Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

File format for storage of scalable video*

BAI Gang†1, SUN Xiao-yan2, WU Feng2, YIN Bao-cai1, LI Shi-peng2

(1Beijing Municipal Multimedia and Intelligent Software Key Laboratory, Beijing University of Technology, Beijing 100022, China)

(2Microsoft Research Asia, Beijing 100086, China)

†E-mail: baigang@emails.bjut.edu.cn

Received Nov. 27, 2005; revision accepted Feb. 21, 2006

Abstract: A file format for storage of scalable video is proposed in this paper. A generic model is presented to enable a codec independent description of scalable video stream. The relationships, especially the dependencies, among sub-streams in a scalable video stream are specified sufficiently and effectively in the proposed model. Complying with the presented scalable video stream model, the file format for scalable video is proposed based on ISO Base Media File Format, which is simple and flexible enough to address the demands of scalable video application as well as the non-scalable ones.

Key words: File format, Scalable video, Video file format
doi:10.1631/jzus.2006.A0706 **Document code:** A **CLC number:** TN919.8

INTRODUCTION

Digital media is becoming an indispensable part of people's daily life thanks to the rapid development and wide adoption of handy digital media capturing devices, rich digital contents, portable media devices and versatile sharing networks. More and more users show greater demands for enjoying digital media services through various PC and non-PC devices over the Internet or wireless networks. Such kind of ubiquitous multimedia services pose great challenges to traditional coding techniques, such as H.264 coding scheme. Responding to the new requirements, the many scalable coding schemes emerging have drawn great attention in both industry and research areas.

During the past decade, many approaches have been developed to achieve scalabilities of video. For example, several layered scalable techniques, namely, SNR scalability, temporal scalability, and spatial scalability, have been presented in MPEG-2 and MPEG-4 (ISO/IEC 13818-2, 1994; ISO/IEC 14496-2, 1998). Especially, MPEG-4 adopted Fine Granularity Scalable (FGS) coding to enable more flexible bandwidth adaptation (Li, 2001). Based on the FGS coding scheme, several improved schemes have been proposed to achieve better coding performance (Wu *et al*., 2001; van der Schaar and Radha, 2002). Moreover, the forthcoming MPEG-21 is also exploring scalable video coding technologies for its future component, where 3D wavelet-based scalable video coding has been extensively investigated (Flierl and Girod, 2003; Xiong *et al*., 2004; Reichel *et al*., 2004).

In general, a scalable video stream generated by scalable video system is composed of one or more concurrent sub-streams which can be classified into two categories: scalable and non-scalable sub-streams, according to whether it can be truncated arbitrarily or not. For instance, the video stream generated by FGS consists of at least two sub-streams, base layer sub-stream and FGS enhancement layer sub-stream. The base layer sub-stream is a non-scalable sub-stream while the FGS enhancement layer is fine granular scalable since it is embedded. Obviously, it is the scalable sub-streams that enable ready adaptation to variance requirements in terms of the process ability, network bandwidth, display capacity and other status

* Project supported by the National Natural Science Foundation of China (No. 60572104) and the Natural Science Foundation (No. 40410031) of Beijing, China. This work has been done while the author is with Microsoft Research Asia

by reshaping manipulations directly on the compressed data.

On the other hand, scalable video stream has created new challenges to video file format. To fit different application scenarios, various combinations of sub-streams need to be effectively produced. Moreover, there are light or heavy dependencies among these sub-streams which should be taken into account during scalable stream reshaping. However the mainstream file formats designed for traditional video content lack power in supporting scalable-related features. The ISO Base Media File Format (BMFF) (ISO/IEC 14496-12, 2004) is a typical one which has been widely accepted for storage and exchange of normal digital media and was adopted by MP4 file format (ISO/IEC 14496-14, 2004) and AVC file format (ISO/IEC 14496-15, 2004).

Due to the new features introduced by scalable coding method, new requirements have been presented to video storage, delivery and experience which cannot be fully and efficiently supported by traditional media file format. Thus, some file formats have been developed for scalable video (Visharam *et al*., 2004; Mukherjee and Said, 2002; Singer and Visharam, 2005). But the scalable video models presented in (Visharam *et al*., 2004) cannot sufficiently describe the dependency among sub-streams and the file format defined in (Mukherjee and Said, 2002) is not backward compatible with ISO BMFF. The dependency among sub-streams is fully but redundantly represented since each sub-stream contains the IDs of all the sub-streams on which it depends either directly or indirectly (Singer and Visharam, 2005). Therefore, a storage file format of scalable video independent of coding method is proposed in this paper to enable the best visibility of and access to the scalable features, and to enhance the opportunities for the interchange and interoperability of scalable video.

