For smooth region, it means every column has a single pixel value or every row has a single pixel value.
M: number of blocks not belonging to C or S but can find a matching block by hash value.

The MV resolution is determined as:
- If CSMRate < 0.8, use quarter-pel MV.
- Otherwise, if C == T, use integer-pel MV.
- Otherwise, if AverageCSMRate < 0.95, use quarter-pel MV.
- Otherwise, if M > (T−C−S)/3, use integer-pel MV.
- Otherwise, if CSMRate > 0.99 and MRate > 0.01, use integer-pel MV.
- Otherwise, if AverageCSMRate + AverageMRate > 1.01, use integer-pel MV.
- Otherwise, use quarter-pel MV.

T is the total number of blocks in one picture. CSMRate = (C+S+M)/T, MRate = M/T.
AverageCSMRate is the average CSMRate of current picture and the previous 31 pictures.
AverageMRate is the average MRate of the current picture and the previous 31 pictures.

8.3 Lossless and visually lossless coding algorithms

8.3.1 Residual DPCM

Differential pulse code modulation (DPCM) has been widely used to reduce spatial and temporal redundancy in the video content. The subtraction of the prediction signal from the current block signal generates the residual signal or the prediction error, containing the part of the original signal which could not be predicted by the selected predictor [25]. The residual signal can be further compressed by any method. In the HEVC, compression can be achieved by the application of a transformation, which is applied to represent the correlated parts of the residual signal in the residual block by a potentially small number of transform coefficients. These coefficients are then quantized and coded into the bitstream.

Sample-by-sample residual DPCM (RDPCM) of intra-predicted residuals was proposed [26] in the context of H.264/AVC lossless coding. When using this technique, instead of performing conventional intra-prediction each residual sample is predicted from neighboring residuals in the vertical or horizontal direction when the intra prediction is equal to one of these two directions. Average bitrate reductions of 12% were reported using this technique compared with conventional H.264/AVC lossless coding. This technique was later extended and adapted to the HEVC standard [27] achieving on average 8.4% bitrate reductions on screen content sequences.

The residuals of each sample are calculated by sample-by-sample DPCM in vertical and/or horizontal mode. When the intra prediction mode is vertical, the RDPCM elements \( \tilde{r}_{i,j} \) is given by
or when the intra prediction mode is horizontal mode

\[
\tilde{r}_{i,j} = \begin{cases} 
  r_{i,j} & , i = 0, 0 \leq j \leq (N-1) \\
  r_{i,j} - r_{(i-1),j} & , 1 \leq i \leq (M-1), 0 \leq j \leq (N-1)
\end{cases}
\] (8.5)

In the vertical mode, the samples in the first row in the block are left unchanged. All other samples are predicted from the sample immediately above in the same column. The horizontal intra RDPCM is given in a similar way.

The RDPCM elements are signaled to the decoder so that the original residual samples are reconstructed by

\[
r_{i,j} = \sum_{k=0}^{i} \tilde{r}_{i,j} , 0 \leq i \leq (M-1), 0 \leq j \leq (N-1) , \text{ in vertical mode}
\]

\[
r_{i,j} = \sum_{k=0}^{j} \tilde{r}_{i,k} , 0 \leq i \leq (M-1), 0 \leq j \leq (N-1) , \text{ in horizontal mode.}
\] (8.7)

Thus, the RDPCM is implemented in one-dimensional direction in the HEVC-SCC. However, two dimensional RDPCM was suggested as [28]

\[
\tilde{r}_{i,j} = \alpha_{1} r_{j-1,i} + \alpha_{2} r_{i-1,j} + \alpha_{3} r_{i,j-1} + \alpha_{4} r_{i-1,j-1} + \ldots
\] (8.8)

where \(\alpha_{i}\) denotes weighting factor for neighboring pixels.

To find the best mode for the current residual block, the SAD distortion metric can be used for each mode (i.e. horizontal, vertical or no RDPCM mode). The mode with minimum SAD is selected as the best.

When performing the RDPCM on a block, samples in the first column and the first row for horizontal and vertical RDPCM, respectively, are not predicted. Therefore it is beneficial to exploit redundancy by performing prediction on these samples in the direction orthogonal to the main RDPCM direction, as shown in Figure 8.11.
8.3.2 Sample-based weighted prediction with directional template matching

The sample-based weighted prediction (SWP) algorithm is proposed in [30] to introduce a weighted averaging of neighboring pixels for intra-prediction of the current pixel. The predicted pixel \( p_{\text{SWP}}[i] \) is calculated as follows:

\[
p_{\text{SWP}}[i] = \text{round} \left( \sum_{j \in S} w_{\text{int}}[i, j] g[j] / \sum_{j \in S} w_{\text{int}}[i, j] \right),
\]

(8.9)

where \( g[j] \) is the pixels around the reconstructed current pixel, \( S \) is the set of supporting pixels, and \( w_{\text{int}}[i, j] \) are the integer weighting values, which are calculated as follows:

\[
w_{\text{int}}[i, j] = \text{round} \left( a_{\text{SWP}} \cdot b_{\text{SWP}}^{-\text{SAD}(P[i], P[j])^{1/2}} \right),
\]

