Technical Report

ISO/IEC TR 23002-9

First edition
2024-07

Information technology — MPEG video technologies —

Part 9: Film grain synthesis technology for video applications

Technologies de l'information — Technologies vidéo MPEG —

Partie 9: Technologie de la synthèse du grain de film pour les applications vidéo

Reference number
ISO/IEC TR 23002-9:2024(en)

Contents

Page

Foreword ... v
Introduction ... vi
1 Scope ... 1
2 Normative references ... 1
3 Terms and definitions ... 1
4 Abbreviated terms ... 2
5 Conventions ... 2
5.1 General ... 2
5.2 Arithmetic operators ... 2
5.3 Bit-wise operators ... 3
5.4 Assignment operators ... 3
5.5 Relational, logical and other operators ... 3
5.6 Range notation ... 4
5.7 Mathematical functions ... 4
5.8 Order of operations ... 4
6 Overview of film grain technologies ... 5
6.1 General ... 5
6.2 Film grain technical characteristics ... 5
6.3 Film grain modelling ... 7
6.4 Film grain use cases and applications ... 8
6.5 Film grain workflow ... 8
7 Film grain synthesis ... 10
7.1 General ... 10
7.2 General description of film grain synthesis ... 10
7.2.1 General ... 10
7.2.2 Grain pattern template generation ... 11
7.2.3 Randomization ... 12
7.2.4 Local adaptation ... 15
7.2.5 Deblocking ... 17
7.2.6 Blending ... 17
7.3 Examples of film grain synthesis using the frequency filtering model ... 17
7.3.1 SMPTE RDD 5 ... 17
7.3.2 Variants based on SMPTE RDD 5 ... 19
7.4 Examples of film grain synthesis using the autoregressive model ... 20
7.4.1 FGC SEI message based autoregressive model ... 20
7.4.2 AFGS1 model ... 21
7.5 Example of film grain synthesis supporting both the frequency filtering and autoregressive models ... 24
7.5.1 General ... 24
7.5.2 Film grain template generation ... 24
7.5.3 Randomization ... 24
7.5.4 Local adaptation ... 24
7.5.5 Deblocking ... 25
7.5.6 Blending ... 25
8 Film grain analysis ... 25
8.1 General ... 25
8.2 Denoising and image analysis ... 26
8.2.1 Denoising ... 26
8.2.2 Edge and texture analysis ... 26
8.3 Determination of grain scaling function ... 27
8.3.1 General ... 27
8.3.2 An example of FGC SEI message scaling factor estimation ... 27

8.3.3 An example of AFGS1 scaling factor estimation ... 31
8.4 Determination of cut-off frequencies for frequency filtering model ... 31
8.4.1 General ... 31
8.4.2 An example of FGC SEI message cut-off frequency estimation ... 32
8.5 Determination of autoregressive model coefficients ... 33
9 Film grain metadata ... 33
9.1 General ... 33
9.2 Film grain characteristics SEI message ... 34
9.2.1 General ... 34
9.2.2 Interpretation of FGC SEI message syntax ... 34
9.3 AFGS1 metadata ... 36
9.3.1 General ... 36
9.3.2 Interpretation of AFGS1 metadata syntax ... 36
Annex A (informative) **Example implementations of the derivation of x/y offset ... 39**
Annex B (informative) **Example implementations of film grain synthesis technologies ... 41**
Bibliography ... 47

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*, in collaboration with ITU-T (as ITU-T twin H.Sup-FGST).

A list of all parts in the ISO/IEC 23002 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

Film grain synthesis technology can provide subjective quality benefits for certain video applications and can be used to effectively achieve improved video compression. The use of such technology can involve pre-processing to reduce film grain and sensor noise that is present in a video or image signal prior to compression. Metadata information can then be conveyed to a decoder and used to synthesize noise with similar characteristics as in the original content as a post-processing stage that follows the compression decoding process. This metadata can be signalled using appropriate mechanisms, such as the supplemental enhancement information messages that are supported by several video coding standards.

This document provides a referenceable overview of the end-to-end processing steps for film grain and sensor noise removal, estimation, parameterization, synthesis, and blending for consumer distribution applications. This document includes examples of encoder-side and post-decoding processing steps for grain blending for some of the currently defined technologies.

Information technology — MPEG video technologies —

Part 9: Film grain synthesis technology for video applications

1 Scope

This document provides a description of the film grain synthesis technology in video applications, including for use with Rec. ITU-T H.264 | ISO/IEC 14496-10, Rec. ITU-T H.265 | ISO/IEC 23008-2 and Rec. ITU-T H.266 | ISO/IEC 23090-3.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

Rec. ITU-T H.264 | ISO/IEC 14496-10, *Information technology — Coding of audio-visual objects — Part 10: Advanced video coding*

Rec. ITU-T H.265 | ISO/IEC 23008-2, *Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 2: High efficiency video coding*

Rec. ITU-T H.266 | ISO/IEC 23090-3, *Information technology — Coded representation of immersive media — Part 3: Versatile video coding*

Rec. ITU-T H.274 | ISO/IEC 23002-7, *Information technology — MPEG video technologies — Part 7: Versatile supplemental enhancement information messages for coded video bitstreams*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in Rec. ITU-T H.264 | ISO/IEC 14496-10, Rec. ITU-T H.265 | ISO/IEC 23008-2, Rec. ITU-T H.266 | ISO/IEC 23090-3 and Rec. ITU-T H.274 | ISO/IEC 23002-7 apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

4 Abbreviated terms

AFGS1	AOMedia film grain synthesis 1
AVC	Advanced video coding (Rec. ITU-T H.264 | ISO/IEC 14496-10)
DCT	Discrete cosine transform
FGS	Film grain synthesis
FGC	Film grain characteristics
HD	High definition
HEVC	High efficiency video coding (Rec. ITU-T H.265 | ISO/IEC 23008-2)
IDCT	Inverse discrete cosine transform
LFS	Linear feedback shift register
LUT	Look-up table
MCTF	Motion-compensated temporal filtering
SD	Standard definition
SEI	Supplemental enhancement information
UHD	Ultra-high definition
VSEI	Versatile supplemental enhancement information (Rec. ITU-T H.274 | ISO/IEC 23002-7)
VVC	Versatile video coding (Rec. ITU-T H.266 | ISO/IEC 23090-3)

5 Conventions

5.1 General

The mathematical operators used in this document are similar to those used in the C programming language. However, the results of integer division and arithmetic shift operations are defined more precisely, and additional operations are defined, such as exponentiation and real-valued division. Numbering and counting conventions generally begin from 0, e.g. "the first" is equivalent to the 0-th, "the second" is equivalent to the 1-th, etc.

5.2 Arithmetic operators

The following arithmetic operators are defined as follows:

+	Addition
−	Subtraction (as a two-argument operator) or negation (as a unary prefix operator)
*	Multiplication, including matrix multiplication
x^y	Exponentiation. Denotes x to the power of y. In other contexts, such notation is used for superscripting not intended for interpretation as exponentiation.
/	Integer division with truncation of the result towards zero. For example, 7 / 4 and (−7) / (−4) are truncated to 1 and (−7) / 4 and 7 / (−4) are truncated to −1.

÷ Used to denote division in mathematical formulae where no truncation or rounding is intended.

$\frac{x}{y}$ Used to denote division in mathematical formulae where no truncation or rounding is intended.

$\sum_{i=x}^{y} f(i)$ The summation of f(i) with i taking all integer values from x up to and including y.

x % y Modulus. Remainder of x divided by y, defined only for integers x and y with x >= 0 and y > 0.

5.3 Bit-wise operators

The following bit-wise operators are defined as follows:

& Bit-wise "and". When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.

| Bit-wise "or". When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.

^ Bit-wise "exclusive or". When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.

x >> y Arithmetic right shift of a two's complement integer representation of x by y binary digits. This function is defined only for non-negative integer values of y. Bits shifted into the MSBs as a result of the right shift have a value equal to the MSB of x prior to the shift operation.

x << y Arithmetic left shift of a two's complement integer representation of x by y binary digits. This function is defined only for non-negative integer values of y. Bits shifted into the LSBs as a result of the left shift have a value equal to 0.

5.4 Assignment operators

The following assignment operators are defined as follows:

= Assignment operator

++ Increment, i.e. x++ is equivalent to x = x + 1; when used in an array index, evaluates to the value of the variable prior to the increment operation.

– – Decrement, i.e. x– – is equivalent to x = x – 1; when used in an array index, evaluates to the value of the variable prior to the decrement operation.

+= Increment by amount given, i.e. x += 3 is equivalent to x = x + 3, and x += (−3) is equivalent to x = x + (−3).

−= Decrement by amount given, i.e. x −= 3 is equivalent to x = x − 3, and x −= (−3) is equivalent to x = x − (−3).

5.5 Relational, logical and other operators

The following operators are defined as follows:

==	Equality operator
!=	Not equal to operator
!x	Logical negation "not"
>	Larger than operator
<	Smaller than operator
>=	Larger than or equal to operator
<=	Smaller than or equal to operator
&&	Conditional/logical "and" operator. Performs a logical "and" of its Boolean operators, but only evaluates the second operand if necessary.
||	Conditional/logical "or" operator. Performs a logical "or" of its Boolean operators, but only evaluates the second operand if necessary.
a ? b : c	Ternary conditional. If condition a is true, then the result is equal to b; otherwise the result is equal to c.

5.6 Range notation

y..z range operator/notation.
This function is defined only for integer values of y and z. When z is larger than or equal to y, it defines an ordered set of values from y to z in increments of 1. Otherwise, when z is smaller than y, the output of this function is an empty set. If this operator is used within the context of a loop, it specifies that any subsequent operations defined are performed using each element of this set, unless this set is empty.

5.7 Mathematical functions

The following mathematical functions are defined as follows:

$$\text{Abs}(\,x\,) = \begin{cases} x & ; \quad x >= 0 \\ -x & ; \quad x < 0 \end{cases}$$

Ceil(x) the smallest integer greater than or equal to x.

$$\text{Clip3}(\,x, y, z\,) = \begin{cases} x & ; \quad z < x \\ y & ; \quad z > y \\ z & ; \quad \text{otherwise} \end{cases}$$

Floor(x) the smallest integer lower than or equal to x.

$$\text{Min}(\,x, y\,) = \begin{cases} x & ; \quad x <= y \\ y & ; \quad x > y \end{cases}$$

5.8 Order of operations

When order of precedence in an expression is not indicated explicitly by use of parentheses, the following rules apply:

— Operations of a higher precedence are evaluated before any operation of a lower precedence.

— Operations of the same precedence are evaluated sequentially from left to right.

Table 1 specifies the precedence of operations from highest to lowest; a higher position in the table indicates a higher precedence.

NOTE For those operators that are also used in the C programming language, the order of precedence used in this document is the same as that used in the C programming language.

Table 1 — Operation precedence from highest (at top of table) to lowest (at bottom of table)

operations (with operands x, y, and z)
"x++", "x−−"
"!x", "−x" (as a unary prefix operator)
x^y
"x * y", "x / y", "x ÷ y", "$\frac{x}{y}$", "x % y"
"x + y", "x − y" (as a two-argument operator), "$\sum_{i=x}^{y} f(i)$"
"x << y", "x >> y"
"x < y", "x <= y", "x > y", "x >= y"
"x == y", "x != y"
"x & y"
"x | y"
"x && y"
"x || y"
"x ? y : z"
"x..y"
"x = y", "x += y", "x −= y"

6 Overview of film grain technologies

6.1 General

This clause provides an overview of film grain technologies in the context of video/image compression and distribution. It includes historical information on the development of such technologies in subclause 6.2, information on some of the use cases and applications in subclause 6.3, and a high-level description of film grain modelling in subclause 6.3. More details are provided in subsequent clauses as follows:

— Clause 7 describes some of the already defined film grain synthesis technologies, and in particular the frequency filtering and autoregressive models.

— Clause 8 provides examples of technologies that can be used for film grain analysis, including techniques for video denoising, edge and complex texture detection, film grain characteristic analysis, and model parameter estimation.

— Clause 9 describes some of the film grain metadata that have already been defined in current image/video coding standards and specifications, and how each metadata is interpreted, if appropriate, in the context of the frequency filtering and autoregressive models.

— Example implementations of such technologies are then described in Annex A and Annex B.

6.2 Film grain technical characteristics

The multimedia distribution industry began using celluloid (analogue) film as the medium for capture, editing and distribution. Content distribution evolved to analogue technology and then digital technologies. During this evolution, the attraction to the visual characteristics of analogue film did not fade. Due to its physical nature, analogue film produced a visual experience that was appreciated by many. Film grain is one of the characteristics of analogue film and is considered a primary contributor to the visual appearance of analogue film, e.g. film look or also known as the cinematic look. Film grain is a product of the physical characteristics of analogue film. It refers to the spatiotemporal variations in optical density of processed film that resulted from photographically developing the light-exposed silver-halide crystals dispersed in photographic emulsion.[1] Images are thus formed by exposure and the development of these crystals. In colour images, where the silver is chemically removed after development, dye clouds (like soft, tiny grains) are formed on the sites where the silver crystals have been exposed. Grains are randomly distributed in the resulting image because of the random formation of silver crystals in the original emulsion. The naked eye cannot distinguish individual grains, which are about 2 microns down to about a tenth of that in size. Instead, the eye resolves groups of grains in an image, that an observer identifies as a grainy look that is commonly called “film grain”. This is illustrated in Figure 1. Another example is shown in Figure 2, with a different colour image formation process called “autochrome”. This was one of the first colour image techniques invented by Auguste and Louis Lumière in 1903, where a classical black and white photo emulsion was exposed through a colour filter made of a fine dust of potato starch dyed with different colours.

In general, the higher the image resolution, the higher likelihood for perception of film grain. Film grain can be clearly noticeable in cinema and high definition (HD) images; although it progressively loses importance in standard definition (SD) images and becomes imperceptible in smaller formats as described in Reference [2].

a) 4000 dots per inch (2.54 cm) scan of 2 mm × 2 mm area

b) raw negative 500× microscope view of 0.1 mm × 0.1 mm area

Figure 1 — Fuji Superia™1) 400 film

Figure 2 — 24 mm × 24 mm and 2.2×2.2 mm crops of a 1916 autochrome picture (13 cm × 18 cm glass plate) photograph by Albert Samama Chikly, scan courtesy of Ministère de la Culture / RMN-GP (France)

1) Fuji Superia™ 400 is an example of film product available commercially. This information is given to describe examples used in this document and does not constitute an endorsement by ISO or IEC of the use of this product.

Film grain appearance is therefore inevitable because of the physical nature of the process embedded in the film design itself. However, historically, it was considered as noise, and as such, technological advances have gone in the direction of its elimination.