The storage file format for scalable video presented in this paper is based on the ISO BMFF which is described in detail in (ISO/IEC 14496-12, 2004). All the features of ISO BMFF, such as object-oriented file structure and so on, are readily inherited. In order to address the demands of both different kinds of scalabilities and variable coding methods, a generic scalable video stream model is first proposed in this paper. Based on the presented model, a new file format is presented to effectively support scalable video application as well as the traditional ones.

The paper is organized as follows. In Section 2, a generic model of scalable video stream is proposed. Section 3 describes the new file format in detail and gives an example. Finally, Section 4 concludes this paper.

GENERIC MODEL OF SCALABLE VIDEO STREAM

In order to develop a codec-independent file format, we need to propose a model of scalable video stream to describe the relationship among sub-streams generally. Before proposing the generic model of scalable video stream, three important concepts, presentation, presentation group and ensemble, on scalable video are introduced in this section first.

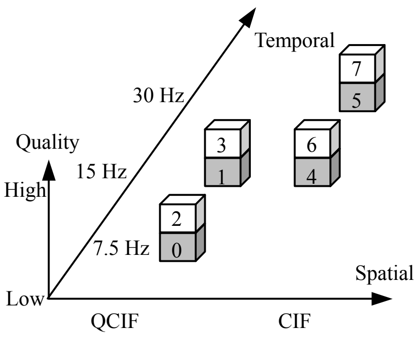
Presentation, presentation group and ensemble

As aforementioned, a scalable video stream consists of one or more scalable and non-scalable sub-streams which can be grouped into different sets to offer different presentation of the video. In this paper, each valid generation of a scalable video presented to user or application for consumption is called a presentation of the scalable video. Moreover, a set of sub-streams that offers a valid presentation is called a presentation group. The key feature of a presentation is that all the sub-streams in a presentation group combined together can be decoded to generate an acceptable underlying exhibition. Each presentation is associated with a set of sub-streams among which certain dependencies existed.

In addition, to describe a scalable video more precisely, we introduce another concept in this paper: ensemble. An ensemble is a collection of sub-streams associated with a single video such that any presentation group of the video is either fully inside the ensemble or completely outside the ensemble. It is disallowed that a presentation group contains sub-streams from different ensembles. In other words, each sub-stream of the scalable video stream can only belong to one ensemble. Furthermore, an ensemble should be created by a single encoder and decoded by a single decoder.

Here, we use a scalable video stream generated by Scalable Video Model (SVM) 3.0 (Reichel *et al*.,

2004) as an example to further clarify the meaning of presentation and ensemble. As shown in Fig.1a, the exemplified scalable video stream contains eight sub-streams (numbered from 0 to 7) with different frame rates (say 7.5 fps, 15 fps or 30 fps), bit rates (say from 32 kbps to 256 kbps) and resolutions (CIF or QCIF). In Fig.1b, we illustrate some presentations provided by different sub-stream sets. For instance, the presentation group II is composed of sub-stream 0 and sub-stream 1 to provide a presentation at QCIF-15 fps-41 kbps.



(a)

Present group	Sub-streams	Frame size	Frame rate (fps)	Bit rate (kbps)
I	0	QCIF	7.5	32
II	0,1	QCIF	15	41
III	0,2	QCIF	7.5	66
IV	0,1,2,3	QCIF	15	80
V	0,1,2,3,4	CIF	15	88
VI	0,1,2,3,4,5	CIF	30	115
VII	0,1,2,3,4,6	CIF	15	222
VIII	0,1,2,3,4,5,6	CIF	30	256

(b)

Fig.1 Exemplified scalable stream of MPEG SVM. (a) Sub-streams; (b) Presentations and presentation groups

Notice that in scalable video stream, each presentation group may also be scalable. That is, a subset of sub-streams extracted from a presentation group can form another valid presentation. For example, as denoted in Fig.1b, presentation group II and group III can be extracted from presentation group IV. However, an arbitrary set of sub-streams may not be a presentation group. For example, the set of sub-stream 1, sub-stream 2 and sub-stream 3 is not a presentation group since it depends on sub-stream 0 which is a base layer sub-stream to be decodable. In fact, it is at least one of the base layer sub-streams that should be included in every presentation.

