ARCHIVED 2011

SMPTE 258M-2004
Revision of
ANSI/SMPTE 258M-1993

SMPTE STANDARD

for Television — Transfer of Edit Decision Lists

1 Scope

This standard describes the data format for the interchange of a list of audio/video content decisions which specify an audio/video product. Exchange of this information allows the audio/video product to be reproduced on a compatible editing system. Such a list is commonly referred to as an edit decision list. The transfer medium and storage format are not specified in this standard.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

ANSI X3.4-1986 (R1992), Information Processing — Coded Character Set - 7-Bit American National Standard Code for Information Interchange (7-Bit ASCII)

SMPTE 12M-1999, Television, Audio and Film — Time and Control

SMPTE RP 162-1993, Storage of Edit Decision Lists on 3-1/2 in Disks

3 Definitions

alphanumeric: The term refers to any alphabetic character or the digits 0 through 9.

ASCII: American National Standard Code for Information Interchange — Coded Character Set — 7-Bit, ANSI X3.4, which describes the character encoding used by this practice.

auto-assembly: Recording one or more edits automatically under system control from data in an edit decision list.

drop frame time code: A time code numbering system which omits frame numbers to compensate for the variance between nominal fields per second and actual fields per second. For NTSC (field rate approximately 59.94 fields per second), frames 0 and 1 are omitted at the start of each minute, except minutes 0, 10, 20, 30, 40, and 50. See also non-drop frame time code.

edit: One or more interrelated elements treated as a unit.

edit decision list: One or more edits.

Copyright © 2004 by THE SOCIETY OF
MOTION PICTURE AND TELEVISION ENGINEERS
595 W. Hartsdale Ave., White Plains, NY 10607
(914) 761-1100

Approved
April 6, 2004

editing mode: Specification of whether or not video is to be recorded on this edit and which audio tracks, if any, are to be recorded.

effect: An audio/video action which is used to combine two or more sources into a single source.

element: A series of printable ASCII characters, terminating in the ASCII characters 'CR' 'LF.' Elements contain information specifying audio/video content decisions in an audio/video product.

field: A syntactical unit within an element that contains a functional grouping of data. Fields are delineated within an element by separators.

hexadecimal: A notation for representing numbers in base 16. Each 8-bit byte is represented by two ASCII characters. The first character represents the 4 high-order bits, and the second character represents the 4 low-order bits. The range of values of each of the 4-bit nibbles is 0 to 15, which is represented by the ASCII characters '0' through '9' for values 0 through 9, and ASCII characters 'A' through 'F' for values 10 through 15.

mix: A method of combining two or more sources to produce a single source.

nondrop frame time code: A time code numbering system in which the number of frames per second is constant. See also ''drop frame time code.''

separator: Character(s) used to delineate fields within an element.

source: A reference to video or audio program material.

submaster: A submaster is a medium, other than the master record medium, on which material is recorded for later use as source material. A typical use of a submaster would be the creation of a multiple mix effect on a system with only one mix effect bank.

virtual edit: Specification of an effect that is not to be recorded directly, but can be referenced later as a source.

virtual source reference: A reference to a virtual edit defined earlier in the edit decision list.

4 Notation

The syntax of the edit decision list format is specified using a notation commonly used in the software industry called BNF (Backus-Naur form). The use of this notation allows precise, yet easy to understand, syntactic specification of the edit decision list format.

A BNF grammar consists of a list of rules which define grammatical terms. The first term defined is typically the highest level construct in the grammar, in this case, an EDL. As there are many variations of the notation known as BNF, the following is a formal definition of the notation used in this practice:

– Grammatical terms are capitalized, with an underscore (_) separating words.

– Grammatical rules consist of: the name of the grammatical term being defined, followed by a colon (:), followed by a list of grammatical terms, BNF operators, and/or literals, and terminated with a semi-colon (;).

– Literal strings are enclosed within double-quotes (").

– Literal characters are enclosed by single-quotes (').