The silver-halide crystals were engineered to be smaller and less visible, however due to the physical design and characteristics of analogue film, it was not possible to completely eliminate the grainy look. With the advancements of digital camera sensors and their widespread utilization, the grainy look has been mostly eliminated. Although digital sensors have brought many possibilities in terms of visual quality and visual processing, the “film look” lives on among professionals and film enthusiasts. Within the new era, film grain has turned into a visual tool and not just a by-product of chemical processing as in the case of analogue film stock.

Note that the term film grain also includes synthetic film grain that can be added in post-production to digitally captured high-value content for artistic effect or to mask imperfections in digital footage, which can otherwise look too sharp and unnatural. The term “film grain” can sometimes be used informally to refer to image sensor noise, particularly in low light and high-speed captures.

Therefore, perception of moderate grain texture is a desirable, often sought after, characteristic in motion picture and video productions. Although the exact effect of the grain is not clear, it is considered a requirement in the motion picture industry to preserve the grainy appearance of images throughout the image processing and delivery chain. Intuitively, in this context, the presence of film grain can help to differentiate ‘real-world’ images from ‘computer-generated’ material, which are commonly created with no film grain. Furthermore, it is possible that film grain provides some visual cues that facilitate the correct perception of depth in two-dimensional pictures.[2] Even when movies are captured with digital cameras, artificial film grain can be added at a post-processing stage to create a specific look, which artists qualify as “soft”, “organic”, or “living”. Synthetic film grain is also used to harmonize capture from different cameras, potentially mixing film and digital capture and different lighting conditions.

Film grain preservation during video distribution, and especially when targeting low bitrate applications, can be challenging for two reasons. First, compression gains related to temporal prediction cannot be fully leveraged because of the random nature of the grain. Film grain noise is temporally independent, and as a result, motion compensation cannot be efficiently used for its prediction. Second, the grain commonly appears at high spatial frequencies, and it is typically filtered with other noise by in-loop filters, such as deblocking filters, or due to the quantization process.[2] This challenge is more severe with recent coding formats, as bitrate gains have come along with noise elimination. In addition, introduction of pre-filtering in the video distribution chain can potentially remove film grain prior to compression. The use of quantization matrices[3],[4] could potentially assist in the preservation of some of the film grain within the video content, however this also can have severe limitations, especially at lower bitrates, and for streaming applications.

This report focuses on film grain technology from the video compression and distribution point of view. It includes encoder-side and decoder-side aspects. On the encoder side, film grain technology provides means to denoise and/or analyse source video to improve compression and to determine statistical characteristics of the film grain to be synthesized at the decoder. At the decoder side, film grain technology provides the means to synthesize and blend film grain with the decoded video.

6.3 Film grain modelling

To synthesize grain, use of light-dependent film grain model parameters can be useful, particularly for the simulation of photographic film grain, as photographic film grain is intrinsically light intensity (exposure) dependent. First, variation of film opacity is the result of a variation of grain density, as seen in Figure 1, which has an impact on the perceived grain. Also, film is organized in several layers (usually 3) for each colour component, with various light sensitivities to reach to its full dynamic range. Light-sensitive crystals have a distribution of sizes. Larger crystals capture more photons and are more likely to be exposed than smaller crystals, particularly in darker regions. This results in a dependency of the noise characteristics on brightness. An example with both grain size and strength variation is shown in Figure 3.

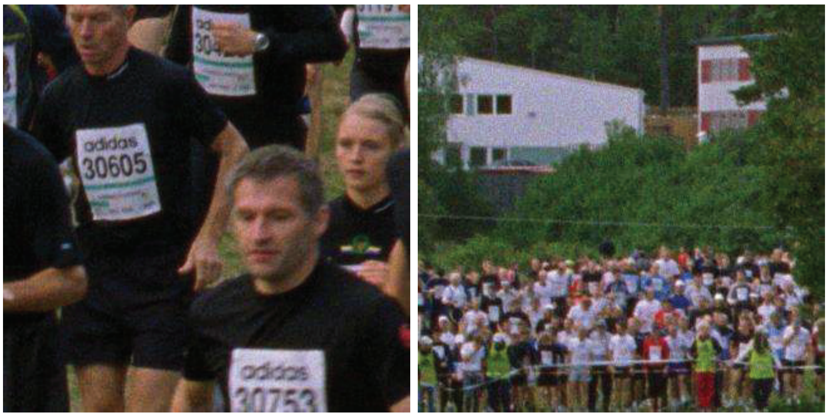


Figure 3 — Kodak Vision™[2)] 250D film and 3063 dots per inch (2.54 cm) scan of 2.7×2.7 mm areas of the same picture

6.4 Film grain use cases and applications

Two main film grain use cases are presented below:

a) The first use case of film grain synthesis is artistic intent: recreate the film grain at the decoder side, which was unavoidably lost by compression involved in content distribution at practical bitrates. In this case, the film grain is considered to be a significant aspect of the video, and the content provider wants it to be part of the user experience. Preserving film grain through video compression would require too high bit rates for applications such as adaptive streaming and broadcasting. On the other hand, removing film grain allows using the full potential of video compression technologies, while requiring film grain synthesis after decoding.

b) The second use case, which can also complement the first one, is the masking of compression impairments, like blocking, banding, “mosquito” noise, etc., including impairments due to quantization. If there is no artistic intent, then the constraints on film grain model accuracy can be relaxed. For this use case, the encoder can adjust film grain parameters to fit the coding parameters, so that the intended defect masking is effective (e.g. by adjusting noise amplitude based on quantization step sizes).

It was determined that removing the film grain by filtering the content, compressing, and providing information that enables the regeneration of the film grain, even if that is just an approximation, can result in more efficient coding performance and a better visual outcome. This is called film grain modelling. The use of film grain modelling technologies can be beneficial for image and video compression by providing improved subjective quality at a lower bitrate for certain types of video content. For example, these technologies can potentially provide benefits to video content that contains noise, such as film grain or image sensor noise.

Thus, film grain modelling technologies provide a means of optionally removing noise prior to or during the encoding process to improve compression efficiency and, subsequently, to reconstruct an approximation of the film grain during or after the decoding process. It can also be used to add visual noise to decoded video to mask or attenuate the visibility of compression artefacts. Note here that visually pleasant noise can be added to the decoded video even if the source video had no visible noise/film grain to fulfil the masking task mentioned above.

6.5 Film grain workflow

Use of film grain modelling technologies to denoise a source video by using a pre-processor is illustrated in Figure 4. Source video is input to a denoising process that outputs a video sequence from which noise or film grain is attenuated or removed. A film grain parameterization process then compares the source and denoised videos to determine film grain model parameter values, which relate to the variance, spatial frequency characteristics, colour correlation, and other statistical characteristics of the film grain. The

2) Kodak Vision™ 250D is an example of film product available commercially. This information is given to describe examples used in this document and does not constitute an endorsement by ISO or IEC of the use of this product.

process of denoising followed by the film grain model parameter estimation is commonly referred to as the film grain analysis process. Such process will be further discussed in Clause 8.

After these processes are performed, the denoised video is then encoded and the film grain model parameter values are either signalled in the coded bitstream or provided to the decoder by some external means.

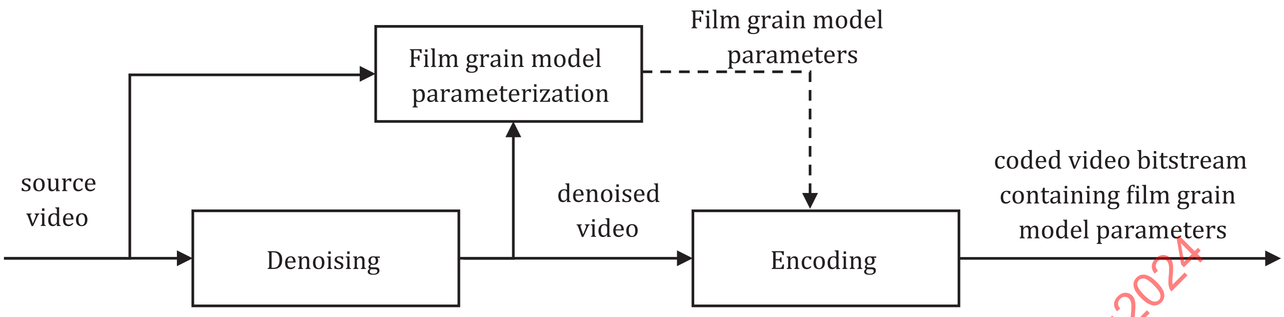


Figure 4 — Use of film grain modelling technologies with a denoising pre-processing stage

An alternative implementation of a film grain denoising and modelling system is also illustrated in Figure 5. In this case, the encoder itself acts as the denoising process.

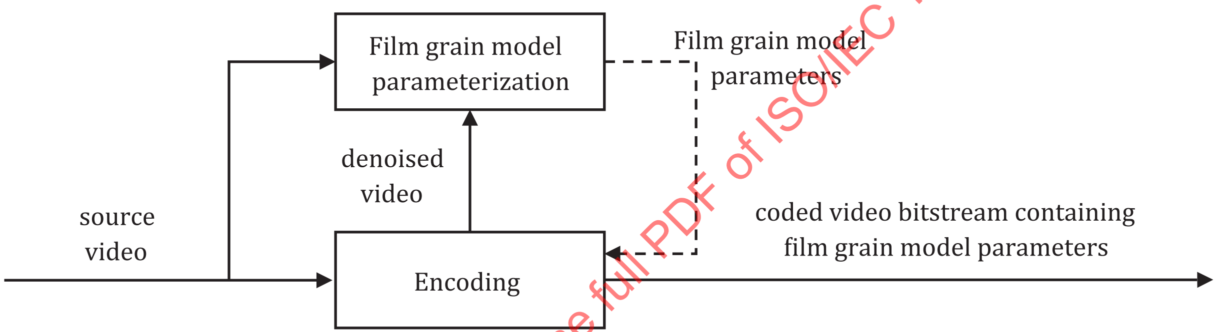


Figure 5 — Use of film grain modelling technologies with an encoder to denoise source video

The use of film grain modelling technologies with pre-determined film grain parameter values is illustrated in Figure 6. Note that film grain parameter values can, for example, have been pre-determined during post-production or when the statistical characteristics of the film grain are otherwise known *a priori*. The film grain parameters could be adjusted by the encoder depending on coding parameters or a default film grain configuration could be selected to mask coding artefacts. In such case, film grain can be added at the decoder side even if it was not present in the source video.

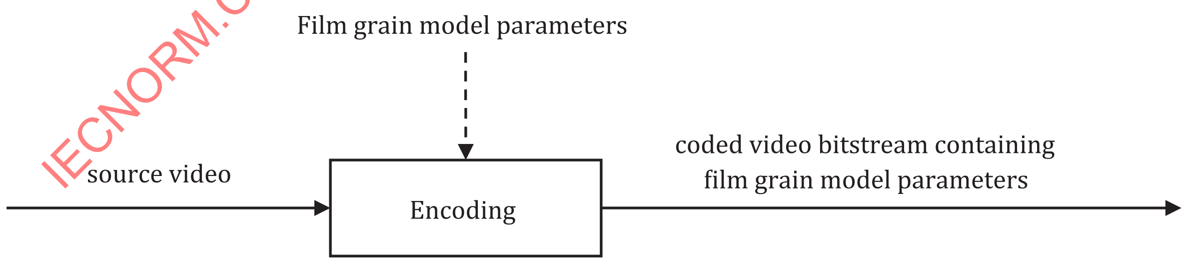


Figure 6 — Use of film grain modelling technologies with pre-determined film grain model parameter values

In any case, model parameters can be manually tuned/fine-tuned by a skilled person and provided to the encoder, for example, to be used as illustrated on Figure 6.

Figure 7 presents a decoding process, along with the film grain synthesis post-processing, for each of the examples provided in Figure 4 to Figure 6. At the decoder, the film grain model parameter values are parsed and input to a film grain synthesis process that generates simulated film grain and blends the grain with the decoded video to output decoded video with simulated film grain.

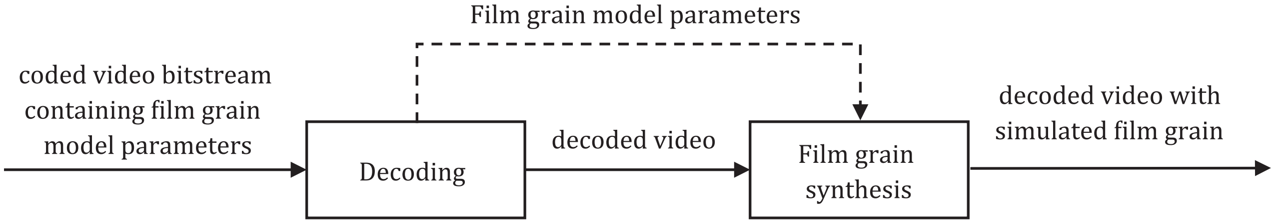


Figure 7 — Decoding process along with the film grain synthesis post-processing

7 Film grain synthesis

7.1 General

Film grain synthesis is commonly based on Gaussian noise generation, with spatial correlation modelled either by frequency limits or autoregressive parameters, and local adaptation that consists of adjusting grain amplitude and, optionally, correlation, to target image intensity levels.

Gaussian noise can be generated with a random generator and a Gaussian distribution table. The Gaussian noise generator can be run for every sample in the picture, along with spatial correlation methods. Alternatively, it can be run for a limited area (e.g. 64×64 samples), further called a “template”, that is then randomized to generate the full picture. Several such templates need to be generated when spatial correlation varies across intensity intervals (e.g. when film grain shape is not the same across the image).

Template pattern randomization (i.e. extension to the full picture) can be performed by dividing the picture into blocks smaller than the template (e.g. 16×16 or 32×32 blocks compared to a 64×64 template), and by choosing a pseudo-random offset within the template space for each of those smaller blocks. A pseudo-random sign inversion can be added to the random offset to improve randomization. When such a process is performed, deblocking can be needed across randomization blocks, especially when spatial correlation within the grain pattern is significant (in other words: when the grain is large).

Working with templates avoids running the Gaussian generator and correlation process for the full picture, which can be costly, not only because of more computations, but also because storing neighbouring lines (for spatial correlation) can be problematic in hardware. In contrast, using random offsets does not involve line storage but just reading pre-computed templates at specific locations.

Local adaptation can be based on sample intensity or a local average (e.g. the average intensity of a sub-block); a scaling factor and, optionally, a specific template are selected depending on the underlying image intensity.

7.2 General description of film grain synthesis

7.2.1 General

This subclause describes the general process of template-based film grain synthesis, including film grain template generation (see subclause 7.2.2), block-based randomization (see subclause 7.2.3), and local adaptation (see subclause 7.2.4), as illustrated in Figure 8.