(8.10)

where the factor \( a_{\text{SWP}} \) is chosen to be \( 2^{11} \), if the internal bit depth is 8, the basis factor \( b_{\text{SWP}} \) is 2 for exponential decaying weights, and the parameter \( h_{\text{dir}} \) is empirically chosen to be 4.75 for luma and for 4:4:4 chroma. \( \text{SAD}(\cdot) \) is the operator for the sum of differences between two patches, \( P[i] \) and \( P[j] \), which are formed by causally neighboring pixels (typically four pixels as shown in Figure 8.12). Thus, the \( \text{SAD}(\cdot) \) can be defined as a similarity measure in the supporting area and given by

\[
\text{SAD}(P[i], P[j]) = \sum_{i \in N_v} |g[i+v] - g[j+v]|.
\]

(8.11)

Pixel X is predicted by a weighted average from the candidate pixels a, b, c, and d. For each candidate pixel, the \( \text{SAD} \) of the corresponding patches is calculated, e.g., for calculation of the \( \text{SAD} \) between X and b, the patch for X (pixels a, b, c, and d in the center of the figure) is compared to the patch for b (shaded blue area on the right hand side of the figure).
The prediction performance of the SWP is well for natural images that maintain high correlation among neighboring pixels, while degrades for such as text images that contain sharp edges among letters. With the weighted/averaged prediction, sharpness may be lost by averaging effect. Thus, another compromising prediction technique, directional template matching (DTM) has to be introduced. The main idea is to reduce the averaging effect by selecting the minimum SAD patch within the support area. The sharp edges could be kept without smoothing.

8.3.3 Sample-based angular intra-prediction

HEVC has adopted block-based angular intra-prediction which is useful for lossless coding to exploit spatial sample redundancy in intra coded CUs. A total of 33 angles (Figures 5.8 and 5.9) are defined for the angular prediction that can be categorized into three classes: 1 diagonal, 16 horizontal, and 16 vertical predictions. In HEVC, the total number of intra prediction modes is 35, including Mode 0 for INTRA_PLANAR, Mode 1 for INTRA_DC, and Mode 2 to 34 for INTRA_ANGULAR [31]. Given an \( N \times N \) prediction unit (PU), the number of reference samples are \( 4N+1 \), i.e., 2N upper, 2N left, and 1 diagonal, belonged to neighboring PUs. All the samples inside the PU share the same prediction angle in the block-based angular intra-prediction. The value of prediction angle should be informed to the decoder. However, the sample-based angular prediction is performed sample by sample. Since four effective intra prediction block sizes ranging from \( 4 \times 4 \) to \( 32 \times 32 \) samples, each of which supports 33 distinct directions, there are 132 combinations of block sizes and prediction directions. The prediction accuracy is 1/32 in the horizontal or vertical direction via linear interpolation.

The coding gain of the SAP provides a 1.8% to 11.8% additional bitrate reduction on average [32] in the lossless coding mode. In addition the SAP provides more gain in 10-bit configurations due to the fact that there are more blocks with large prediction residuals, the difference between the original pixel value and its prediction, in the 10-bit video than in 8-bit video. The SAP also improves coding efficiency by increasing the usage of angular intra prediction and the number of intra coded
CUs, while decreasing the usage of the planar and DC modes.

8.3.4 Sample-based angular intra-prediction with edge prediction

Another type of compound contents includes whole slide images (WSIs), which is the digitized version of microscope glass slides. Pathologists can send WSIs to others for sharing, collaborating, consulting and making diagnosis. WSIs usually feature a high number of edges and multifunctional patterns due to great variety of cellular structures and tissues. They should be scanned at high resolutions, resulting in huge file sizes. Therefore, designing efficient and accurate lossless or visually lossless compression algorithms are an important challenge. Edge prediction is proposed in [33] as a post-processing step on the residual signal computed by the original intra coding process. This method adds an extra coding step to the pipeline and alters the block-wise coding structure of HEVC as in [30].

In order to maintain the inherent block-wise coding and decoding structure of HEVC, an alternative intra coding modes are suggested in [34], while HEVC-RExt includes the optional use of the SAP, which is limited to the horizontal and vertical directions. For the case of the DC mode, a sample prediction is computed as an average of neighboring samples at positions \( \{a, c\}; \; P_{x,y} = (a + c) >> 1 \) in Figure 8.12. For the case of the PLANAR mode, an edge predictor is proposed and calculated as follows:

\[
P_{x,y} = \begin{cases} 
\min(a,c) & \text{if } c \geq \max(a,c) \\
\max(a,c) & \text{if } c \leq \min(a,c) \\
 a + c - b & \text{otherwise}
\end{cases}
\]  

(8.12)

The edge predictor mode and the SAP modes require that samples be decoded sequentially and be readily available for the prediction and reconstruction of subsequent samples. This inevitably breaks the block-wise decoding structure of HEVC. However, the reconstruction can be regarded as a spatial residual transform that only depends on the residual samples and the reference samples. Such spatial residual transform can be expressed in matrix form and applied during the decoding process in order to maintain the block-wise decoding structure. This technique improves coding efficiency by an average of 6.64% compared to SAP-1, which applies the SAP in all angular modes with a constant displacement among any two adjacent modes, and by an average of 7.67% compared to SAP-HV, which applies the SAP only in the pure horizontal and vertical directions [34]. This is mainly due to the introduction of a DC mode based on the SAP and an edge predictor in lieu of the PLANAR mode.