On the other hand, a presentation should be fully independent from any other sub-streams out of the presentation group to be decodable. In other words, all the sub-streams needed to provide the presentation should be involved in the corresponding presentation group. For instance, to achieve the presentation II with QCIF-15 fps-41 kbps, only the sub-stream 0 and sub-stream 1 are necessary. As a consequence, when a presentation is chosen for a scalable video, all the sub-streams in the presentation group should be selected. Nevertheless, the data blocks of each scalable sub-stream in the presentation can be dropped or scaled to enable further adaptation.

As mentioned before, it is forbidden that a presentation group contains sub-streams from different ensembles. Therefore, all the sub-streams of the exemplified scalable stream belong to one ensemble because that presentation group VIII contains all the sub-streams.

DAG model of scalable video stream

Regarding the two concepts described above, the sub-streams together with the relationships inherited among the sub-streams is abstracted into a directed acyclic graph (DAG). A DAG G is composed of a set of directed edges $\{V\}$ and a set of nodes $\{E\}$, i.e., $G=\{V, E\}$, $V=\{v_0, v_1, \ldots, v_N\}$, $E=\{e_0, e_1, \ldots, e_M\}$ and $v_k=(e_i, e_j)$, if and only if e_i has directed edges point to e_j (Jiang, 2001).

An exemplified DAG G is shown in Fig.2, which reflects the sub-streams and the relationships among the sub-streams illustrated in Fig.1. Here $G=\{V, E\}$,

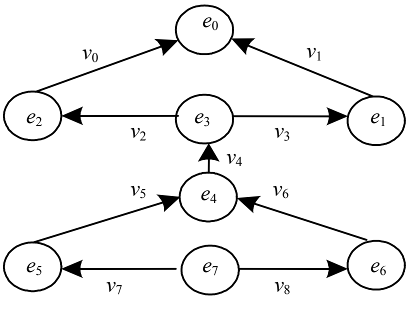


Fig.2 The DAG G of the scalable stream shown in Fig.1

$V=\{v_0, v_1, \ldots, v_8\}$, $E=\{e_0, e_1, \ldots, e_7\}$. Each node e_i of graph G indicates a sub-stream i in the scalable video stream together with its property (e.g., frame rate, size, quality, and so on). The property of one node implies that the sub-stream combined with its dependent sub-streams can provide a presentation complying with the specified property. The dependencies among sub-streams are described by edges in DAG. There is no directed edge starting and ending at the same node in the DAG model because it does not make sense for a sub-stream.

To unambiguously and effectively describe the dependencies among sub-streams, some constrains are presented in forming the DAG model. First, we limit the dependency described in the model to be the direct dependency only. That is, two nodes in graph G are connected by a directed edge if and only if the two sub-streams have direct dependency determined by the scalable video generation method. As shown in Fig.2, there is a directed edge pointing from e_2 to e_0 in graph G which means that sub-stream 2 directly depends on sub-stream 0. In contrast, sub-stream 3 actually depends on sub-stream 0, in addition to sub-stream 1 and sub-stream 2, in decoding, though there is no edge between e_3 and e_0 due to the absence of direct dependency. Second, no matter directly or indirectly, if e_i is dependent on e_j, then e_i will belong to DAG G only if e_j is inside too. Finally, it is only one node that has no output edge in a DAG G, which is regarded as the base layer in the scalable video. If a DAG model has more than one base layer node, we strongly recommend separating it into two independent DAGs. Although it causes a bit overhead, it will significantly reduce the complexity of the scalable stream model. Due to the aforementioned constraints, it can be seen that each graph G can readily represent an ensemble of the scalable stream.

On the other hand, the presentations of scalable video stream are not explicitly described in the proposed model, but can be automatically generated in the model. To achieve a presentation meeting certain requirements on frame rate, size, quality and so on, a node in the DAG with the closest properties are selected first. Then, by searching the DAG from the selected node, a sub-graph will be given for the presentation. All the nodes inside the sub-graph compose the corresponding presentation group. For example, if a presentation of CIF-30 fps-115 kbps is required, then the node e_5 in Fig.2 is selected adaptively. Starting from node e_5, we get a sub-graph G_1 from G by searching the paths from e_5 to e_0, which are defined as $G_1=\{V_1, E_1\}$, $V_1=\{v_0, v_1, v_2, v_3, v_4, v_5\}$ and $E_1=\{e_0, e_1, e_2, e_3, e_4, e_5\}$. Consequently, the presentation group VI shown in Fig.1b is successfully achieved.