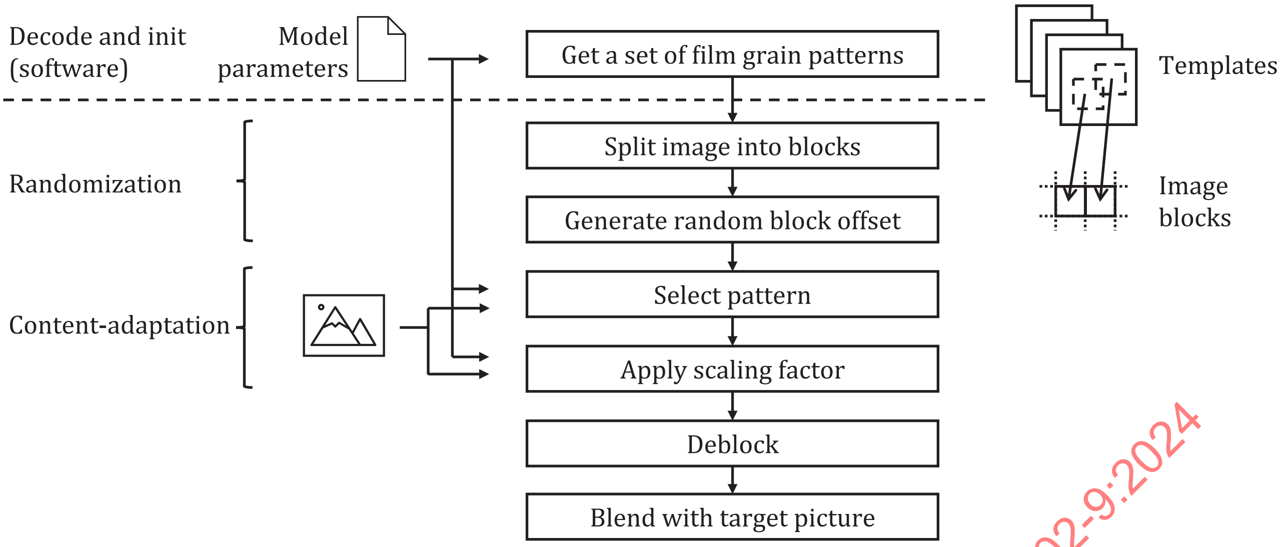


Figure 8 — Template-based film grain synthesis workflow

7.2.2 Grain pattern template generation

7.2.2.1 General

The first step in the synthesis process is to generate film grain pattern templates (small patches, typically 64×64 samples) according to the model parameters that are received by the decoder. When the model parameters specify different grain characteristics for different sample intensities, then as many templates can be generated. Some implementations can limit the number of templates, by potentially merging the characteristics for different intensity intervals. The limit on the number of templates available can be imposed by memory constraints. Depending on the implementation, templates can be precomputed and stored for further use (e.g. during the initialization process) or they can be created on-the-fly. The key model parameters are the amplitude and spatial correlation of the film grain as a function of the intensity (e.g. luma value) of the source video. Other model parameters relate to the bit depth and colour characteristics of the film grain compared to the source video, the way in which film grain is blended with the source video (additive or multiplicative), the persistence of model parameters from frame to frame, and the type of spatial correlation model used (frequency-filtering or auto-regressive, as explained below).

7.2.2.2 Frequency filtering model

In a frequency filtering model, the film grain characteristics are specified by horizontal and vertical spatial cut-off frequencies. The film grain template for a given set of cut-off frequencies can be generated as follows:

a) Generate a two-dimensional array of random-value elements having a normalized Gaussian distribution, here referred as n. The two-dimensional array represents discrete cosine transform (DCT2) coefficients. The column and row indices of the array represent horizontal and vertical frequencies, respectively. The array size N can be implementation dependent.

b) Set the value of all elements of the random-value DCT2 array to 0 when the corresponding column and row indices are not within the corresponding high and low horizontal and vertical cut-off frequencies. Also set the DC element to zero.

```
n(0,0)=0
for( y=0; y<N; y++ )
  for( x=0; x<N; x++ )
    if(( x<low_horizontal && y<low_vertical) ||
         x>high_horizontal || y>high_vertical )
      n(x,y)=0
```

c) Calculate the inverse discrete cosine transform (IDCT2) of the array produced in step 2 to get final film grain template *G*.

```
G = IDCT2( n )
```

7.2.2.3 Autoregressive model

The film grain characteristics could be specified by autoregressive filter coefficients instead of spatial cut-off frequencies. In this case, a film grain template can be generated using an autoregressive filter as follows:

$$G(x,y,c)=a_{0,0}*n+\sum_{i=-L}^{-1}a_{0,j}*G(x+i,y,c)+\sum_{i=-L}^{L}\sum_{j=-L}^{-1}a_{i,j}*G(x+i,y+j,c)+\sum_{k<c}b_k*G(x,y,k) \quad (1)$$

where n is a zero-mean Gaussian random variable, $a_{i,j}$ are autoregressive coefficients, b_k are colour correlation coefficients, L is a lag parameter, c is the colour component index, and $G(x,y,c)$ is the grain value for colour component c, at the sample location (x, y). An autoregressive filter with lag parameter value L equal to 2 is illustrated in Figure 9.

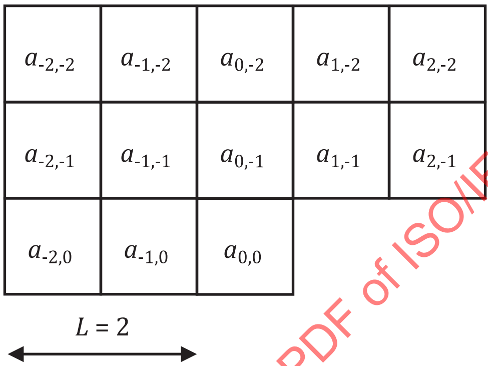


Figure 9 — Sample grid of autoregressive filter with lag parameter value *L* equal to 2

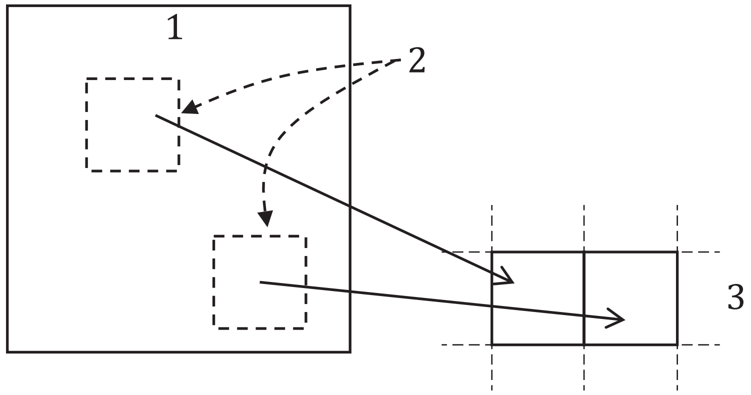
This is a generic description of an autoregressive film grain model in which, depending on the specific format used to transmit this information, some coefficients could be fixed or combined.

7.2.3 Randomization

7.2.3.1 General

Randomization consists of extending a film grain template (or a set of them), generated according to the model parameters, to the full picture.

It essentially consists in dividing the picture into blocks smaller than the template and, for each block, selecting a random region of the template as illustrated in Figure 10. Methods considered here can bring additional diversity by randomly inverting the sign of the template for each block, which does not change the statistical properties since the Gaussian noise is centred around zero. Other known methods of randomization are the use of the horizontal and/or vertical flip, which could be used if the film grain model is symmetrical, and rotations if the model is isotropic. Such methods are not further described here.



Key

1 template

2 random window

3 image blocks

Figure 10 — Image blocks each taking a random window from template, where 1, 2, and 3 represent the template, random window, and image blocks, respectively

Selecting a random region (or window) within the template(s) for a given block can be performed by generating a (pseudo) random x/y offset with the correct range, while considering the size of the template and of the window.

The challenge of randomization is to extend the template(s) to the full picture without the human eye noticing repeated elements. In fact, even though the template samples are pseudo-random, the resulting texture, or part of it, can be recognized by the human visual system as a pattern. If such a pattern is repeated in a predictable manner (see Figure 12), it can draw the attention of the observer. Typically, such repetitions would be masked by the video/picture content and changes in the grain between consecutive video frames. However, these patterns can become visible in smooth areas and in still pictures. The visibility of such patterns can also depend on picture resolution. Several approaches described further can be used to reduce visible repetitions.

7.2.3.2 Choice of initialization parameters for the pseudo-random generator

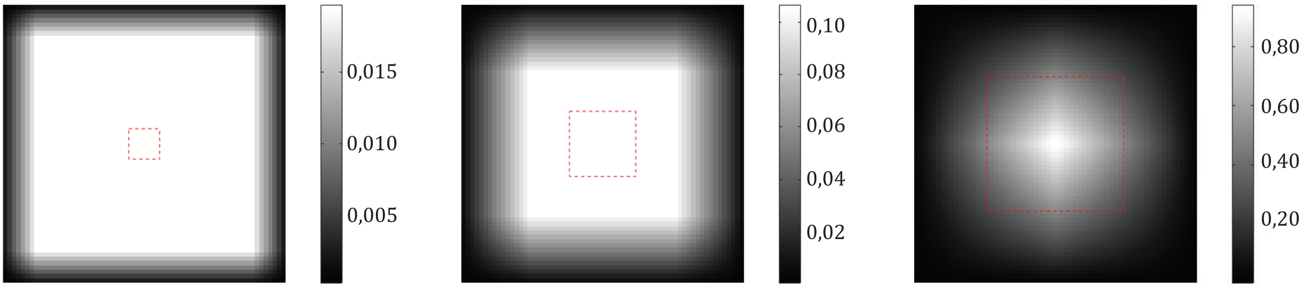
Certain pseudo-random Gaussian sequences can form somewhat prominent visual features when arranged in a rectangular template. Well-chosen initialization of the pseudo-random process helps in avoiding the generation of film grain templates with such prominent features.

One approach can be to transmit initialization values along with the model parameters, which has certain signalling cost but can guarantee that a desired pattern has been generated. Initialization can also be performed based on a set of known well-chosen initial values.

7.2.3.3 Block size

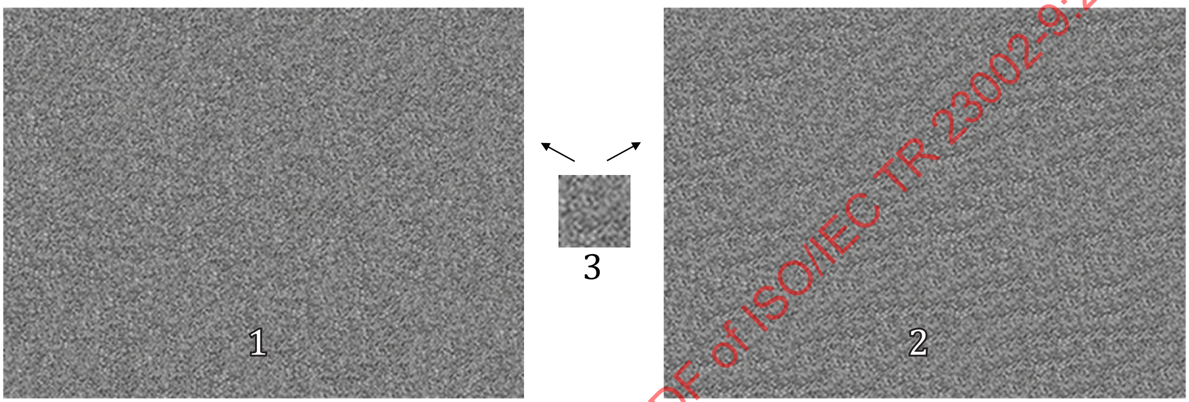
Another consideration to reduce visibility of repeated patterns is to choose a random window (matching block size) that is smaller than the template. The probability of template samples to be part of the random window for different sizes is shown in Figure 11. As an example, a window ¼ the size of the template in each dimension barely exhibits visible repetitions, while a window ½ the size of the template in each dimension can; the eye can recognize the centre of the pattern and identify displacements, as illustrated in Figure 12.

However, the window is kept large enough compared to the size of the grains, so that the statistics within the window are still representative of the grain pattern and, ultimately, the model parameters. The compromise between window size and template size likely depends on implementation considerations.



a) n/N = 1/8 b) n/N = 1/4 c) n/N = 1/2

Figure 11 — Probability of (N x N) template samples to be part of a random (n × n) window



Key

1 window = 1/4

2 window = 1/2

3 same pattern

Figure 12 — Potential visual impact of the size of the randomization window

7.2.3.4 Offset randomness

In addition to block size, randomization quality also depends on x/y offset randomness; adjacent blocks ideally avoid selecting similar offsets, which would mean a similar region of the template, thus visible repetitions. Similarly, repeating sequences of offsets would cause repetition of groups of blocks, which would be visible and is not desirable.

To reach these goals:

— The repetition period of the pseudo-random generator used to derive the x/y offsets has a direct impact on the potential repetition of offsets. In the case of an LFSR (linear feedback shift register), the repetition period is related to the order of the polynomial used. The larger it is, the better the randomness. In addition to the repetition period, a longer size allows extracting longer or more bit fields to generate different offsets and sign flips at the same time. Figure 13 illustrates an example of a 32-bit LFSR using a 31-order polynomial.

— Reducing the choice to a limited number of evenly spaced positions (e.g. one out of 4), as illustrated in Figure 14, ensures both a minimal spacing and better scrambling, because the (finite) pseudo-random possibilities are spread over a smaller number of larger displacements instead of many small (and potentially close) displacements. This has the additional benefit of enabling aligned reads of template memory in a practical implementation. On the other hand, the number of positions is kept large enough to allow sufficient randomization diversity.

— As the number of random values needed for the x and y offsets depends on the block size and offset alignment, this number is not always a power of two. In that case, care is needed in the derivation of offsets from the random generator (or a bitfield extracted from it) so that the random distribution of offsets is uniform (as assumed in Figure 11). This is discussed in Annex A, including hardware cost considerations.

— The successive offsets are ideally as far as possible from each other. Depending on how offsets are derived from the pseudo-random generator, the method can differ (examples are given in A.2).