Although the edge predictor is capable of detecting horizontal and vertical edges accurately by
selecting the best predictor for each pixel, more efficiency needs to be achieved in textual smooth regions by taking median values of adjacent five pixels as

$$P_{x,y} = \text{median}\{a, b, c, d, e\}$$

(8.13)

By adding the median prediction, coding efficiency increases by 16.13% on average compared to HEVC intra prediction coding [35].

### 8.4 Fast coding algorithms

As discussed in chapter 5, the design of HEVC was based on two main objectives: achieving higher coding efficiency compared to previous standards and attaining low enough complexity to enable video applications on mobile devices. The second objective, however, is not easily fulfilled due to highly complex algorithms such as quadtree-based coding structure, large block transforms, an advanced motion prediction, additional filtering operations, and a high number of intra-prediction modes [36]. Among those properties, motion prediction and intra-prediction can be taken into account to design the screen content codec, responding to different nature from camera captured sequences.

#### 8.4.1 Adaptive motion compensation precision

Fractional precision motion compensation usually improves the video coding efficiency, especially for natural video. But considering the fact that slow moving or text-based screen content may be motion compensated in integer precision, using fractional precision is a waste of bits. Thus, adaptive precision methods are developed to improve coding efficiency of screen content, saving bits by using integer motion vectors in some cases. Encoder is designed to use integer precision when encoding the content captured from a screen and to use fractional precision when encoding the content captured from a normal camera.

To adopt the adaptive precision method in the HEVC structure, encoder should signals a flag at the slice header level to indicate whether integer precision or fractional precision is used for the current slice. Moreover, two-pass encoding is required to decide the precision of motion vectors, taking approximately double the encoding time, which is undesirable in practical use. Thus, fast algorithm has to be developed to minimize the encoding time while preserving most of the benefits brought by adaptive motion compensation precision.

In [37], fast algorithm is developed to efficiently design hash-based block matching scheme, using cyclic redundancy check (CRC) as the hash function to generate the hash value for every block. The complexity of CRC operation is about $O(m \cdot n \cdot w \cdot h)$, by considering the block size and picture size. More than 8G operations are required to calculate all the hash values of 64x64 blocks in 1080p
video [38], for example. To reduce the complexity of hash table generation, reuse of the intermediate results is suggested. In fact, there exists \((m-1)\) overlapping rows between two block operations. Thus, 63 of the intermediate data could be reused in the next 64x64 block, resulting in reduced complexity \(O(m \cdot w \cdot h + n \cdot w \cdot h)\), i.e., about 256M operations for the same example. Next step is the block matching using the hash values and another reduction is possible. This can be done by checking all the blocks having the same hash value and selecting a block to predict the current block. The complexity of block matching is about \(O(l \cdot (m \cdot n))\), where \(l\) denotes the number of blocks in a picture, resulting in about 2M operations for example above. The overall complexity of the proposed hash-based block matching method is about \(O(m \cdot w \cdot h + n \cdot w \cdot h + l \cdot m \cdot n)\), resulting in reduced complexity from 4T (full picture search) to 258M, which is more than 15,000 \times speedup.

Although we can reduce the complexity in the block matching operations, further reduction is possible using the adaptive precision decision for screen content. For example, in [37], the blocks are classified into four categories: collocated matched blocks (C) that the optimal motion vector should be \((0, 0)\), smooth blocks (S) that every row or column has a single pixel value, matched blocks (M) that an exact match by hash values can be found, and other extra blocks (O). Some logical analysis can be given as follows:

- If the percentage of O is too large, fractional precision MV is used.
- If the percentage of M is not large enough, fractional precision MV is used.
- If the percentage of C, S, and M is larger than a threshold, integer precision MV is used.

Using the adaptive precision MV, the maximum bit saving was 7.7% for the YUV Desktop sequence under Low Delay coding structure without a significant impact on encoding time.

### 8.4.2 Fast intra coding

HEVC SCC has adopted several enhanced coding methods to improve compression efficiency by considering the properties of computer generated contents. Due to lot of redundancy in the spatial domain, intra prediction and processing are mainly dealt with in the standard. Slight changes can be made to generate coding tree unit (CTU) and other intra-based tools such as intra block copy, palette mode, adaptive color transform are newly introduced. Many researches are focused on these intra coding techniques that a large amount of computation is required.