– Nonprintable characters are represented by their name as shown in annex A.

– Comments begin with ‘‘/*’’ and end with ‘‘*/.’’

– The logical OR operation is specified with a vertical bar (|).

– Use of square brackets ([and]) indicates that zero or one occurrence of the items in the brackets is allowed.

– Use of braces ({ and }) indicates that one or more occurrences of the items in the braces are allowed.

The complete grammar is defined in annex B.

5 Edit decision list structure

5.1 General information

The information in an edit decision list (EDL) consists of 8-bit data bytes. The low-order 7 bits are interpreted as characters following the ASCII character set. The high-order bit is ignored. No distinction is made between uppercase and lowercase alphabetic characters when interpreting source elements, trigger elements, device control elements, playback elements, and system directives. Uppercase and lowercase alphabetic characters are considered distinct in comment elements and user elements.

The EDL consists of one or more SMPTE_DIRECTIVEs, followed by a FRAME_RATE_DIRECTIVE, followed by one or more EDIT_SEQUENCEs or PLAYBACK_SEQUENCEs. Both EDIT_SEQUENCEs and PLAYBACK_SEQUENCEs consist of one or more ELEMENTs.

The end of an edit decision list is indicated by either:

– the end of the storage medium;
– a CONTROL-Z (SUB) character.

The physical order of edits on the transfer medium determines the order in which the edits are to be executed. Note that this may result in later edits recording over portions of prior edits.

5.2 Elements

There are eight types of elements:

– INITIAL_DIRECTIVE;
– SYSTEM_DIRECTIVE;
– COMMENT_ELEMENT;
– USER_ELEMENT;
– TRIGGER_ELEMENT;
– DEVICE_CONTROL_ELEMENT;
– SOURCE_ELEMENT;
– PLAYBACK_ELEMENT.

An element consists of a PRINTABLE_STRING, as defined by the grammar in annex B. Each element is terminated by a TERMINATOR (’CR’ ’LF’). The basic unit of information in an element is the field, consisting of one or more characters. A field is delineated from the following field by a SEPARATOR. If a TERMINATOR (’CR’ ’LF’) is preceded by a backslash (’\’), the TERMINATOR is treated as a SEPARATOR.

Zero or more separators, followed by a TERMINATOR, is legal but has no semantic meaning in an EDIT_LIST.

Each element, except for INITIAL_DIRECTIVEs and SYSTEM_DIRECTIVEs, begins with an EDIT_NUMBER. The element type is determined by an optional element type character immediately following the EDIT_NUMBER field. The absence of this character indicates that the ELEMENT is a SOURCE_ELEMENT. INITIAL_DIRECTIVEs and SYSTEM_DIRECTIVEs are easily identified, as they are the only elements that do not begin with an EDIT_NUMBER.

5.2.1 Initial directives

The following directives must occur at the start of the EDL, and cannot be used later in the EDL:

```
SMPTE_DIRECTIVE:
    ''SMPTE'' SEPARATOR ''STD'' NUMBER ''M''
  | ''SMPTE'' SEPARATOR ''STD'' NUMBER ''M'' '-' NUMBER
  | ''SMPTE'' SEPARATOR ''RP'' NUMBER
  | ''SMPTE'' SEPARATOR ''RP'' NUMBER '-' NUMBER
  | ''SMPTE'' SEPARATOR ''EG'' NUMBER
  | ''SMPTE'' SEPARATOR ''EG'' NUMBER '-' NUMBER
        /* Indicates that this list is compatible with the specified standard, recommended practice, or
        engineering guideline. At least one of these directives must be used at the start of the list to identify
        the documents that the list conforms to. */
;

FRAME_RATE_DIRECTIVE:
  ''TIME_CODE_MODULUS'' SEPARATOR NUMBER
        /* Specifies the nominal number of frames per second for time code arithmetic. One and only one
        shall immediately follow the SMPTE directive(s) at the head of the list. */
;
```

5.2.2 System directive

A SYSTEM_DIRECTIVE does not have an EDIT_NUMBER. Its purpose is to provide additional information to the editing system, or to the operator, for edit(s) subsequent to the directive. SYSTEM_DIRECTIVEs are described in clause 6.