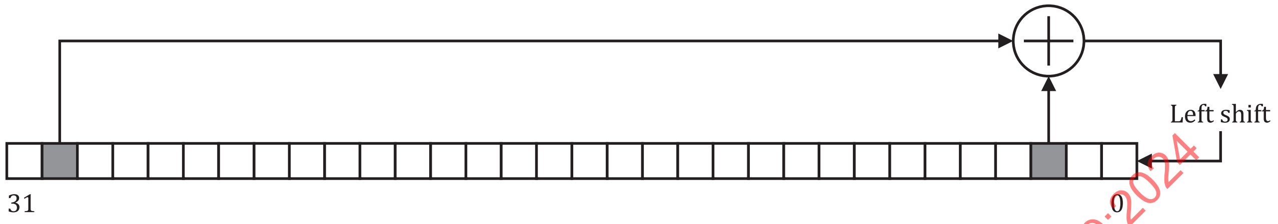


Figure 13 — LFSR with 31st order polynomial ($x^{31} + x^3 + 1$)

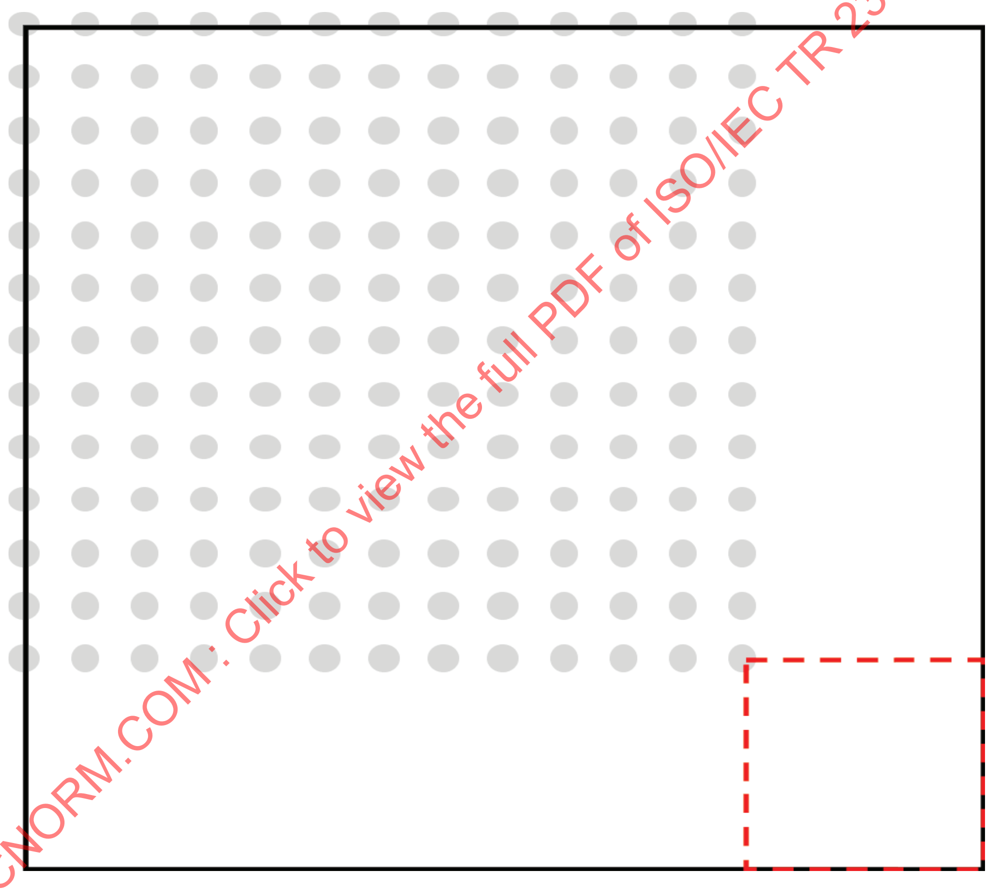


Figure 14 — Random window offsets limited to a few, evenly spaced positions (grey dots)

7.2.4 Local adaptation

7.2.4.1 General

As explained in the introduction, film grain characteristics vary significantly depending on exposure, both in terms of amplitude and spatial correlation, with the amplitude variation being the most obvious. These variations are inherent to the image formation process at the physical film level. Replicating them on synthetic grain makes it look natural by anchoring it into the image, while a fixed overlay would likely not be visually pleasing.

Here, local adaptation consists of selecting the relevant film grain template and grain amplitude according to sample values and model parameters. For example, a dependency of grain strength/amplitude on the signal intensity can be defined for each colour component. In addition, specific grain correlation parameters can depend on the signal intensity.

Local adaptation can be performed per sample or on a sub-block basis (based on the local intensity average), with different implications, which will be detailed in the following subclauses.

7.2.4.2 Adaptation of grain shape

Adaptation of grain shape consists of selecting a specific film grain template depending on local intensity, according to the model. This requires the initial stage to generate as many film grain templates as required to fit the model within practical implementation limits, as described in subclause 7.2.2. For implementations using a single template, local adaptation of grain shape is not supported.

Template selection can be applied either per sample or on a sub-block basis. This question arises because, since grain is spatially correlated, changing the template within a block can destroy this spatial correlation, even though the random window and potential sign flip (see subclause 7.2.3) is kept constant for the whole block regardless of the template selected:

— Selecting the template on a sub-block basis guarantees that the grain characteristics are preserved within the sub-block as long as the grain size is significantly smaller than the sub-block. However, that requires computing a local intensity average to select the relevant template. Computing an average over a sub-block requires line buffers, which can be too costly in hardware implementations. Also, changing template on sub-block boundaries can require deblocking so that the transition is not visible, as illustrated in Figure 15.

— Selecting a template on a sample basis does not require line buffers nor deblocking but requires solving the problem of spatial correlation preservation. This can be done during the template generation time, by ensuring template patterns are correlated with each other, making transitions smooth, as illustrated in Figure 16. For example, templates can be the result of filtering the same underlying Gaussian noise pattern with different frequency cut-offs or autoregressive coefficients. In implementations supporting a limited number of templates for storage cost reasons, transition smoothness could be further improved by pattern interpolation.

a) Target picture **b) Template selection** **c) No sub-block deblocking** **d) Sub-block deblocking**

Figure 15 — Sub-block-based template selection

a) Target picture **b) Template selection** **c) Uncorrelated patterns** **d) Correlated patterns**

Figure 16 — Sample-based template selection

7.2.4.3 Adaptation of grain amplitude

Adaptation of grain amplitude consists of scaling the amplitude of the film grain template selected in the previous step by a scaling factor that depends on local intensity, according to the model. Hereafter, the relationship of grain amplitude to local intensity is called the scaling function.

Compared to performing this adaptation on a sub-block basis, a sample-based approach is more similar to the physical process, more closely follows the image, and avoids transitions on a fixed grid, which could potentially be visible in a video sequence.

7.2.5 Deblocking

Deblocking is required because of block-based randomization (see subclause 7.2.3), especially when grain is large.

When local adaptation is performed on a sub-block basis, deblocking can be needed on sub-block borders too.

Deblocking traditionally involves low-pass filtering block borders. However, for horizontal borders, this means vertical filtering, requiring line buffers (typically two, when using 3 coefficients).

To avoid the cost of such line buffers in hardware implementations, it can be preferable to use an overlapping process instead of a filter. For example, on the first two lines, the samples of the current template matching two different random windows are blended together, the windows of the current block and of the block above. No line storage is required, just reading from two places in the template (the address of these places can be computed on the fly).

Another option is to attenuate grain amplitude on horizontal borders.

7.2.6 Blending

The final stage is to blend the synthesized film grain with the picture. Either multiplicative or additive blending can be performed, followed by appropriate clipping.

7.3 Examples of film grain synthesis using the frequency filtering model

7.3.1 SMPTE RDD 5

7.3.1.1 General

SMPTE RDD 5[5] specifies a fixed point, bit-accurately reproductible process for film grain synthesis that makes use of the FGS SEI message, with some restrictions:

— Frequency filtering mode (model_id = 0)

— Additive blending mode (blending_mode_id = 0)

— Limitation of the range of numbers, so that the computation bit depth is limited and practical (log2_ scale_factor from 2 to 7)

— No overlaps between intensity intervals

— The number of model parameters is limited to 3, limiting it to scale and high frequency cut-offs (no low frequency cut-offs, which means it uses low-pass filtering instead of band-pass, and no cross-component correlation)

— The scale parameter is limited to 8 bits, and the frequency cut-offs limited to the 2 ..14 range (full band being 15)

— 4:2:0 sampling format

The random generator is defined as an LFSR (the one illustrated in Figure 13) and an array of initial values is specified. A table of Gaussian values is also specified to enable conversion of LSFR values, which have a uniform distribution, into a Gaussian distribution. It also specifies an integer DCT2 transform matrix.

When the picture colour format is YUV 4:2:0, a specific rule is defined to convert the scaling factors and frequency cut-offs for chroma components, before any further processing. The process is then the same for all colour components (template size, randomization block size, etc, is the same).

7.3.1.2 Grain pattern template generation

The grain pattern template generation process is similar to what is described in subclause 7.2.2.2, with a 64×64 template size. It is made reproducible by making use of specific LFSR initialization values, the Gaussian table, and a specific DCT2 transform matrix.

In addition, vertical pre-deblocking is baked into the film grain templates, using an attenuation technique as described at the end of subclause 7.2.5; an attenuation factor is applied on two lines every 8 lines, with the factor depending on the vertical frequency cut-off. This can work because template selection is made constant within every 8×8 block in the current image, and the vertical random offset is a multiple of 8.

Note that since the initialization values are always the same, depending on the frequency cut-offs only, the template generation process does not have to be repeated for every frame. Also, as the frequency cut-off range is limited to 13 values in each direction, only 169 different templates exist, which could potentially be stored in a fixed database. However, only 10 different templates are allowed to be referenced by a given FGC SEI message (in other words: within a picture).

7.3.1.3 Randomization

The randomization block size is 16×16. Random offsets are derived from the LFSR, which is initialized with a value that depends on the colour component and picture identifiers read from the bitstream (poc and idr_pic_id). It is required that pictures 32 or fewer frames apart in decoding order do not have the same identifier.

Random offsets are extracted from 16-bit fields of the LFSR (16 MSBs for x, 16 LSBs for y), and mapped to the required range with a modulo operation as described in Annex A.1, with R equal to 16 and N equal to 52 or 56. The horizontal offset is a multiple of 4, with 13 possible values (0 to 48 in steps of 4). The modulo is then computed as 13 * 4 = 52, applied on the 16-bit field, and the two lower bits of the result are set to zero; this is equivalent to integer division by 4, modulo 13, then multiplication by 4, which also means that only 14 bits of the LFSR are actually used in the end. Similarly, the vertical offset is a multiple of 8, with 7 possible values (0 to 48 in steps of 8), giving modulo 56 (or 7 after integer division by 8), and actually using 13 bits of the LFSR.

Additionally, a random template sign inversion is derived from the LSB of the LFSR.

7.3.1.4 Local adaptation

Local adaptation of both the grain amplitude and spatial frequency limits is supported by selecting template and scaling factors based on the average sample value I over 8×8 blocks (non-overlapping). As explained in subclause 7.2.4.2, this has some hardware cost implications, needing a 7-line buffer to compute the 8×8 average. Template and scaling factor selection can make use of pre-computed look-up tables (LUTs) driven by intensity I, the scaling factor LUT being a representation of the scaling function (e.g. see scaling function $f(I)$ in Figure 17). Since SMPTE RDD 5 and its variants make use of the FGC SEI, that defines constant scaling factors per intensity interval, the scaling function is typically a stepwise function as illustrated in Figure 17, unless intervals are very small.

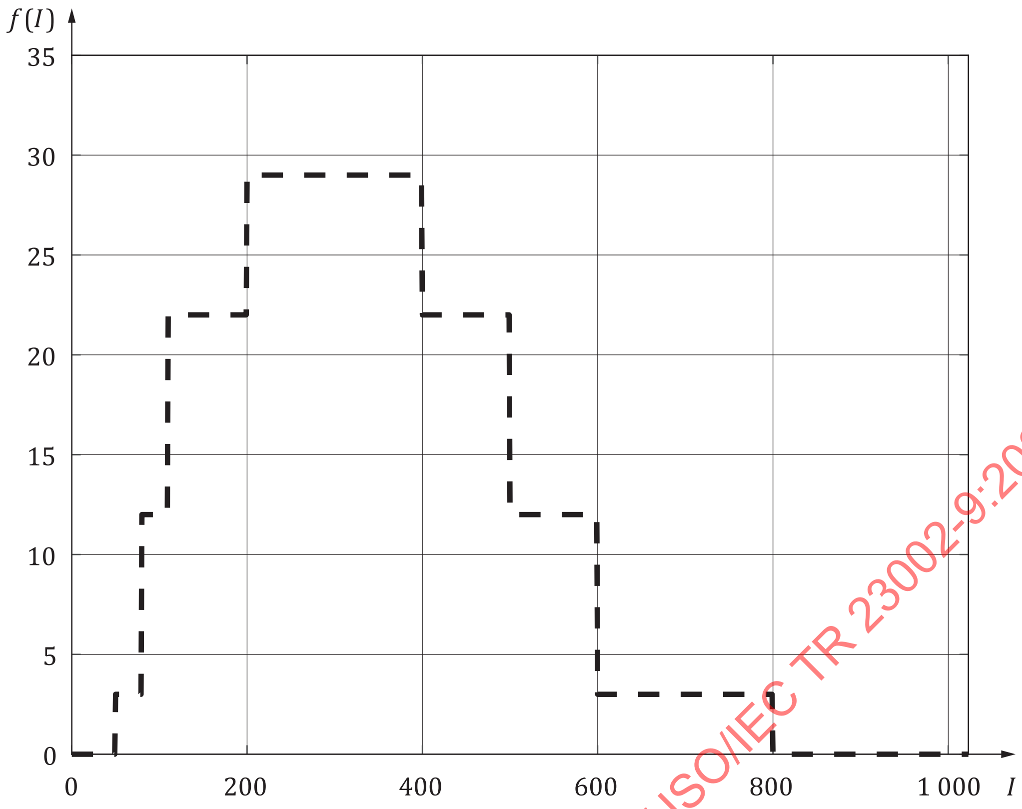


Figure 17 — Example of a scaling function f used for local grain amplitude adaptation in SMPTE RDD 5 and its variants, making use of the FGC SEI message that defines constant scaling factors per intensity interval

7.3.1.5 Deblocking

As discussed in subclause 7.2.4.2, since local adaptation can change template for each 8×8 block and template patterns are independent (uncorrelated with each other), deblocking is needed on 8×8 blocks, rather than (or in addition to) the 16×16 randomization blocks.

As exposed in the template generation subclause, vertical deblocking is handled during the template generation stage. Horizontal deblocking is performed by smoothing both sides of the block borders, using [0.25 0.5 0.25] filter coefficients.

7.3.2 Variants based on SMPTE RDD 5

7.3.2.1 Implementation in the VTM and HM

An example film grain synthesis implementation based on SMPTE RDD 5 is present in both the VTM[6] and HM[7] reference software. The differences from SMPTE RDD 5 are the following:

— Bit depth higher than 8 is supported, by scaling the film grain appropriately; for example, for 10-bit video, film grain is shifted left by 2 bits before blending.

— The VTM 64×64 inverse DCT2 transform is used for template generation.

7.3.2.2 Resolution-adaptive block size

In addition to what is described in subclause 7.3.2.1, this example implementation uses resolution-adaptive local adaptation blocks:

— 8×8 up to HD resolution.

— 16×16 up to UHD resolution.

— 32×32 above UHD resolution. In this case, the randomization blocks are also enlarged to 32×32, with an adjustment of modulo values for random offset derivation: 36 for horizontal offset and 40 for vertical offset.