CTU is a quad-tree structure that can be a CU or can be split into four smaller units, if necessary. Since the SCC often includes a lot of redundancy in the spatial domain, the CTU structure may be different from that of normal HEVC contents. Fast CTU partition algorithm is suggested based on entropy of the individual CU [39]. The entropy is quite low in the screen content, because most of area is smooth and the pixel values are mostly equal. Several rules are obtained to terminate CTU partition earlier. For example, if the entropy of a 64x64 CU is 0, if the entropy of its four 32x32 sub-
blocks are equal, or if there are two 32x32 sub-blocks that have the same entropy as the other two 32x32 sub-blocks, then partition may be terminated. Using this method, the encoding time attained 32% reduction on average, BD rate exhibited 0.8% loss, and PSNR obtained a 0.09% loss, compared to the test mode HM-12.1+RExt-5.1. Fast CU partition decision using machine learning with features that describe CU statistics and sub-CU homogeneity is suggested in [40], achieving 36.8% complexity reduction on average with 3.0% BD-rate increase.

The IBC is a prediction method that finds a matched block within a current frame and sends a block vector (BV) information to the decoder. Since the amount of BV data is significant, the block vector prediction (BVP) is used to reduce the BV data. A block vector difference (BVD), which is calculated by the difference between a BV and a BVP, is coded by the 3rd order exponential Golomb coding. The IBC using these techniques provides a significant coding gain up to 69.39% BD-rate [41] at the cost of computational complexity increased by more than 30% [42]. Therefore, fast algorithm for block vector search in the IBC is considered. One way is to terminate the block vector search early as possible using a threshold when computing SAD between the block and predicted block. The threshold value is defined based on statistical experimental results. The average time savings was 29.23% with BD-rate 0.41% compared to SCM-2.0 [42]. Other algorithms [43] using thresholds and block activities were reported to reduce the block matching time.

HEVC SCC has also adopted the transform skip mode (TSM). Since screen content usually has more regular and sharper edges, prediction method might work well with no transform that may be inefficient and even worsen the coding performance. For these TUs, DCT or DST is skipped and transform_skip_flag (TSF) is signaled to the decoder. Thus, it is based on the similar principle of IBC that sample values are predicted from other samples in the same picture. These two techniques are highly related in terms of statistical occurrence. For example, up to 94.5% of TSM occurs in IBC-coded CUs [44]. Note that coded_block_flag (CBF) is sent for every TU, indicating all transform coefficients are zero, if CBF is set to 0 and any of transform coefficients are non-zero, if it is set to 1. Thus, we can save bits for two flags by careful modification of signaling, such that, for a 4x4 TB of IBC-coded CU, TSF is not signaled and CBF plays the same role as TSF. That is, when CBF is 1, it indicates TSM is selected in encoder. By the method and clever adjustment, [44] obtained reduction of 4x4 transform block encoding time by 28.2% with a slight increase of BD-rate by 0.22%.

The most probable mode (MPM) is used to reduce bits in the 4x4 intra-prediction instead of nine prediction modes. The encoder estimates the MPM for the current block based on the availability of neighboring blocks. If the MPM is the same as the prediction mode, we need to send only one bit instead of four bits. In screen content, if all boundary samples are exactly having the same value, all samples can be filled by the boundary samples with the MPM index and any rate-distortion
optimization (RDO) can be skipped, named simple intra prediction (SIP) in [45]. Direct prediction from boundary samples is possible by introducing single color mode [46] and independent uniform prediction mode [47] [48].

8.5 Visual quality assessment
The main objective of image/video coding is to compress data for storage and transmission, while retaining visual quality reasonable to human eye. The simplest way to assess visual quality is peak signal-to-noise ratio (PSNR) that is computed by difference between the original data and reconstructed data. If it is infinite, that means there is no difference and the quality loss does not exist. However, the lossy coding usually results in degradation due to quantization after prediction and transform. Therefore, it is necessary to take out the irrelevant data to human sensitivity that should be defined by certain dedicated models.

8.5.1 Screen image quality assessment
Numerous researches have been performed to develop perceptual quality assessment for images (IQA) that can be classified into three categories depending on the type of contents: natural image quality assessment (NIQA), document image quality assessment (DIQA), and screen image quality assessment (SIQA). It can be further classified into two categories depending on the measurement method: objective and subjective quality assessment. Objective assessment is preferred with its advantage: firstly, they are usually low complexity and secondly, we can classify distortions into several known components such as blocking, ringing, and blurring. Widely used objective assessment metrics are PSNR, SSIM (Structural Similarity) [49], gradient-based [50], image feature-based, and machine learning-based algorithms. There, of course, have been some limitations that they may not be exactly suited to real human observers. Subjective quality assessment is a human judgment-based method. Several test procedures have been defined in ITU-R Rec. BT.500-11: namely, SS (Single Stimulus), SC (Stimulus Comparison), SSCQE (Single Stimulus Continuous Quality Evaluation), DSIS (Double Stimulus Impairment Scale), SDSCE (Simultaneous Double Stimulus Continuous Evaluation), and DSCQS (Double Stimulus Continuous Quality Scale). Another classification is possible depending on the existence of reference images: full-reference, reduced-reference, and no-reference IQA algorithms [51]. The result of quality assessment is reported as either a scalar value or a spatial map denoting the local quality of each image region. Some of best-performing algorithms have been shown to generate quality estimates that correlate with human ratings, typically yielding Spearman rank-order and Pearson linear correlation coefficients in excess of 0.9. These IQA algorithms are summarized in Figure 8.13.
Firstly, NIQA has been studied tremendously during the last several decades. Natural images are usually obtained by visual camera that produces pictorial data. Recently, DIQA has attracted attention in the research community due to the necessity of digitization of old documents or imaged documents that their original features should be maintained. Most DIQA algorithms are designed in no-reference manner, since the original documents may not exist. The effectiveness of DIQA methods can be expressed by accuracy of character recognition. Since SCIs include pictorial regions beside textual regions without environmental degradations, features quite differ from those of the document images and DIQA methods cannot be directly adopted to evaluate the visual quality. The NIQA methods cannot be applied to evaluate the quality of SCIs either. Thus, new screen image database and quality assess metrics have to be developed. In [52], 20 reference and 980 distorted SCIs are included in database that can be downloaded in [53]. Distorted images are generated by applying the typical seven distortions: Gaussian noise, Gaussian blur, motion blur, contrast change, JPEG, JPEG2000 (see chapter 5), and layer segmentation based coding (LSC) [54] that firstly separates SCIs into textual and pictorial blocks with a segmentation method and applies different encoding method.