Furthermore, as the proper presentation group is selected, all sub-streams included in the group are able to be appended or removed from the presentation group according to the dependency, except the base layer. In each sub-graph G_i, the base layer is the mandatory data and should be included in every presentation.

FILE FORMAT OF SCALABLE VIDEO

Utilizing the generic DAG model of scalable stream presented in Section 2, a file format based on ISO BMFF is proposed for scalable video stream in this section. It fully takes advantages of the ISO BMFF. In other words, the scalable video stream will be stored using the existing features provided by ISO BMFF, such as sample, track, media information, media data container and so on.

Compliant with the ISO BMFF, the whole scalable video stream is managed by a single track and each sub-stream inside the scalable stream is presented by layering structure. Assume that the scalable video stream is composed of a sequence of contiguous Network Abstraction Layer (NAL) units. Similar with ISO BMFF, the NAL units that are to be processed at the same instant in time shall constitute a sample. It means that if two or more NAL units have the same time stamp, then they should be in the same sample. Moreover, in a sample, the NAL units belonging to the same sub-stream form a layered sample. Then all the layered samples of the scalable stream are classified into different layers by Sample to Group Box (“sbgp”) defined in ISO BMFF. Generally, the layered samples belonging to the same sub-stream are classed into the same layer. An example of layered sample, sample and layer is presented in Fig.3. Notice that in the proposed file format, layered sample rather than sample is utilized as the basic unit to be processed. Thus, the Sample to Group Box provided by ISO BMFF is modified to categorize the layered sample instead of sample.

On the other hand, in the proposed file format, new boxes are introduced to support the new features of scalable video based on the DAG model. First, the box named as SVC Layer Description Entry is presented to describe the properties and stream dependency of different layers. Then the box, Layered Sample Information box, is proposed to depict the size information on each layered sample. In the following, the definition of the boxes are given in the syntax description language (SDL) defined in MPEG-4 (ISO/IEC 14496-2, 1998).

SVC Layer Description Entries

BoxTypes: "svcl";
Container: Sample Group Description Box ("sgdb") (Reichel *et al*., 2004);
Mandatory: No;
Quantity: Zero or more.

1. Syntax

```
aligned(8) class SVCLayerEntry() extends VisualSampleGroupEntry ("svcl") {
    unsigned int(8)      layerNumber;
    unsigned int(16)     avgBitRate;
    unsigned int(16)     avgFrameRate;
    unsigned int(32)     width;
    unsigned int(32)     height;
    unsigned int(8)      dependencyCount;
    int i;
    for (i=0; i<=dependencyCount; i++) {
        unsigned int(32) dependent_layerNumber;
        unsigned int(32) dependent_type;
        }
}
```

2. Semantics

layerNumber is a non-negative integer which indicates the number of a layer, with the base layer being numbered as one and all enhancement layers being numbered as two or higher.

avgBitRate gives the bit rate that the layer combined with other depended layers can present.

avgFrameRate gives the frame rate that the layer combined with other depended layers can present. The first byte describes the fractional part with unit 1/256 and the second byte describes the integer part.

width gives the width of picture that the layer combined with other depended layers can present.

height gives the height of picture that the layer combined with other dependent layers can present.

dependencyCount gives the number of layers that the current layer directly depends on.

dependent_layerNumber is the number of the layer that the current layer directly depends on.

dependent_type specifies the dependency type of the layer indicated by *dependent_layerNumber*. Details are shown in Table 1.

Table 1 Dependent type

Dependent-type	Spec. of the dependent-type
"unel"	Unknown enhance layer
"teel"	Temporal enhance layer
"spel"	Spatial enhance layer
"quel"	Quality enhance layer

Layered Sample Information box

BoxTypes: "lsif";
Container: Sample Table Box ("stbl") (Reichel *et al*., 2004);
Mandatory: No;
Quantity: Zero or more.