5.2.3 Element type

The element type character indicates the type of ELEMENT. Element type identifiers are as follows:

- '*' – COMMENT_ELEMENT;
- '?' – USER_ELEMENT;
- '#' – TRIGGER_ELEMENT;
- '%' – DEVICE_CONTROL_ELEMENT;
- '' – SOURCE_ELEMENT;
- '=' – PLAYBACK_ELEMENT.

The absence of this character indicates a SOURCE_ELEMENT.

5.2.4 Edit sequence

An EDIT_SEQUENCE consists of one or more of the following element types, each terminated by a TERMINATOR:

- – COMMENT_ELEMENT;
- – USER_ELEMENT;
- – TRIGGER_ELEMENT;

– DEVICE_CONTROL_ELEMENT;
– SOURCE_ELEMENT.

An EDIT_SEQUENCE need not contain all five types, but must contain at least one.

All ELEMENTs of an EDIT_SEQUENCE have the same EDIT_NUMBER. The end of an EDIT_SEQUENCE is indicated by a change in EDIT_NUMBER on the next ELEMENT, or by the end of the EDL.

5.2.5 Playback sequence

A PLAYBACK_SEQUENCE consists of one or more PLAYBACK_ELEMENTs, each terminated by a TERMINATOR. It is used to specify dynamic source motion in a manner which allows speed variation to be precisely duplicated.

6 System directive specification

System directives are as follows:

```
SYSTEM_DIRECTIVE:
    INFORMATIONAL_DIRECTIVE
  | AUTO_ASSEMBLE_DIRECTIVE
  | CONTROL_DIRECTIVE
  | MISCELLANEOUS_DIRECTIVE
  | MOTION_DIRECTIVE
;

INFORMATIONAL_DIRECTIVE:
    '*' PRINTABLE_STRING
        /* List Comment (Information for the Operator) */
  | ''TITLE'' SEPARATOR PRINTABLE_STRING
        /* Indicates that all subsequent edits until another TITLE directive are to be associated with the
        specified title string. */
;

AUTO_ASSEMBLE_DIRECTIVE:
    ''WAIT''
        /* Stop auto-assembly when the following edit is encountered. */
  | ''SKIP"
        /* Do not perform the following edit. */
  | ''BELL''
        /* Sound an audible indicator before performing the following edit.*/
;

CONTROL_DIRECTIVE:
    ''RECORD'' SEPARATOR MODE_FIELD SEPARATOR MEDIUM_IDENTIFIER '=' TIME_CODE
        /* Identifies a reel or other medium upon which subsequent edits will be recorded. The video and/or
        audio tracks specified in both this mode field and in a subsequent edit's mode field will be recorded
        for that edit. The TIME_CODE indicates the offset to be added to the sync entry and sync exit times
        of each edit to determine that edit's record times for this record medium. Note that more than one
        record directive can be active at any time. */

  | ''NORECORD'' SEPARATOR MODE_FIELD SEPARATOR MEDIUM_IDENTIFIER
        /* Terminates any RECORD directive for the specified video and/or audio tracks on the specified
        record medium. */
```

| ‘‘SYNC’’ { SEPARATOR SOURCE_IDENTIFIER ’=’ TIME_CODE }
/* A sync group is established, comprised of the indicated sources. The members of the sync group roll synchronously, based on the time code associated with each source, whenever any member of the sync group rolls. Establishing a sync group cancels any previous master/slave links between the members of the sync group.
If more than one source in a sync group has a SOURCE_ELEMENT in an edit, only the first source specified in the edit will be used to determine synchronization of unspecified members of the sync group. Time code values explicitly stated in a SOURCE_ELEMENT will take precedence over any SYNC relationships for the duration of the given edit. */