### 8.5.2 Objective quality assessment

It has been observed that natural image and textual image have different properties in terms of energy in the spatial frequency domain. To examine this, we decompose images using Fourier transform and then compute energy of the frequency coefficients. Energy of natural images linearly falls off from low to high frequency, while that of textual images has a peak at high frequency, since there are lot of small characters and sharp edges. Thus, SCIs consisting of two or more different contents need to be evaluated by relevant IQA metrics for each content. Since the final decision for quality assessment is to be made for the compound whole images rather than regional images, we
still have to develop how to aggregate them.

There are various ways to classify textual and pictorial content, such as gradient-based [55], text detection [56], and segmentation-based, etc. In [57], a block classification approach is suggested by making use of the information content map computed based on the local variance in the 4x4 block. Since textual regions contain high contrast edges, the local information is higher than in pictorial regions. By applying an empirical threshold on the mean of the block information, the textual and pictorial regions can be separated. The quality of each content can be assessed by any methods, although the most popular one would be the SSIM that combines local luminance, contrast and structural similarities. The three types of similarity between the reference and distorted images are pooled into an aggregated quality index. In SCIs, however, some incorrect quality scores happen from the average pooling. Therefore, [58] suggests a structure induced quality metric (SIQM) based on structural degradation model (SDM) defined by

$$\text{SIQM}(r,d) = \frac{\sum_{i=1}^{M} \text{SSIM}_\text{MAP}(r_i,d_i) \cdot \text{SDM}(r_i)}{\sum_{i=1}^{M} \text{SDM}(r_i)} \quad (8.14)$$

where $r$ and $d$ denotes reference and distorted image signal and $\text{SDM}(r_i)$ is defined by

$$\text{SDM}(r) = 1 - \text{SSIM}(r,r_f) \quad (8.15)$$

where $r_f$ is generated by applying a simple circular-symmetric Gaussian low-pass filter. Distortion maps generated by this method show more highlighted around the texts than in the pictorial regions. Performance of the SIQM is 0.852 on average Spearman and Pearson correlation coefficient, while the SSIM produces 0.750.

Another pooling method is suggested in [57], based on weighted average of textual quality $Q_t$ and pictorial quality $Q_p$, defined by

$$Q_s = \frac{Q_t \cdot E(\omega_t) + Q_p \cdot E(\omega_p)}{E(\omega_t) + E(\omega_p)} \quad (8.16)$$

where $E(\omega_t)$ and $E(\omega_p)$ denote the expectation of the local energy for the textual and pictorial regions, respectively. These quantities take a role of weighting factor for each content. The higher local energy, the more importance in the region. $Q_t$ and $Q_p$ are computed by another weighted SSIM metric. Performance of this method is 0.851 on average Spearman and Pearson correlation coefficient, while the SSIM produces 0.744, which are similar to those in [58].

### 8.5.3 Subjective quality assessment

Subjective testing methodologies can be roughly categorized into two types: the single stimulus
and double stimulus. The former asks the viewers to rate the quality of one distorted image, while the later asks the viewers to rate the quality between reference and distorted images. After testing, mean opinion score (MOS) of ten levels is computed. The higher MOS value is, the more correlation with human eye is. It reveals that the subjective quality scores for SCC is better than that for HEVC [59]. That is, SCC provides better performance than HEVC at the same distortion level as shown in Figure 8.14. However, there are many factors affecting human vision when viewing SCIs, including area ratio and region distribution of textual regions, size of characters, and content of pictorial regions, etc. [60]. When testing by subjects, the consistency of all judgments for each image should be examined. It can be measured by the confidence interval derived from the value and standard deviation of scores. Generally, with a 95% confidence level, the testing scores is regarded as confident.