7.3.2.3 Single-line block

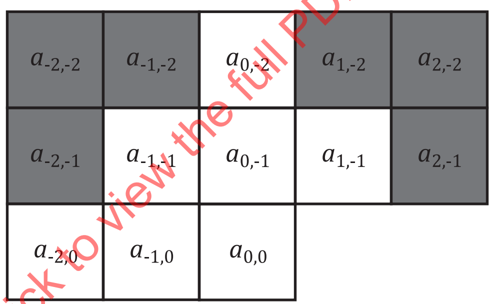
In addition to what is described in subclause 7.3.2.1, this example uses single-line blocks to avoid the usage of a line buffer in the average block intensity computation used for local adaptation. An example of block sizes used includes 8×1 for up to HD resolution, 16×1 for up to UHD resolution and 32×1 for resolutions above UHD.

This example also restricts the number of different templates within a picture to a single one (disabling local adaptation of pattern), to avoid the problem of template switching every line potentially destroying the spatial consistency of film grain, as explained in subclause 7.2.4.2.

7.4 Examples of film grain synthesis using the autoregressive model

7.4.1 FGC SEI message based autoregressive model

In the FGC SEI message, when using the autoregressive model as described in subclause 7.2.2.3, the lag parameter is not signalled explicitly. Instead, it is implicitly specified to be in the range of 0 to 2, inclusive, depending on the number of model values present for each intensity interval in which the film grain has been modelled. The coefficients allowed for the FGC autoregressive model are constrained as illustrated in Figure 18 where the variables p_i are independent parameters and A reflects the sample aspect ratio of the film grain. The film grain sample aspect ratio could be useful in cases in which a picture is stretched in resizing or was captured with isomorphic optics[2].



$$a_{-2,-2} = a_{-1,-2} = a_{0,-2} = a_{1,-2} = a_{2,-2} = 0$$
$$a_{0,0} = p_0$$
$$a_{-1,0} = p_1 \,;\, a_{0,-1} = A \times p_1$$
$$a_{-1,-1} = a_{1,-1} = A \times p_2$$
$$a_{-2,0} = p_3 \,;\, a_{0,-2} = A \times A \times p_3$$

Figure 18 — Autoregressive model parameters in the FGC SEI message

The following describes an example of an autoregressive model for film grain synthesis that follows the process described in the FGC SEI message semantics, using a similar process for each colour component.

This example process does not use template-based synthesis as described in subclause 7.2.1, but can hypothetically be used to generate film grain templates in a template-based scheme, where only the film grain template generation differs between autoregressive or frequency filtering model.

In this example, a Gaussian noise n with zero mean and unity variance is generated for each sample, scaled by coefficient $a_{0,0}$, and coefficients $a_{i,j}$ are applied to the causal neighbourhood, as described in Formula (1) and Figure 9, with coefficients $a_{i,j}$ derived from p_i and A as described in Figure 18.

Different parameters p_i and A can be defined for each intensity interval. Consequently, local adaptation is performed through coefficients in Formula (1) that can change for each target image sample, based on the intensity of the sample.

Depending on the number of model parameters specified in the content of the FGC SEI message, noise scale could change, a limited number of neighbourhood coefficients, or also A. Model parameters are specified in the following order: p_0, p_1, p_2, A, p_3. When less than five parameters are specified, the remaining ones take default values. The default value is zero for p_i and one for A. The same number of parameters is specified for each intensity interval.

The grain generated by this process for each colour component is then mixed on top of the target picture.

7.4.2 AFGS1 model

7.4.2.1 General

AFGS1[8] is a film grain synthesis specification that uses an autoregressive model for the generation of film grain. The technique can use an ITU-T T.35 user registered SEI message to signal the film grain characteristics. The specification uses an autoregressive model for the film grain template generation and a piece-wise linear function to model the dependency between the grain strength and the signal intensity.

7.4.2.2 Grain pattern template generation

AFGS1 uses an autoregressive model for representing the film grain pattern. This model is used to synthesize the film grain template.

In AFGS1, lags of size 0 to 3 are supported as illustrated in Figure 9 for a lag L of size 2.

The description above applies to modelling of the grain for the luma component. Since film grain in YCbCr video components can be correlated, the autoregressive models for the chroma components have an additional autoregressive coefficient to capture the correlation between the chroma sample grain and of the collocated luma sample grain.

For luma, a 64×64 film grain template is generated. The resolution of the chroma template depends on the chroma format. For example, for YCbCr 4:2:0 video, a chroma film grain template of size 32×32 is generated for each chroma component.

7.4.2.3 Randomization

The randomization block size is equal to 32×32 for luma, and 16×16 for chroma (when in 4:2:0 chroma format). The random offsets for chroma and luma blocks are synchronized, to keep the correlation between the luma and chroma grain.

The pseudo-random number generator used in the algorithm is a shift-back linear-feedback shift register (LFSR, see Figure 19) based on XOR operations, with a length of 16 bits. The corresponding feedback polynomial is $x^{16} + x^{15} + x^{13} + x^4 + 1$. Two offsets for the film grain blocks are generated using eight most significant bits on the register. The chroma offsets are from 0 to 15, while luma offsets are equal to the chroma offsets multiplied by two. The pseudo-random number generator is initialized in the beginning of each 32×32 luma block row to enable parallel processing.

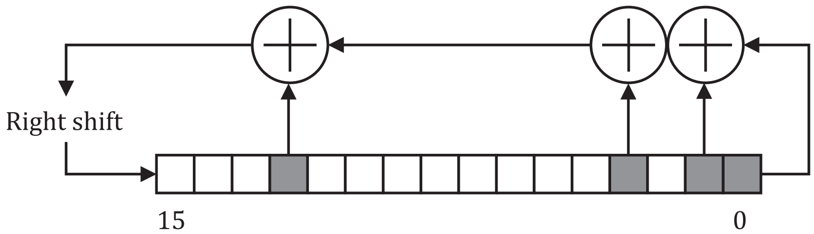


Figure 19 — LFSR used in AFGS1

7.4.2.4 Local adaptation

The model used does not allow changing the grain shape within a picture (generation of only one template is allowed, which is defined by one set of autoregressive coefficients per frame), thus only local adaptation of grain strength is needed. When adding film grain to the Y component, the following model is used:

$$Y' = Y + f(Y) * G_L \tag{2}$$

where Y' is the luma re-noised with film grain, Y is the reconstructed value of luma (before adding film grain), and G_L is the luma film grain sample. $f(Y)$ is a function that scales the film grain based on the value of the luma (since the film grain typically depends on the intensity of the signal, the luma film grain is modelled as a function of the signal intensity). $f(Y)$ is represented with the piecewise-linear function that, as with stepwise scaling function from subclause 7.3.1.4, can be implemented as a pre-computed LUT. For all bit depths, the LUT takes 256 values, and for bit-depths higher than 8, the values between the LUT entries are obtained with linear interpolation. Up to 14 pairs (pivot points) can be signalled to represent the scaling function for the Y component. One illustration of such scaling function is given in Figure 20 where f is a scaling function and I is intensity level.

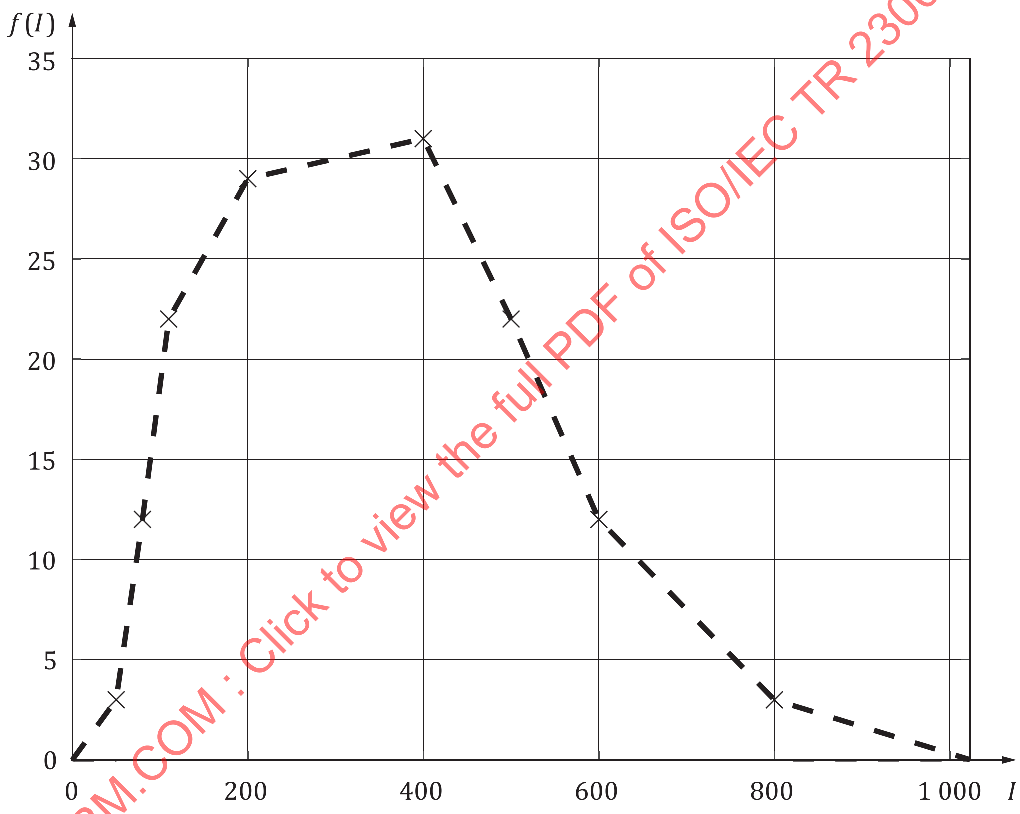


Figure 20 — An example of scaling function used for local grain amplitude adaptation in AFGS1 model. It is defined with the pivot points (x marks). The values in between pivot points are interpolated.

For a chroma component (e.g. Cb), the film grain is scaled using Formulae (3) and (4):

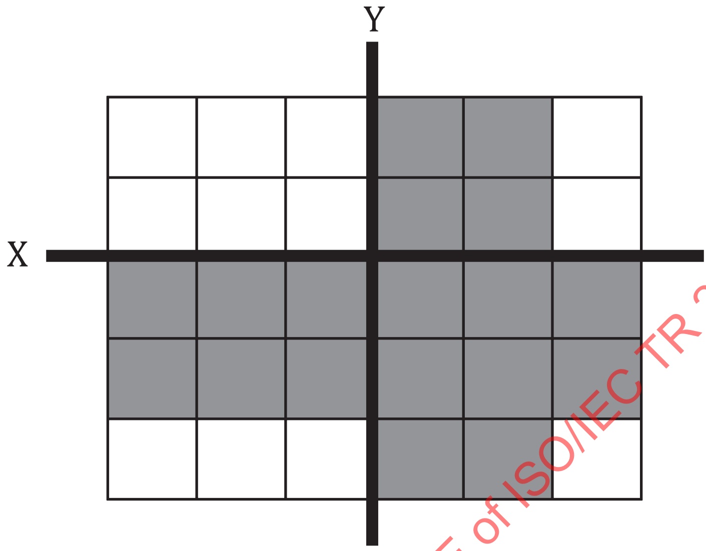
$$u = b_{Cb} * Cb + d_{Cb} * Y_{av} + h_{Cb}, \tag{3}$$

$$Cb' = Cb + f(u) * G_{Cb}, \tag{4}$$

where $f(u)$ is a function that scales the film grain, u is the index corresponding to a Cb component scaling function, and G_{Cb} is the film grain sample for the Cb component. The index u depends on both the chroma and luma component values for the sample, and parameters b_{Cb}, d_{Cb}, and h_{Cb} are signalled in the SEI message. Y_{av} is the average luma corresponding to the chroma sample, taken from one line of samples. For 4:2:0 YCbCr format, $Y_{av} = (Y_1 + Y_2 + 1) >> 1$, where Y_1 and Y_2 are neighbouring (collocated) luma samples located on an even line (numbering starts from 0).

7.4.2.5 Deblocking

There is an option to use overlap between the film grain values at the 32×32 film grain block boundaries. The overlap can be used to ensure that possible artefacts at the film grain block boundaries are attenuated. The overlap is applied before scaling the grain samples and adding them to the reconstructed blocks. As shown in Figure 21, current block samples overlap only with grain blocks to the right and below[9].



Key

X horizontal block boundary

Y vertical block boundary

[gray box] overlapped samples (gray)

Figure 21 — Overlapped blocks

The overlap between the luma blocks is two samples, and between the chroma blocks (in YCbCr 4:2:0) is one sample. To enable the overlap, the random offsets of overlapped grain values for the above row can be saved before the current row is processed. The operations used in the grain overlap for horizontal block boundaries are as follows:

$$G_{cur}(x, 0) = (27 * G_{up}(x, 32) + 17 * G_{cur}(x, 0) + 16) >> 5 \quad (5)$$

$$G_{cur}(x, 1) = (17 * G_{up}(x, 33) + 27 * G_{cur}(x, 1) + 16) >> 5 \quad (6)$$

where $G_{cur}(x, 0)$ are samples of row 0 of the current block and $G_{up}(x, 32)$ are samples of row 32 of the upper block.

The overlap operation between chroma blocks is done:

$$G_{cur}(x, 0) = (23 * G_{top}(x, 16) + 22 * G_{cur}(x, 0) + 16) >> 5 \quad (7)$$

A similar process is applied for vertical block boundaries.

7.4.2.6 Blending

The generated film grain is consecutively applied for each 32×32 luma block (16×16 chroma block) of reconstructed video samples, in raster scan order, using additive blending.

The following operation for adding grain to the samples of a luma block is used:

$$Y'(x,y) = \text{Clip3}(\ Y(x,y) + (\ (\ G_L(x + s_x, y + s_y) * f(\text{Y}) + 2^{\text{shift}-1}\ ) >> \text{shift}\ ), a, b\ ), \quad (8)$$

where a and b define the legal range, x and y are coordinates inside the block, and the parameter `shift` controls the scaling of the film grain.

7.4.2.7 Additional features

In AFGS1, it is required to provide the film grain model parameters for the decoded picture resolution and colour space. The film grain analysis module can also provide optional film grain model parameters that correspond to picture resolutions and/or colour spaces that are different from the decoded picture resolution and colour space. A film grain synthesis module that supports this optional capability can select the film grain parameters that are close to its display resolution.

7.5 Example of film grain synthesis supporting both the frequency filtering and autoregressive models

7.5.1 General

The “Versatile Film Grain” synthesis software described in Annex B.2.2 supports both frequency filtering and the autoregressive models, with local adaptation of both grain shape and strength, while not requiring line buffers.

7.5.2 Film grain template generation

Depending on the frequency or auto-regressive model, 64×64 templates are generated according to the process described in subclause 7.2.2.2 or 7.2.2.3. For chroma in the 4:2:0 colour format, the template size is 32×32.