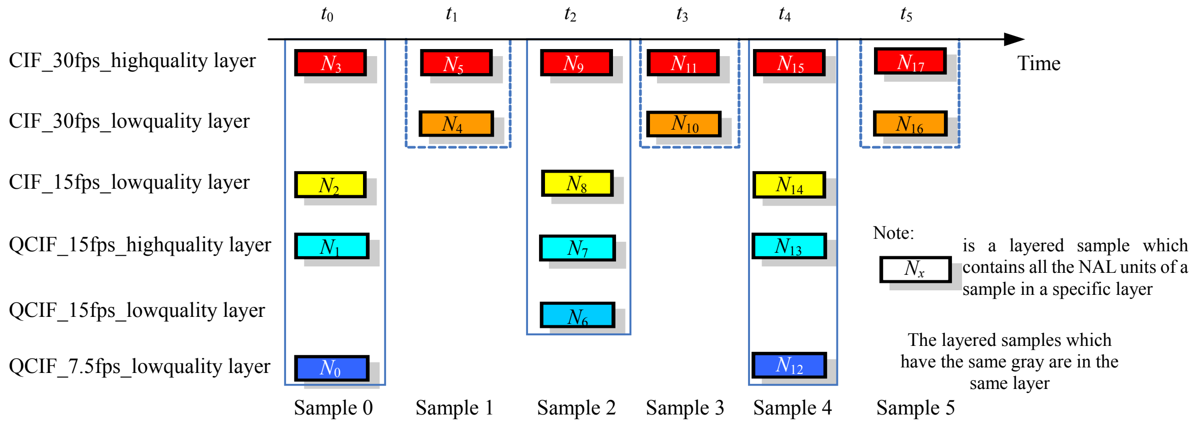

t_0
t_1
t_2
t_3
t_4
t_5
Time
CIF_30fps_highquality layer
CIF_30fps_lowquality layer
CIF_15fps_lowquality layer
QCIF_15fps_highquality layer
QCIF_15fps_lowquality layer
QCIF_7.5fps_lowquality layer
N_3
N_5
N_9
N_{11}
N_{15}
N_{17}
N_4
N_{10}
N_{16}
N_2
N_8
N_{14}
N_1
N_7
N_{13}
N_6
N_0
N_{12}
Note:
N_x
is a layered sample which contains all the NAL units of a sample in a specific layer
The layered samples which have the same gray are in the same layer
Sample 0
Sample 1
Sample 2
Sample 3
Sample 4
Sample 5

Fig.3 An example of layered sample, sample and layer

1. Syntax

```
aligned(8) class LayeredSampleInfoBox extends
 FullBox(“lsif”, version=0, 0) {
  unsigned int(24)  reserved=0;
  unisgned int(8)  field_size;
  int i,j;
  for(i=0; i<sample_count; i++) {
     unsigned int(16) layered_sample_count;
      for(j=0; j<layered_sample_count; j++) {
         unsigned int(field_size)  entry_size;
      }
    }
  }
```

2. Semantics

version is an integer that specifies the version of this box.

field_size is an integer specifying the size in bytes of the entries in the following table; it shall take the value 1, 2 or 4.

sample_count is an integer that gives the number of samples in the track, which can be found in Sample Description Box (“stsd”) defined by ISO BMFF.

layered_sample_count is an integer that gives the number of layered samples in a sample.

entry_size is an integer specifying the size of a layered sample, indexed by its number in a sample.

Example of the proposed file format

The scalable video stream shown in Fig.1 is used as an instance in this subsection to illustrate the proposed file format. The dependency among sub-streams is described in Fig.2. According to the definition given before, the setting of the SVC Layer Description Entries is shown in Table 2.

In case a presentation with CIF-30 fps-115 kbps is required, the complied file parser is utilized to find the best matching layer, *layer*6, in the SVC Layer Description Entries and go through the DAG generated from the SVC Layer Description Entries to result in a set of layers, that is *L*={*layer*6, *layer*5, *layer*4, *layer*3, *layer*2, *layer*1}. With the help of the Track Box (“trak”) (ISO/IEC 14496-12, 2004) provided in ISO BMFF, the file parser can find the position of any sample and get the information, such as index, size and position of the layered samples in a sample, from Layered Sample Information Box. Moreover, the Sample to Group box (“sbgp”) (ISO/IEC 14496-12, 2004) will tell whether the layered sample belongs to the set of candidate layers *L* or not. Finally, the data of selected layered samples will be sent to decoder to make the presentation available.

We have extended the tools of MPEG4IP-1.2 to create and parse the new format file proposed in this paper. The scalable video stream generated by SVM3.0 is used to test the file format. It can be verified that the proposed file format can successfully support scalable-related features of the exemplified scalable video stream as well as the non-scalable video streams. This file format has been proposed in (Bai *et al*., 2005) to MPEG.