![Figure 8.14. Histogram of the MOS values for (left) SCC and (right) HEVC [59]. Higher MOS values are achieved by SCC © IEEE 2015.](image)

### 8.6 Other SCC algorithms

#### 8.6.1 Segmentation

Since the SCIs are mixed with text, graphics, and natural pictures, researchers have interested in segmentation of those into several regions and applied different compression algorithms to different image types. In [61], two-step segmentation was developed: block classification and refinement. The first step is to classify 16x16 non-overlapping blocks into text/graphics blocks and picture blocks by counting the number of colors in each block. If the number of colors is larger than a certain threshold, the block is classified as picture block. The underlying reason is that natural pictures generally have a large number of colors, while text has a limited number of colors. If the number of colors is more than a threshold, it will be classified into pictorial block, otherwise to text/graphics. In this step, it produces a coarse segmentation, because it may contain different image types. Therefore, a refinement segmentation is followed to extract textual pixels from pictorial pixels. Shape primitives such as horizontal/vertical line or rectangle with the same color are extracted and
compared the size and color with some threshold. Thus, it is called shape primitive extraction and coding (SPEC) \[61\].

In \[62\], foreground and background separation algorithm is developed. They use the smoothness property of the background and the deviation property of the foreground. The overall segmentation algorithm is summarized as: firstly, if all pixels in the block have the same color, it can be background or foreground by considering neighboring blocks. Second, if all pixels can be predicted with small enough error using least square fitting \[63\] method, it can be background. Third, they run the segmentation algorithm after decomposing the block size into four smaller blocks until the size 8x8. This algorithm outperforms SPEC in terms of precision and recall.

### 8.6.2 Rate control

The rate control is always an important factor to define codec’s performance even in SCC. It is required to decide importance of different types of images. The more bit rate for textual regions, the finer quality we obtain for them, while obtaining worse quality for pictorial regions. In video coding it is related to the frame rate. It helps to utilize the bandwidth more efficiently. Rate control can be performed in two procedures: bit allocation and bit control. Once available bits are allocated in GOP level, picture level, and CU level, the next step is to adjust the coding parameters so that the actual amount of bits consumed is close to the pre-allocated target bits. It is desired for a video codec to minimize the bit rate as well as to minimize the distortion which is caused by data compression coding. Thus, the rate control problem is formulated to minimize the distortion \( D \), subject to a rate constraint \( R \) to derive the optimal coding parameter \( P_{opt} \):

\[
P_{opt} = \arg \min_{P} D \quad s.t. \quad R \leq B
\]  

(8.17)

where \( B \) is the given bit budget. The coding parameter is a set including coding mode, motion estimation, and quantization parameter (QP). Such constrained problem which is too complicated to be solved in real video codec is converted into unconstrained optimization problem, called rate-distortion optimization (RDO) by using Lagrange multiplier \( \lambda \) as

\[
P_{opt} = \arg \min_{P} (D + \lambda R).
\]  

(8.18)

\( \lambda \) serves as a weighting factor for the rate constraint and also indicates the slope of the R-D curve \[64\]. In the practical applications, however, 8.18 is formulated in a simple form such as the quadratic R-D function \[65\], which has been adopted by most of video coding standards, defined by

\[
R = aQ^{-1} + bQ^{-2}
\]  

(8.19)

where \( Q \) is quantization scale instead of distortion due to its simplicity.
The work in [64] proposes $\lambda$ domain rate control for HEVC. The benefits of $R - \lambda$ rate control are: it is more equivalent to finding the distortion on the R-D curve and it can be more precise than adjusting integer QP since $\lambda$ can take any continuous positive values. It outperforms the R-Q model in 8.19 by 0.55 dB on average.

However, since the screen content different characteristics, e.g., abrupt changes, a more appropriate rate control scheme is required. The work in [66] proposes an enhanced algorithm based on the $R - \lambda$ model. First, they analyze the complexity of each picture using a sliding window to handle the discontinuities. Then, bits are allocated and the parameters of the model are adjusted. As a result, it decreases the distortion by 2.25% on average and improves the coding efficiency by 5.6%.

Another important aspect in screen content is that there are many repeating patterns among pictures and in the same picture as stated earlier. This feature is utilized to introduce the IBC in the screen content coder. Problem is that rate at picture level should be maintained at all coding process. The work in [67] proposes weighted rate distortion optimization (WRDO) solution for screen content coding. A weighting factor $\omega$ is now applied to 8.18 as

$$P_{opt} = \arg \min_p (\omega D + \lambda R). \quad (8.20)$$

This algorithm was already implemented in HEVC test model, HM-16.2 with uniform distortion weight for all blocks in the picture. $\omega$ is kept as 1 in the normal picture which has less influence, while $\omega > 1$ in the more important picture. In case of hierarchical coding structure, pictures in the highest temporal level are never used as reference pictures and $\omega = 1$. As $\omega$ is only determined by the coding structure, it is not related to the content. To solve the problem, the block distortion weight should be determined by each block's influence instead of the fixed uniform weight. In [67], block distortion weight is calculated in two ways: inter weight among pictures considering temporal correlations and intra weight within one picture considering the correlations in IBC process. Then, the overall distortion weight is obtained by taking both aspects together. It results in 14.5% of coding gain for the IBBB coding structure at the cost of 2.9% of coding complexity increase.