For the autoregressive model, a larger template is first created then cropped to 64×64 (or potentially 32×32 for chroma) to leave space for the convergence of the autoregressive filter.

The same underlying Gaussian noise (same random generator initialization) is used to generate all film grain templates to be used in a given picture, so that the transition from one to another is smooth if change happens at the sample level, as discussed in subclause 7.2.4.2.

The maximal number of simultaneous templates supported is configurable, and when more than supported are requested by received parameters, graceful degradation is implemented by restricting to the ones listed first. This requires transmitting the most important film grain characteristics first in an FGC SEI message, which is possible because the ordering of intensity intervals is flexible.

7.5.3 Randomization

The random generator is a 32-bit LFSR, the same as in the SMPTE RDD 5 example but bit-reversed (the right-shifting variant shown in Figure A.3 and discussed in Annex A.2). Similar to SMPTE RDD-5, an array of initialization values is defined.

The randomization block size is equal to 16×16 (for luma). Random offsets and template sign flips for all colour components are derived from different 10-bit bitfields of the 32-bit LFSR register; those bitfields partially overlap.

For a luma 16×16 block, 13 horizontal and 12 vertical positions are allowed. The transition from a 10-bit field to 12 or 13 positions uses the multiplication and shift technique described in Annex A.1, which translates into one (12 = 8 + 4) or two (13 = 8 + 4 + 1) adders in hardware.

7.5.4 Local adaptation

Local adaptation is sample-based. Based on the intensity of the current sample, the appropriate scaling factor and template index is selected. Pattern interpolation is supported.

Performing sample-based adaptation avoids the potential cost of line buffers in a hardware implementation.

7.5.5 Deblocking

Horizontal deblocking uses a three-tap filter, while vertical deblocking uses overlapping, to avoid line buffers.

7.5.6 Blending

Only additive blending is supported, with clipping to the allowable range.

8 Film grain analysis

8.1 General

The film grain analysis process is applied on the encoder side and is a non-normative process regardless of the synthesis method and standard in use. It can be implemented as a pre-processing step or as a part of the encoding process. Ultimately, it provides indicative features of the film grain in accordance with the selected parameterized model and supported metadata. It determines film grain model parameters to be sent in the appropriate metadata mechanism, e.g. in an SEI message, to enable content-aware decoder-side film grain synthesis. The analysis process is not mandatory, and manually tuned parameters can be provided to the encoder.

Figure 22 depicts a simplified framework for film grain analysis. As an input to the process, besides the source video, the denoised representation is also required. Edge and complex texture analysis is performed in order to determine a map of flat and non-flat regions in the scene. This is done since high-frequency components, such as edges and texture, can interfere with the analysis process of the film grain, which also resides in high-frequencies. It is to note that in such approach, the performance of the analysis is highly influenced by the effectiveness of the denoiser and precision of the edge and texture analysis.

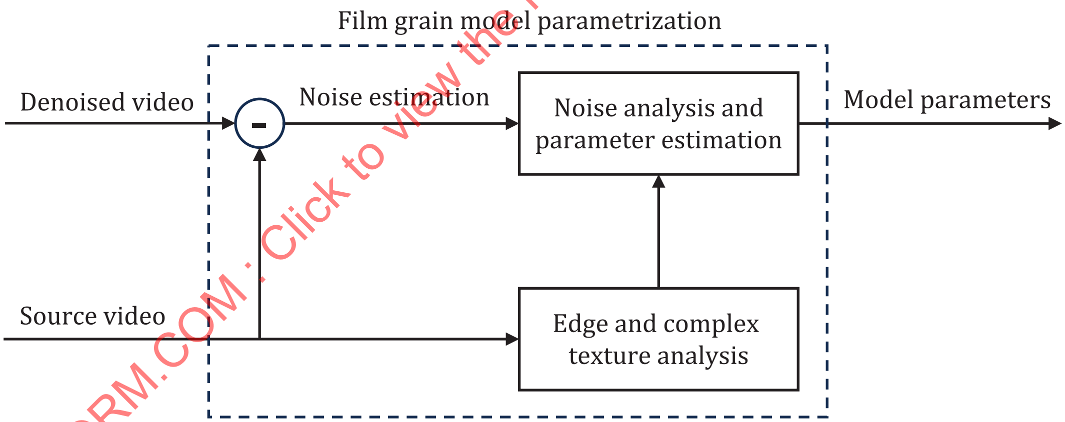


Figure 22 — Film grain analysis and parameter estimation general steps.

The presented generalized model is one common approach to estimate noise and noise parameters. Different approaches are possible, for example, by using appropriate techniques noise can be estimated without the need for a denoised version of the source video. Anyhow, the most commonly used film grain analysis workflow consists of the following commonly implemented steps, divided into two groups.

Film grain model independent steps (independent from film grain parameter format):

a) a pre-processing step to produce a noise estimate (a.k.a. film grain image). It consists of:

    1) denoising the video and

2) finding the difference between source and denoised pairs to get noise estimate (film grain image).

b) a pre-processing step to produce other information about the characteristics of the input video such as edge and texture analysis.

Model specific step:

c) film grain analysis and parameter estimation step to determine model parameters based on the outputs from 1) and 2).

1) Estimate film grain strength as a function of underlying picture intensity.

2) Estimate frequency limits (cut-off frequencies) for frequency based model or auto-regressive coefficients on relevant picture intensity levels.

Film grain analysis and parameter estimation depend on the selected parameterized model. Some examples of a possible implementation are provided in the following subclauses.

8.2 Denoising and image analysis

8.2.1 Denoising

In existing video distribution systems, denoising is a commonly available and widely used process. For example, coding the denoised sequence can significantly improve compression efficiency since uncorrelated noise is removed.[10] Possible solutions for denoising the input video source include the use of motion adaptive, or motion compensated temporal filtering (MCTF) methods. MCTF methods utilize motion search and motion compensation algorithms to exploit temporal redundancies and isolate the video signal from the noise signal, and therefore achieve denoising. Motion adaptive methods perform simple decisions using temporally neighbouring samples to determine noise. Although they are commonly less performing, they usually also have lower complexity, e.g. one variant is given in Reference [11]. Other approaches can be used instead of MCTF. Block-matching and 3D filtering (BM3D) is one of the most popular image denoising algorithm.[12] In some implementations, reconstructed video sequence can be used, reducing the need for the denoised sequence in film grain estimation (depending on the quantization parameter, reconstructed sequence can highly approximate the denoised sequence with most of the details preserved, but with noise removed).

In contrast to traditional analytical denoising approaches, recent advances utilize data-driven deep learning techniques to provide state of the art performance.[11][13][14] However, deep learning approaches usually suffer from excessive computational complexity.

After denoising, film grain images that represent noise estimates are generated by computing the difference between the original input frames and the filtered/denoised frames that correspond to the same time instance.

8.2.2 Edge and texture analysis

Since denoising often alters actual picture details in addition to noise, the film grain image resulting from the difference of source picture and the denoised picture can contain more than film grain in the textured or edge areas. It is better to avoid those areas and estimate film grain parameters on flat (textureless) areas.

One possibility could be to detect flat blocks using gradient-based features with thresholds that are more lenient to allow for correct grain modelling in extreme cases, for example, as proposed in Reference [15]. In this method, for each block in the denoised image, horizontal and vertical gradients are computed and stored in g_x and g_y vectors, then their 2×2 covariance matrix C is computed, and its features are compared with thresholds for the block to be accepted as flat enough: the ratio of eigenvalues (indicative of texture directionality), their sum (indicative of texture strength), and the norm of C.

Another approach could involve well established edge detection algorithms such as Canny edge detector in conjunction with morphological operations. To create a map of flat regions, one approach can be to perform analysis on three scales using the source sequence: the original and two subsampled resolutions, one by a factor of 2 and another by a factor of 4 horizontally and vertically. At each scale, a Canny edge detector is applied (typically without the classical Canny detector blurring step) to determine edges. Morphological operations are applied afterwards at each scale to extend the influence of the edges and to have higher confidence that the edges do not jeopardize the process of estimating the film grain. The obtained maps at the subsampled scales can then be upsampled to the original resolution and combined to form a final map. Additional morphological operations can be subsequently applied to the final map. As an illustration, the VTM[6] implementation applies four-pass dilations at original scale, three-pass dilatation on subsampled scale by factor 2, and two-pass dilations on subsampled scale by factor 4. After upsampling and combining all scales in one flat region map at full resolution, two-pass dilations followed by one-pass erosion are applied.

The film grain image and map of flat regions are then provided to the noise analysis & parameter estimation module.

Additionally, a planar trend removal can be applied.[16] In this implementation, planar trends in the selected flat blocks are removed by fitting a plane, $h(x, y) = a * x + b * y + c$, to the block, and by computing the trend-removed noise patch as $N'(i, j) = N(i, j) - h(i, j)$.

8.3 Determination of grain scaling function

8.3.1 General

Since film grain is dependent on the local characteristics of an image, different amplitude of film grain can be applied for different intensities or intensity intervals of an input image in order to get film grain of appropriate strength before finally blending it to the input image. Thus, initial film grain pattern is scaled to the proper intensity based on the scaling factor. Multiplication of a pattern by the scaling factor determines the level at which the film grain will be perceived at the final image, and by doing that it is ensured that the film grain is simulated at the correct scale.

The relationship between local image intensity and grain amplitude, called here scaling function, can be determined by analyzing the film grain image, the filtered image, and the flat-region map produced during pre-processing. Scaling function, for example, can be represented as polynomial function, or lower-degree approximation of a (polynomial) scaling function can be considered depending on the implementation. For example, piece-wise linear function, e.g. Figure 20, or stepwise constant scaling function, e.g. as illustrated in Figure 17, can be considered implementations.

The analysis is typically performed on a grid of non-overlapping bocks, that belong to a flat region, as determined by the edge and texture analysis stage. For each block, two features are computed: the first one is the average intensity of the block in the denoised image, and the second one is the noise level of the block in the film grain image. It leads to the set of pairs, here called observation points, that are used in further analysis.

Scaling function can be obtained by fitting a polynomial function to the observation points. Some additional processing of the observation points can be required before fitting function, for example, to remove outlier and to improve estimated function precision.

8.3.2 An example of FGC SEI message scaling factor estimation

8.3.2.1 General

This example is taken from the VTM reference code,[6] where image is analyzed by blocks of different size depending on resolution: 8×8 for HD and below, 16×16 up to UHD, and 32×32 above. The further process, that operates on observation points collected in previous step, is described as follows.

8.3.2.2 Data points regularization and discarding potential outliers

Data points regularization is used to regularize high variations and to control the excessively fluctuating points. The following regularization function can be used:

```
k=3.0;
m=0.5;
```

```
tmp = k * noise_level^m + 0.5;
noise_level = Floor( tmp );
```

Coefficient m defines regularization of dispersity/heteroscedasticity of observation points.

Coefficient k defines the level of film grain to be added back, it is aligned to the level of the film grain being removed from the source (during the filtering and compression).

After regularization is applied, the observation points with noise_level higher than 16 << (bitDepth − 8) are discarded from the calculation to limit observation points to meaningful values. If noise_level is higher than the maximum defined value, there is a possibility that noise estimation is biased by edges and other high frequency components that are removed during the filtering.

8.3.2.3 Curve fitting and curve quantization

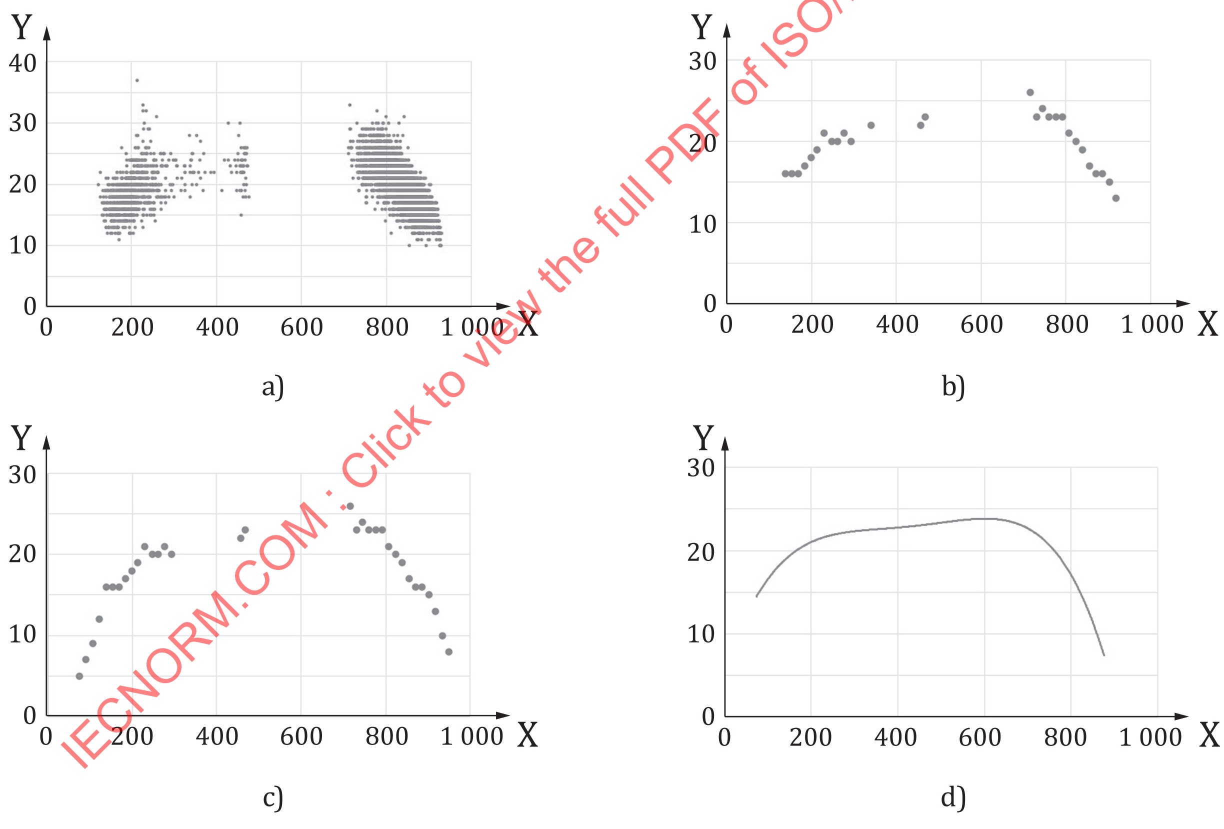
8.3.2.3.1 General

The observation points are used to determine scaling factors (grain strength) in two steps:

— derive a scaling function by two-pass curve fitting

— quantize the fitted curve to produce a stepwise function

The process is also illustrated in Figure 23.