| ‘‘NOSYNC’’ { SEPARATOR SOURCE_IDENTIFIER }
/* Removes the indicated sources from all prior sync groups. */

| ‘‘SLAVE’’ SEPARATOR MASTER_SRC_ID ’=’ TIME_CODE { SEPARATOR SLAVE_SRC_ID ’=’ TIME_CODE }
/* Establishes master/slave link(s) between the master source and each indicated slave source. All slaves roll synchronously with the master source, whenever the master source is rolled, based on the time codes in the SLAVE directive. Establishing a master/slave link cancels any previous master/slave links between the indicated sources. Time code values explicitly stated in a SOURCE_ELEMENT will take precedence over any master/slave links for the duration of the given edit. */

| ‘‘NOSLAVE’’ SEPARATOR MASTER_SRC_ID { SEPARATOR SLAVE_SRC_ID }
/* Breaks the master/slave link between the indicated master source, and each listed slave source. */
;

MISCELLANEOUS_DIRECTIVE:
‘‘AUDIO’’ SEPARATOR MODE_AUDIO_SUBFIELD
/* The track(s) indicated are the ones that are used by the MODE_FIELD when all audio is called for. A range of audio tracks can be specified by separating two audio track numbers with a dash (-). */

| ‘‘INCLUDE’’ SEPARATOR FILENAME ’=’ TIME_CODE
/* This directive causes the contents of the specified file to replace the include directive in the edit decision list. The time code specifies an offset which will be added to all sync entry and sync exit times of source elements in the included file. The filename is a PRINTABLE_STRING enclosed in quotes. */

| ‘‘MEDIUM’’ SEPARATOR SOURCE_IDENTIFIER SEPARATOR MEDIA_TYPE
/* Allows specification of the media type for a source. Options include LIVE, TAPE, and OTHER. Each media type can be optionally followed by a printable string to give more details as to media type. */

| ‘‘WIPES’’ SEPARATOR ‘‘SMPTE’’
/* Wipe patterns are defined by figure 1. */

| ‘‘WIPES’’ SEPARATOR PRINTABLE_STRING
/* Wipe patterns are not defined by figure 1. The PRINTABLE_STRING specifies the name of the switcher that the wipe patterns conform to. */
;

MOTION_DIRECTIVE:
‘‘MOTION_CURVE’’ SEPARATOR MOTION_TYPE
/* The motion curve directive indicates how motion changes from one source speed to the next. Motion is made to fit a curve: BEST, SPLINE, or Nth order. Refer to 7.7.4 for details of the meanings of the MOTION_TYPEs. */
;

7 Edits

7.1 Edit number

The first field of an ELEMENT is the EDIT_NUMBER. It consists of up to three subfields, concatenated without separators:

- EDIT_NUMBER_SUBFIELD (required)
- VIRTUAL_EDIT_INDICATOR_SUBFIELD (optional)
- RECORDED_INDICATOR_SUBFIELD (optional)

The EDIT_NUMBER may be preceded by a separator, but embedded separators are not permitted.

The ELEMENT type is determined by an optional element type character immediately following the EDIT_NUMBER field. The absence of this character indicates that the ELEMENT is a SOURCE_ELEMENT.

7.1.1 Edit number subfield

The EDIT_NUMBER_SUBFIELD consists of from one to six numeric characters, followed optionally by an alphabetic character. The range of valid edit numbers is from 1 to 999999Z. The number zero (0) is not a valid edit number. Leading zeroes are insignificant in differentiating edit numbers (i.e., 0001 is equivalent to 1).

7.1.2 Virtual edit indicator subfield

A VIRTUAL_EDIT is indicated by the greater-than character (>) immediately following the EDIT_NUMBER_SUBFIELD. All elements of a VIRTUAL_EDIT must have this indicator.

7.1.2.1 Virtual edits

A virtual edit is not recorded directly but is used as a source in a following edit. It must be used to specify a transition or effect wherein one or more of the sources is itself an effect. The use of a virtual edit is comparable to the use of a submaster.