8.7 Summary

In the context of screen content coding, various aspects have to be taken into account, due to different characteristics from natural camera captured content coding. The SCC is an extension of HEVC standard with several new tools, including IntraBC, palette coding, adaptive color transform, and adaptive motion vector resolution, which were discussed in Section 8.2. IntraBC is a kind of motion estimation/compensation used in the natural video coding but it is performed in intra
picture, since there are many redundancies in the spatial domain. By using palette mode, we can increase coding gain by sending an index instead of real color value. For screen content, there may exist blocks containing different features of colors, which leads to less correlation. For these blocks, direct coding in the RGB space may be more effective. This gives motivation for adopting adaptive color transform. Motion vector resolution should also be adaptively decided in multi-featured screen content. In some case, it can be in fractional resolution, while in other case it should be in integer resolution.

Since there are strong correlation in the spatial domain of screen content, intra prediction is a key factor to be developed. Though it has been developed and standardized in modern video coding standards including H.264 and HEVC, further developments are required for screen video. For example, sample-based angular intra-prediction with edge prediction is useful for the SCC discussed in Section 8.3.

Developments of fast coding algorithms are necessary, since the SCC often requires high computational complexity. Intra coding and motion compensation are two main parts to be speeded up as discussed in Section 8.4.

The main objective of image/video coding is to compress data for storage and transmission, while retaining visual quality reasonable to human eye. Thus, screen image quality assessment has to be introduced in the research arena. We discussed and compared quality assessment methods depending on the type of contents: natural images, document images, and screen images in Section 8.5.

Due to different natures in the screen content, many other algorithms are still being developed such as segmentation and rate control as discussed in Section 8.6. A lot of research results are still being introduced in the literature. Some of them are appeared in Section 8.8, giving readers updated projects.

8.8 Projects

P.8.1. Tao, et al. [68] proposed a re-sampling technique for template matching that can increase prediction performance at the cost of overhead information: position, index, and value. Among these three, position will consume more number of bits than the other two. As a solution, they applied the similarity of non-zero prediction error of pixel positions. Implement the encoder and find the compression performance.

P.8.2. Zhang, et al. [69] proposed a symmetric intra block copy (SIBC) algorithm in which utilizes symmetric redundancy. They conducted a simple flipping operation either vertically or horizontally on the reference block before it is used to predict the current block. The flipping
operation is easy to implement with low cost. They achieved up to 2.3% BD-rate reduction in lossy coding on some sequences that have a lot of symmetric patterns. Please investigate to find the amount of symmetricity in all test sequences that can be downloaded from [15]. Compare the performance to normal IBC using the reference software [16].

P.8.3. Tsang, et al. [70] have developed a fast local search method that can be used for hash based intra block copy mode. Due to the high computational complexity for the IBC [19], they proposed fast local search by checking the hash values of both current block and block candidates. The encoding time is reduced by up to 25% with only negligible bitrate increase. Implement this algorithm and evaluate the performance using the latest reference software.

P.8.4. The Intra Block Copy (IntraBC) tool efficiently encodes repeating patterns in the screen contents as discussed in Section 8.2.1. It is also applicable for coding of natural content video, achieving about 1.0% bit-rate reduction on average. Chen, et al. [71] have worked for further improvements on IntraBC with a template matching block vector and a fractional search method. The gain on natural content video coding increased up to 2.0%, which is not so big, of course, comparing to the efficiency on screen content video. For example, Pang, et al. [72] achieves 43.2% BD rate savings. However, it reveals that there exists some sort of redundancy in the spatial domain of natural video. Carefully design and implement intra prediction with block copy method for natural video content. Evaluate the coding performance in terms of the BD rate.

P.8.5. Fan, et al. [73] have developed quantization parameter offset scheme based on inter-frame correlation, since the inter-frame correlation among adjacent frames in screen content videos is very high. Firstly, they define a measurement of inter-frame correlation and then, quantization parameter offset for successive frames is appropriately adjusted. Number of correlated sub-blocks is counted based on SAD and thresholding between two frames. The more correlation, the larger quantization parameter may be required. The maximum BD rate gain was over 3.8% and the average performance gain is over 2% compared with the reference software. Implement this kind of optimization problem using different correlation measures and block sizes.

P.8.6. Zhao, et al. [74] proposed Pseudo 2D String Matching (P2SM) for screen content coding. Redundancy of both local and non-local repeated patterns is exploited. Different sizes and shapes of the patterns are also considered. They achieved up to 37.7% Y BD rate reduction for a screen snapshot of a spreadsheet. Implement the P2SM and confirm their results.

P.8.7. Sample-based weighted prediction was discussed in Section 8.3.2. Sanchez [75] also proposed sample-based edge prediction based on gradients for lossless screen content coding in HEVC, since a high number of sharp edges causes inefficient coding performance for screen
content video using current video coding tools. It is a DPCM-based intra-prediction method that combines angular prediction with gradient-based edge predictor and a DPCM-based DC predictor. Average bit-rate reduction was 15.44% over current HEVC intra-prediction. Implement the edge predictor and apply it to the latest reference software. Describe the advantage obtained by the edge predictor.