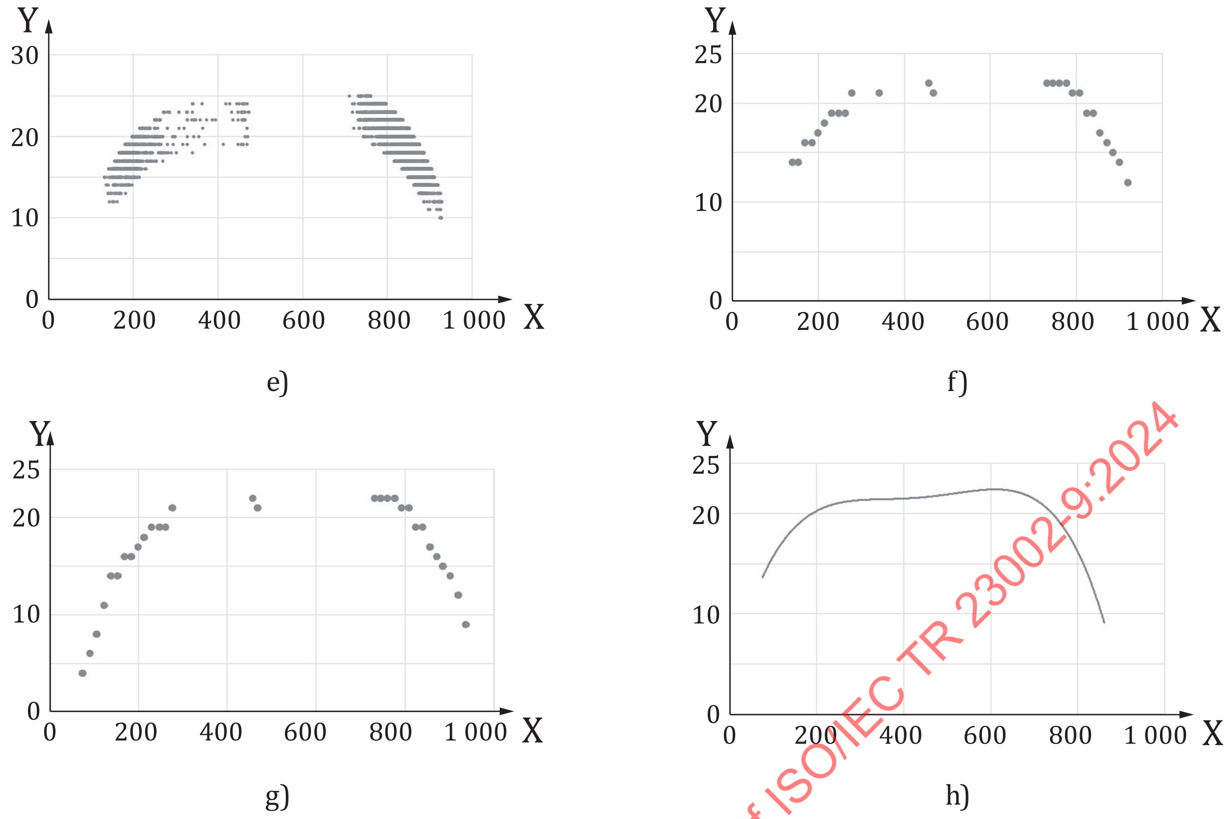


Figure 23 — Two-pass curve fitting process

8.3.2.3.2 Sub-range averaging

On Figure 23 a) observation points are already regularized and limited to maximum value. Complete intensity dynamic range (horizontal axis) is divided (uniformly quantized) into non-overlapping intervals of size 16. Observation points are then grouped per intervals. For each intensity sub-range, the average value of the variance is calculated only if minimum number of points within the sub-interval is $N_{min} > 8$. It is illustrated on Figure 23 b).

8.3.2.3.3 Discarding potential outliers (1st pass)

If there is single point without any neighbouring points, it is considered as an estimation error and it is removed from further calculations. Single points are filtered out, see the missing point in Figure 23 c) compared to Figure 23 b). This step is highly useful in corner cases, for example, if a single point appears at an intensity range where film grain is not present or at least its strength is expected to be low.

8.3.2.3.4 Extreme points extrapolation

A step of extension of points to the left and to the right towards the zero is then applied to smooth transition from intensities with film grain to the intensities without film grain. The process finds the most left and most right point and extends the data-point range, e.g. see added points in Figure 23 c). At most 4 new points are added to the left and 4 new points to the right.

8.3.2.3.5 Discarding potential outliers (2nd pass)

Also remove points if intensity (horizontal axis) is less than 40 or larger than 950 (for 10-bit signal; these values are shifted for bit depths other than 10). Indeed, film grain is usually not added to the very dark and very bright regions.

8.3.2.3.6 Curve fitting (1st pass)

Next step is to fit the parametrized curve by using fourth-order polynomial fitting, Figure 23 d).

8.3.2.3.7 Discarding potential outliers (3rd pass)

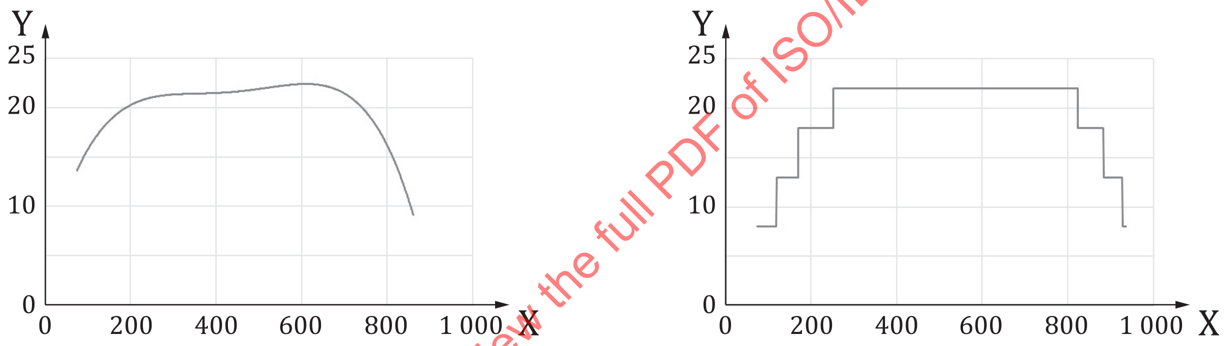
Bounds of +0.6 times the standard deviation and −1.2 times the standard deviation around the first fitted curve is used to filter out the points outside the given range. This step filters out remaining outliers and biased variance points.

8.3.2.3.8 Curve fitting (2nd pass)

The remaining points are entering second pass of curve fitting. Previous processes are repeated (subclause 8.3.2.3.2 to 8.3.2.3.6 illustrated on Figure 23 f) to h) and again the polynomial curve is fitted, but this time using reduced set of observation points. The final scaling function is illustrated on Figure 23 h).

8.3.2.4 Final scaling function approximation

The final step illustrated in Figure 24 approximates the scaling function represented by the fourth-order polynomial curve by a simplified stepwise scaling function. The stepwise function is derived by using Lloyd-max non-uniform quantization with four quantization levels. The quantizer is adapted (trained) for each new set of points on-the-fly.



a) Fitted curve

b) Quantized curve

Figure 24 — Fitted curve and quantized curve

The use of stepwise scaling function leads to several intensity intervals and scaling factors as it is illustrated Figure 24 b). Note, stepwise scaling function is intrinsic to the FGC SEI message design. Currently, film grain SEI message as defined in VSEI and implemented in the VTM[6] comprises the following parameters that define film grain scaling function, which is applied at the decoder/synthesis side:

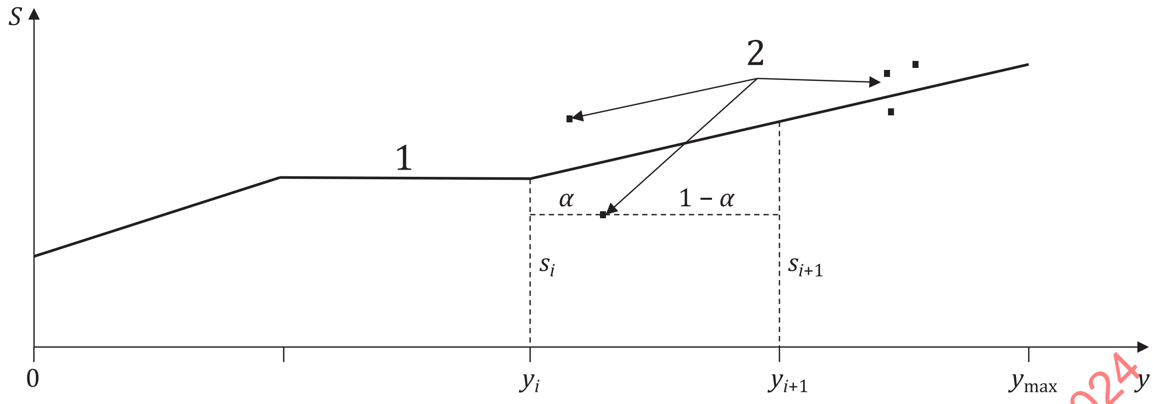
— fg_intensity_interval_upper_bound[c][i] less than fg_intensity_interval_lower_bound[c][i+1]

— fg_comp_model_value[c][i][0],

where fg_comp_model_value[c][i][0] represents the scaling factor, i is index of the interval, and c is colour component. Note that each step of a given stepwise function represents one intensity interval, defined with its bounds (lower and upper bound) and a scaling factor value.

Analysis is performed in the same way for all colour components.

8.3.3 An example of AFGS1 scaling factor estimation



Key

1 fitted curve

2 observations(Y_k, b_k)

Figure 25 — Illustration of the observation points (Y_k, b_k) containing average block intensity

Figure 25 is an illustration of the how observation points (Y_k , b_k) containing average block intensity, Y_k, and the measured noise scale, b_k, are used as constraints when solving for the noise strength values s_i, which are a function of block intensity.[16]

The result of the flat block estimation gives a number of tuples relating the block intensity, *Y,* to the noise level within the block b_k. A piecewise representation of the noise-strength mapping is assumed. Let $\boldsymbol{s} = [s_1, s_2, ..., s_n]$ be the values representing the piecewise linear function, equally sampled on the [0, Y_{max}] domain. Each of the observations, (Y_k , b_k), provides a constraint on the noise strength table. In practice, n << 2^8, so the constraints are applied on interpolated values between adjacent points (Figure 25). Letting i_k denote the largest index of the lookup table below Y_k, we can represent the piecewise constraint on the two adjacent points as:

$$(1 - \alpha) * s_{ik} + \alpha * s_{ik+1} \quad (9)$$

These formulae for all (Y_k , b_k) are stacked into a linear system, which can be solved for ***s***. In order to account for some variables having no or little constraints, a regularization term is added to smooth across lookup table bins that are lacking data. This can happen when all the flat-blocks are identified in low-intensity regions, as there will be no constraints on the higher intensity range. With regularization, the following objective is minimized:

$$\min_s |A * s - b|^2 + | s |^2 \quad (10)$$

The parameter β is scaled proportionally to the number of observations used to construct A. The solution to the regularized problem is the solution to the following system of formulae:

$$A^T * A * s - \beta * \nabla^2 s = A^T * b \quad (11)$$

8.4 Determination of cut-off frequencies for frequency filtering model

8.4.1 General

Cut-off frequencies can be estimated as follows. The film grain image (noise estimate) is scanned block by block, using non-overlapping blocks grid, where block size (denoted as N) depends on the specific implementation. Only blocks within the flat part of film grain image are processed further. For each block within a flat region, a forward transform (usually DCT-2) is applied. Then, each coefficient resulting from the transform is squared. Afterwards, an average squared transformed block B_{avg} is computed over all available squared transformed blocks B_i (sample-wise) as:

$$B_{avg}(x,y) = \frac{1}{K} * \sum_{i=0}^{K-1} B_i(x,y) \tag{12}$$

where K is the total number of blocks that is used for the calculations. Thereafter, the average of columns B_c and of rows B_r vectors are calculated (the DC component for the first row and the first column is discarded from the computation) as follows:

$$B_c(y) = \frac{1}{N-1+(y>0)?1:0} * \sum_{i=(y>0)?0:1}^{N-1} B_{avg}(i,y) \tag{13}$$

$$B_r(x) = \frac{1}{N-1+(x>0)?1:0} * \sum_{i=(x>0)?0:1}^{N-1} B_{avg}(x,i) \tag{14}$$

The average vectors B_c and B_r are regularized to suppress peaks. The average vectors are represented as a curve and its intersection points with a total average value avg of a block B_{avg} is computed: $avg = \frac{1}{N*N} * \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} B_{avg}(i,j)$. The value of avg is therefore subsequently used as a threshold for other computations.

Based on the analysis of the intersection point(s), cut-off frequencies are obtained. The appropriate intersection points are chosen to represent the cut-off frequencies of the frequency-filtered film grain model.

This process can be repeated to estimate frequency limits for different intensity intervals, as is allowed by FGC SEI and could be required to accurately model the film grain, since in a photographic process, its spatial frequency highly depends on exposure.

The scaling parameters signalled for each intensity interval depend on the noise level estimated in the previous step, but also need to factor in the frequency limits, since on synthesis side, different frequency limits lead to different grain amplitudes.

If no intersection points are found, film grain is not present in the input frame. If the analysis process determines that no grain is present in the source video, the appropriate syntax element values could be set to indicate that synthesis will not be performed at the decoder side.

8.4.2 An example of FGC SEI message cut-off frequency estimation

The following method provides an example according to the VTM[6] implementation, which is based on SMPTE RDD 5 specification with some additional modifications, as described in subclause 7.3.2.1 and subclause 7.3.2.2.

The illustrated implementation limits filtering to low pass filtering as defined by SMPTE RDD 5, even though the FGC SEI message supports band pass filtering.

An implementation of the FGC SEI message parameter estimation that conforms to the FGC SEI message semantics in VSEI uses 16×16 arrays to define film grain patches. However, the methods in the VTM[6] and subclause 7.3.2 of this document use 64×64 arrays. Thus, 64×64 DCT-2 as defined in VVC can be used within analysis process or as described in SMPTE RDD 5.