CONCLUSION

In this paper, a file format for storage of scalable video is proposed. A generic model of scalable video stream presented by directed acyclic graph is first proposed to enable a codec-independent description of scalable video stream. The relationships among sub-streams are specified sufficiently and effectively in the proposed model. Then, by taking advantages

Table 2 Setting of the exemplified scalable stream shown in Fig.1

layerNumber	*avgBitRate*	*avgFrameRate*	*width*	*height*	*dependencyCount*	*dependent_layerNumber*	*dependent_type*
1	32	0x8007	176	144	0	–	–
2	41	0x000F	176	144	1	1	“teel”
3	66	0x8007	176	144	1	1	“quel”
4	80	0x000F	176	144	2	2	“quel”
						3	“teel”
5	88	0x000F	352	288	1	4	“spel”
6	115	0x001E	352	288	1	5	“teel”
7	222	0x000F	352	288	1	5	“quel”
8	256	0x001E	352	288	2	6	“quel”
						7	“teel”

from the presented scalable video stream model, the file format for scalable video is proposed based on ISO Base Media File Format. It can readily support the desired features introduced by scalable video streams as well as by non-scalable ones.

Nonetheless, more studies need to be done on the efficient truncation method of the scalable sub-streams to get fine granularity. In future work, we will enable the truncation characteristic, including the information of the rate-distortion in case of different type of dependency and the corresponding reshaping method, of the file format.

References

Bai, G., Sun, X.Y., Wu, F., Li, S., 2005. The Proposed Extension of the ISO Base Media File Format for Supporting SVC Content. ISO/IEC JTC1/SC29/WG11 M11664.

Flierl, M., Girod, B., 2003. Video Coding with Motion-Compensated Lifted Wavelet Transforms. Proc. of PCS, p.59-62.

ISO/IEC 13818-2, 1994. Generic Coding of Moving Pictures and Associated Audio, Part-2: Video, I.

ISO/IEC 14496-2, 1998. Coding of Audio-Visual Objects, Part-2: Visual.

ISO/IEC 14496-12, 2004. Information Technology—Coding of Audio-Visual Objects—Part 12: ISO Base Media File Format.

ISO/IEC 14496-14, 2004. Information Technology—Coding of Audio-Visual Objects—Part 14: MP4 File Format.

ISO/IEC 14496-15, 2004. Information Technology—Coding of Audio-Visual Objects—Part 15: Advanced Video Coding (AVC) File Format.

Jiang, C.H., 2001. Graph Theory and Networks Flow. China Foresty Publishing House, Beijing (in Chinese).

Li, W., 2001. Overview of fine granularity scalability in MPEG-4 video standard. *IEEE Trans. Circuits Syst. Video Technol.*, **11**(3):301-317. [doi:10.1109/76.911157]

Mukherjee, D., Said, A., 2002. Structured Content Independent Scalable Meta-formats (SCISM) for Media Type Agnostic Transcoding. Http://www.hpl.hp.com/techreports/2002/HPL-2002-166R1.pdf.

Reichel, J., Wien, M., Schwarz, H., 2004. Scalable Video Model 3.0. ISO/IEC JTC1/SC29/WG11 N6716.

Singer, D., Visharam, M.Z., 2005. VM Study Text for Scalable Video Coding (SVC) File Format. ISO/IEC JTC1/SC29/WG11 N7856.

van der Schaar, M., Radha, H., 2002. Adaptive motion-compensation fine-granular-scalability (AMC-FGS) for wireless video. *IEEE Trans. Circuits Syst. Video Technol.*, **12**(6):360-371. [doi:10.1109/TCSVT.2002.800319]

Visharam, M.Z., Tabatabai, A., Singer, D., 2004. Supporting the Storage of MPEG-21: Part 13, Scalable Video by an Extension of the ISO Base Media File Format. ISO/IEC JTC1/SC29/WG11 M11422.

Wu, F., Li, S., Zhang, Y.Q., 2001. A framework for efficient progressive fine granularity scalable video coding. *IEEE Trans. Circuits Syst. Video Technol.*, **11**(3):332-344. [doi:10.1109/76.911159]

Xiong, R.Q., Wu, F., Xu, J.Z., Li, S.P., Zhang, Y.Q., 2004. Barbell Lifting Wavelet Transform for Highly Scalable Video Coding. Picture Coding Symposium. San Francisco, CA, USA.