P.8.8. The intra coding in the HEVC main profile incorporates several filters for reference samples, including a bi-linear interpolation filter and a smoothing filter [31]. Kang [76] developed adaptive turn on/off filters for intra-prediction method. The decision is based on two criteria: statistical properties of reference samples and information in the compressed domain. For the former, they used Mahalanobis distance for measuring distances between samples and their estimated distributions. For the latter, they used R-D optimization model in the compressed domain. Implement this method using different distance measures and different transforms. Analyze benefits from turning filters on and off. Derive the most efficient transform for coding screen content by taking into account structural information [77].

P.8.9. Natural video has been commonly coded in 4:2:0 sampling format, since the human visual system is less sensitive to chroma. Screen content video, however, is coded in full chroma format, since the downsampling of chroma introduces blur and color shifting. This is because of the anisotropic features in compound contents [78]. Nevertheless, downsampling is required to increase compression ratio as much as possible. S. Wang, et al. [79] proposed an adaptive downsampling for chroma based on local variance and luma guided chroma filtering [80]. Coding performance is measured in terms of PSNR and SSIM. Implement their adaptive downsampling scheme using screen content reference software. Compare the performance to that using full chroma format in terms of different performance criteria including BD rate.

P.8.10. Three transform skip modes (TSM) have been proposed during the development of HEVC, including vertical transform skipping, horizontal transform skipping, and all 2D transform skipping. Vertical transform skipping means applying only the vertical 1D transform and skipping the horizontal 1D transform. This is called 1D TSM. If the prediction errors have minimal correlation in one or both directions, transform doesn't work well. Thus, it can be skipped. The 1D TSM is efficient in the screen content which has strong correlations in one direction but not the other. J.-Y. Kao, et al. [81] developed the 1D TSM based on dynamic range control, modification of the scan order, and coefficient flipping. D. Flynn, et al. [82] discusses more on TSM. Evaluate their effectiveness using the latest SCM software.

P.8.11. Implement and compare the three intra coding techniques: intra string copy [83], intra
P.8.12. Just Noticeable Difference (JND) has been used for measuring image/video distortions. Wang, et al [86] developed the JND modeling to be used for compressing screen content images. Each edge profile is decomposed into luminance, contrast, and structure, and then evaluate the visibility threshold in different ways. The edge luminance adaptation, contrast masking, and structural distortion sensitivity are also studied. Develop the JND model for lossless and lossy coding of screen content video and evaluate it in terms of human visual sensitivity.

P.8.13. Some fast algorithms for coding screen content are discussed in Section 8.4, mainly dealing with intra coding and motion compensation. Another approach is proposed by Zhang, et al [87] to speed up intra mode decision and block matching for IntraBC. If we could take proper background detection, encoding time can be saved by skipping mode decision process in the region. Background region can be detected by some sort of segmentation technique discussed in Section 8.6.1. Derive the background detection algorithms by your own means and apply it for coding screen content.

P.8.14. Duanmu, et al [88] developed a transcoding framework to efficiently bridge the HEVC standard (chapter 5) and it’s SCC extension. It can achieve an average of 48% re-encoding complexity reduction with less than 2.14% BD-rate increase. It is designed as a pre-analysis module before intra frame mode selection. Coding modes are exchanged based on statistical study and machine learning. Implement this type of transcoding tool and confirm their results.

P.8.15. Chen, et al [89] proposed a staircase transform coding scheme for screen content video coding, that can be integrated into a hybrid coding scheme in conjunction with conventional DCT. Candidates for the staircase transform include Walsh-Hadamard transform and Haar transform [90]. The proposed approach provides an average of 2.9% compression performance gains in terms of BD-rate reduction. Implement this hybrid coder and evaluate the performance.

References


Dec. 2015. http://www-ee.uta.edu/dip. Click on courses and then click on EE5359. Scroll down and go to Thesis/Project Title and click on S. Kodpadi.


[53] SIQAD(Perceptual Quality Assessment of Screen Content Images), "https://sites.google.com/site/subjectiveqa/," [Online].


adaptive color transform (ACT), 5, 11, 20
color space conversion, 12
encoder optimization, 13
adaptive motion compensation precision, 19
adaptive motion vector resolution, 13
BD-rate, 8, 21, 29
block vector (BV), 7
fast intra coding, 20
intra block copy (IBC), 5, 6, 31
  hash-based, 7, 13, 20, 29
  symmetric, 29
intra line copy, 31
intra string copy, 31
Just Noticeable Difference (JND), 31
palette mode, 8
  coding entries, 10
  coding indices, 10
  palette derivation, 9
rate control, 26
residual DPCM, 14
sample-based prediction
  angular intra, 17
  angular intra with edge, 18
  weighted, 16
screen contents, 4
coding tools, 6
  features, 4
screen image quality assessment (SIQA),
  22
  objective, 24
  subjective, 25
segmentation, 25
string matching, 30
template matching, 28
transform skip, 5, 21, 30