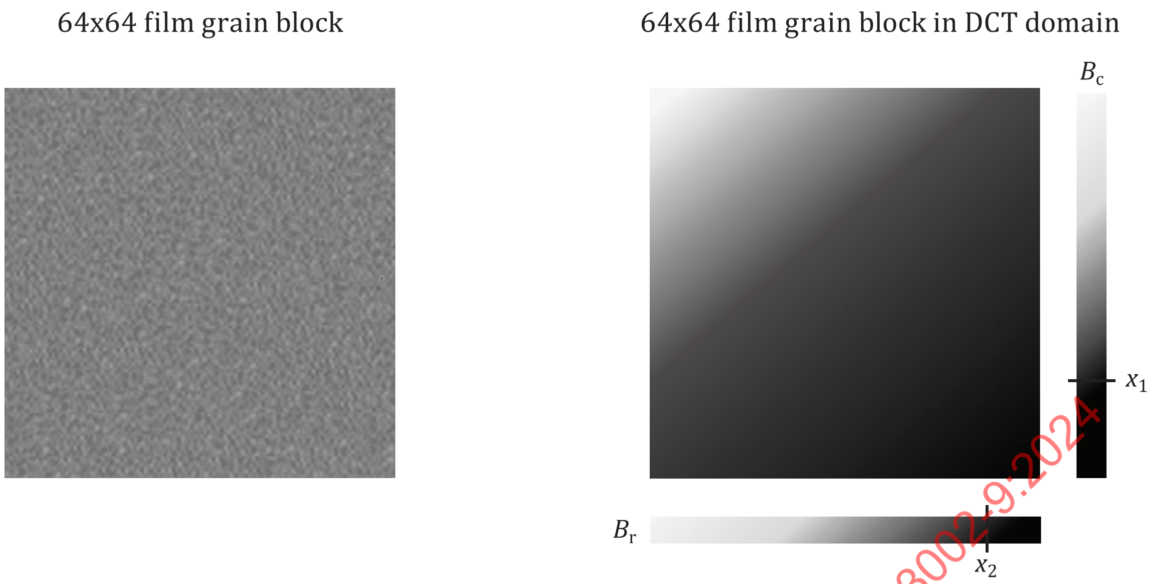


Figure 26 — Cut-off frequency estimation (intersection points x_1 and x_2).

Figure 26 illustrates the process of horizontal and vertical cut-off frequency estimation on a 64×64 array, following the process described in subclause 8.4.1 to estimate x_1 and x_2 points where Bc (column power average) resp. Br (row power average) intersect a specific threshold. Based on estimated x_1 and x_2, the model values are set to (according to the SMPTE RDD 5 specification scaling to 16×16 arrays is needed):

```
comp_model_value[ c ][ i ][ 1 ] = Clip3(2,14,(x1-1)>>2)         (15)
comp_model_value[ c ][ i ][ 2 ] = Clip3(2,14,(x2-1)>>2)         (16)
```

where *i* is the index of the interval, and *c* is the colour component index. If no intersection points are found, film grain is not present in the input frame.

8.5 Determination of autoregressive model coefficients

In one method to estimate the autoregressive coefficients, the model described in subclause 7.4.2 is used, and the coefficients $a_{i,j}$ that best approximate the grain image in the flat regions are chosen.

Either they are determined by least squares error minimization using all samples from grain image in flat regions, or by estimating the auto-correlations necessary to build the Yule-Walker equations.

To estimate the autoregressive coefficients, the coefficients $a_{i,j}$ from Formula (1) are chosen that best approximate the grain image in the flat regions.

For the FGC SEI message autoregressive model described in subclause 7.4.1, the coefficients $a_{i,j}$ are actually optimized by selecting appropriate values for the parameters p_i and A (see Figure 18). Formula (1) can be expressed from p_i and A for this purpose and optimization can be conducted separately for different intensity intervals that can have different values of p_i and A.

For the AFGS1 autoregressive model described in subclause 7.4.2, coefficients $a_{i,j}$ are independent and can be optimized directly. More details on the AFGS1 and the estimation of the autoregressive model parameters can be obtained from Reference [16].

9 Film grain metadata

9.1 General

VVC, HEVC, and AVC natively support signalling of film grain parameter values using well defined supplemental enhancement information (SEI) message. AV1 signals the film grain synthesis parameters as part of the bitstream since film grain synthesis is mandatory in AV1. In addition, AFGS1 provides a mechanism for signalling film grain parameters as ITU-T T.35 user data registered metadata, which is supported by most video coding standards.

VVC, HEVC, and AVC support the film grain characteristics (FGC) SEI message for indicating the usage of film grain synthesis when decoding and rendering video or image data. The FGC SEI message is capable of indicating two different film grain models and different blending modes. More specifically, this SEI message supports a frequency filtering and an autoregressive film grain model, and additive or multiplicative blending modes. It can also indicate one or more intensity intervals, the film grain variance for each intensity interval, and the spatial frequency characteristics of the film grain for each intensity interval, providing considerable flexibility to content creators and encoding manufacturers on signalling different types and levels of film grain noise.

An alternative film grain synthesis scheme, such as a different use of an autoregressive model, can also be indicated through the use of the ITU-T T.35 registered or unregistered user data SEI messages that are also supported in these standards. VVC, HEVC, and AVC intentionally do not specify any constraints or limitations on the post-processing techniques that can be used with decoded data from such user data SEI messages, therefore enabling more applications and implementations. There is also the possibility of applying a film grain synthesis scheme through the use of external means not signalled within in the video bitstream.

9.2 Film grain characteristics SEI message

9.2.1 General

Film grain metadata for use with VVC, HEVC, and AVC can be signalled via an FGC SEI message specified in the corresponding coding standard. For VVC, the SEI payload Type is specified in Rec. ITU-T H.266 | ISO/IEC 23090-3 Annex D and the syntax and semantics are specified in VSEI. For HEVC, the FGC SEI message is specified in Rec. ITU-T H.265 | ISO/IEC 23008-2. For AVC, the FGC SEI message is specified in Rec. ITU-T H.264 | ISO/IEC 14496-10. The different FGC SEI message versions are similar but have some standard-specific differences. For example, persistence of the FGC SEI message is specified differently for AVC than it is for HEVC and VVC.

9.2.2 Interpretation of FGC SEI message syntax

A guide to interpretation of key syntax elements in the FGC SEI message in VSEI is shown in Table 2. Similar interpretations apply to the FGC SEI messages specified in Rec. ITU-T H.264 | ISO/IEC 14496-10 (AVC) and Rec. ITU-T H.264 | ISO/IEC 14496-10 (HEVC).

Table 2 — Guide to the FGC SEI message syntax interpretation

Syntax	Range	Significance
fg_model_id	0 or 1	Model to be used in grain synthesis. 0: Frequency filtering, 1: Auto-regression
fg_separate_colour_description_present_flag	0 or 1	Defines whether the colour description for the film grain specified is same as for the coded video sequence.
fg_blending_mode_id	0 or 1	Blending mode used to combine grain and decoded samples.0: Additive, 1: Multiplicative
fg_comp_model_present_flag	0 or 1	Defines the presence of film grain model parameters for each colour component
fg_num_intensity_intervals_minus1	0 to 255	Defines the number of intensity intervals for each colour component
fg_num_model_values_minus1	0 to 5	Specifies the number of component model values available in the SEI (default values will be used for the remaining component model values)
fg_intensity_interval_lower_bound	0 to 255	Lower bound for each intensity intervals for which the model is applicable

Table 2 *(continued)*

Syntax	Range	Significance
fg_intensity_interval_upper_bound	0 to 255	Upper bound for each intensity intervals for which the model is applicable
fg_comp_model_value[c][i][j]		Component model values have different meaning depending on the value of fg_model_id
fg_characteristics_persistence_flag	0 or 1	Indicates the persistence of the FGC SEI message.

The component model values specify the strength, shape, density, and other characteristics of the film grain. A unique set of component model values can be signalled for each intensity interval for each colour component to be processed. The value range of each syntax element can be constrained by specific practice or implementation.

When the frequency-filtering film grain model is signalled (fg_model_id = 0), values of fg_comp_model_value[c][i][j] are interpreted as follows for each colour component, c, and intensity interval, i.

— fg_comp_model_value[c][i][0] : the standard deviation of Gaussian noise.

— fg_comp_model_value[c][i][1]: horizontal high cutoff frequency.

— fg_comp_model_value[c][i][2]: vertical high cutoff frequency.

— fg_comp_model_value[c][i][3]: horizontal low cutoff frequency.

— fg_comp_model_value[c][i][4]: vertical low cutoff frequency.

— fg_comp_model_value[c][i][5]: correlation between consecutive colour components.

fg_comp_model_value[c][i][j] is in the range of 0 to $2^{\text{fgBitDepth}[\,c\,]} - 1$, inclusive. The derivation of fgBitDepth[c] value is specified in VSEI.

When the autoregressive film grain model is signalled (fg_model_id = 1), values of fg_comp_model_value[c][i][j] are interpreted as follows.

— fg_comp_model_value[c][i][0] : the standard deviation of Gaussian noise.

— fg_comp_model_value[c][i][1]: first order correlation for neighbouring samples (x − 1, y) and (x, y − 1).

— fg_comp_model_value[c][i][2]: correlation between consecutive colour components.

— fg_comp_model_value[c][i][3]: first order correlation for neighbouring samples (x − 1, y − 1) and (x + 1, y − 1).

— fg_comp_model_value[c][i][4]: aspect ratio of the modelled grain.

— fg_comp_model_value[c][i][5]: second order correlation for neighbouring samples (x − 2, y) and (x, y − 2).

fg_comp_model_value[c][i][j] is in the range of $-2^{(\text{fgBitDepth}[\,c\,] - 1)}$ to $2^{(\text{fgBitDepth}[\,c\,] - 1)} - 1$, inclusive. The derivation of fgBitDepth[c] value is specified in VSEI.

The frequency-filtering and autoregressive models for film grain synthesis share the following processing steps:

— Determination of applicable intensity intervals for each sample for each applicable colour component. A different set of FGS model parameters can be signalled in the FGC SEI message for each intensity interval.

— Generation of synthesized grain.

— Blending of synthesized grain and decoded image.

The methods for determining intensity intervals and generating synthesized grain depend on the signalled FGS model. Methods for the frequency-filtering model are described in subclause 7.3. Methods for the autoregressive model are described in subclause 7.4.

fg_comp_model_value[c][i][0] together with intensity interval boundaries is used to define a scaling function (piece-wise constant scaling function). The scaling function indicates the level at which the film grain will be perceived in the final output frame.

Additive (fg_blending_mode_id = 0) and multiplicative (fg_blending_mode_id = 1) grain blending methods are specified in VSEI.

9.3 AFGS1 metadata

9.3.1 General

AOMedia Film Grain Synthesis 1 (AFGS1)[8] is an autoregressive film grain synthesis model that can be indicated in the VVC, HEVC, and AVC standards using user data registered by Recommendation ITU-T T.35 SEI messages. The AFGS1 model is equivalent to the AV1[17] film grain synthesis algorithm on the picture level. See subclause 7.4.2.7 for a description of the additional resolution and colour space features supported in the AFGS1 syntax.

The following subclauses provide interpretation of the syntax for this film grain synthesis model.

9.3.2 Interpretation of AFGS1 metadata syntax

A guide to interpretation of the syntax in the AFGS1 metadata is shown in Table 3.

Table 3 — Guide to the AFGS1 metadata interpretation

Parameter	Range	Significance
apply_grain_flag	0 or 1	specifies whether film grain is added to this frame: 0: not applied, 1: applied
grain_seed	0 to 65535	specifies the starting value for the pseudo-random number register used during film grain synthesis.
film_grain_param_set_idx	0 to 7	an index of a film grain parameter set. Up to 8 parameter sets can be simultaneously stored.
update_grain_flag	0 or 1	1: means that a new set of parameters is sent for the current film_grain_param_set_idx. 0: previous set of parameters in film_grain_param_set_idx is used.
num_y_points	0 to 14	number of points for the piece-wise linear scaling function of the luma component.
point_y_value_increment[i]	0 to 255	increment of x (luma value) coordinate for the i-th point of the piecewise linear scaling function for luma component with respect to point i − 1 for i > 0 and 0 for i = 0 (in case of 10-bit video, these values correspond to luma values divided by 4).
point_y_scaling[i]	0 to 255	scaling (output) value for the i-th point of the piecewise linear scaling function for luma component.
luma_only_flag	0 or 1	1: film grain synthesis process is only applied to the luma component. 0: film grain synthesis process can be applied to the chroma components.
chroma_scaling_from_luma_flag	0 or 1	1: chroma scaling is inferred from the luma scaling. 0: chroma scaling is signalled independently.
num_cb_points	0 to 10	number of points for the piece-wise linear scaling function of the Cb component.

Table 3 *(continued)*

Parameter	Range	Significance
point_cb_value_increment[i]	0 to 255	increment of x coordinate for the i-th point of the piecewise linear scaling function for Cb component with respect to point i − 1 for i > 0 and 0 for i = 0 (in case of 10 bit video, these values correspond to luma values divided by 4).
point_cb_scaling[i]	0 to 255	scaling (output) value for the i-th point of the piecewise linear scaling function for Cb component.
num_cr_points	0 to 10	number of points for the piece-wise linear scaling function of the Cr component.
point_cr_value[i]	0 to 255	x coordinate for the i-th point of the piecewise linear scaling function for Cr component (In case of 10 bit video, these values correspond to luma values divided by 4).
point_cr_scaling[i]	0 to 255	scaling (output) value for the i-th point of the piecewise linear scaling function for Cr component.
grain_scaling_minus_8	0 to 3	represents shift – 8 applied to the values of the chroma component. The parameter determines the range and quantization step of the standard deviation of film grain.
ar_coeff_lag	0 to 3	determines the number of auto-regressive coefficients for luma and chroma.
ar_coeffs_y_plus_128[i]	0 to 255	specifies auto-regressive coefficients used for the Y plane.
ar_coeffs_cb_plus_128[i]	0 to 255	specifies auto-regressive coefficients used for the Cb component.
ar_coeffs_cr_plus_128[i]	0 to 255	specifies auto-regressive coefficients used for the Cb component.
ar_coeff_shift_minus_6	0 to 3	specifies the range of the auto-regressive coefficients. Values of 0, 1, 2, and 3 correspond to the ranges for auto-regressive coefficients of [−2, 2), [−1, 1), [−0.5, 0.5) and [−0.25, 0.25) respectively.
grain_scale_shift	0 to 3	specifies how much the Gaussian random numbers are scaled down during the grain synthesis process.
cb_mult	0 to 255	a multiplier for the Cb component used in derivation of the input index to the Cb component scaling function.
cb_luma_mult	0 to 255	a multiplier for the average luma component used in derivation of the input index to the cb component scaling function.
cb_offset	0 to 511	an offset used in derivation of the input index to the cb component scaling function.
cr_mult	0 to 255	a multiplier for the Cr component used in derivation of the input index to the Cr component scaling function.
cr_luma_mult	0 to 255	a multiplier for the average luma component used in derivation of the input index to the Cr component scaling function.
cr_offset	0 to 511	an offset used in derivation of the input index to the Cr component scaling function.
overlap_flag	0 or 1	1: the overlap between film grain blocks is applied. 0: the overlap between film grain blocks is not applied.
clip_to_restricted_range	0 or 1	1: clipping to the restricted (studio) range is applied to the sample values after adding the film grain 0: clipping to the full range is applied to the sample values after adding the film grain.

The number of luma autoregressive coefficients is determined as:

numPosLuma = 2 * ar_coeff_lag * (ar_coeff_lag + 1)

The number of chroma autoregressive coefficients is typically found as:

numPosChroma = numPosLuma + 1

The last chroma autoregressive coefficient models correlation between the chroma grain sample and a collocated luma grain sample.