7.1.3 Recorded indicator subfield

The RECORDED_INDICATOR_SUBFIELD indicates that this SOURCE_ELEMENT has been recorded; i.e., is not to be auto-assembled. The RECORDED_INDICATOR_SUBFIELD consists of an exclamation mark, (!), immediately following the VIRTUAL_EDIT_INDICATOR_SUBFIELD, which may be empty.

7.2 Comment element

The COMMENT_ELEMENT contains information, specific to the edit, which is to be interpreted by the operator and not the editing system. It consists of an edit number field (with a COMMENT_ELEMENT character) followed by a PRINTABLE_STRING. The COMMENT_ELEMENT is terminated by a TERMINATOR.

7.3 User element

The USER_ELEMENT is provided to allow an escape from the edit decision list format as defined by this standard and will not be infringed upon by any subsequent amendments to the standard. A USER_ELEMENT consists of an EDIT_NUMBER_FIELD (with a USER_ELEMENT character) followed by a PRINTABLE_STRING. The USER_ELEMENT is terminated by a TERMINATOR. The format of the PRINTABLE_STRING is left to the user.

7.4 Trigger element

The TRIGGER_ELEMENT provides the ability to tell the editing system to trigger a specified action at a specified time. It consists of the following fields:

– EDIT_NUMBER;
– TRIGGER_IDENTIFIER;
– TIME_CODE_SRC_IDENTIFIER;
– TIME_CODE_TYPE;
– TIME_CODE.

The TRIGGER_ELEMENT edit number field (with a TRIGGER_ELEMENT character) is followed by a TRIGGER_IDENTIFIER. The TRIGGER_IDENTIFIER is an IDENTIFIER with up to 8 characters, which identifies which user-specified action is to be triggered. This is followed by an optional TIME_CODE_SRC_IDENTIFIER which changes the trigger time from a time code in the audio/video product to a time code in the specified source material. If the TIME_CODE_SRC_IDENTIFIER is present, the specified source must appear in a SOURCE_ELEMENT in the current edit.

The following field, TIME_CODE_TYPE, is also optional. If the TIME_CODE_TYPE field contains any of the following character strings, ‘‘ENTRY+,’’ ‘‘ENTRY-,’’ ‘‘EXIT+,’’ or ‘‘EXIT-,’’ the trigger time code is interpreted as an offset relative to the immediately preceding entry or exit time code of the audio/video product (or the specified source) in the text.

If the TIME_CODE_TYPE field contains the characters, ‘‘DUR,’’ the trigger time code is interpreted as the duration (in real-time) from the previous trigger with the same TIME_CODE_SRC_IDENTIFIER. If a TIME_CODE_TYPE of ‘‘DUR’’ is used without a TIME_CODE_SRC_IDENTIFIER, the trigger time code is relative to the previous trigger with no TIME_CODE_SRC_IDENTIFIER. If there is no such preceding trigger in the edit, the duration is relative to the immediately preceding SYNC_ENTRY time code in the text.

If a TIME_CODE_TYPE is specified, the current edit must include at least one SOURCE_ELEMENT.

The final field is a TIME_CODE which is the trigger time, either relative or absolute. Table 1 summarizes the options for the trigger time.

Table 1 – Types of trigger time specifiers

TIME_CODE_SRC_IDENTIFIER	TIME_CODE_TYPE	Resulting trigger time
		Program absolute
	ENTRY +	Program relative entry plus
	ENTRY –	Program relative entry minus
	EXIT +	Program relative exit plus
	EXIT –	Program relative exit minus
	DUR	Program duration
SOURCE_IDENTIFIER		Source absolute
SOURCE_IDENTIFIER	ENTRY +	Source relative entry plus
SOURCE_IDENTIFIER	ENTRY –	Source relative entry minus
SOURCE_IDENTIFIER	EXIT +	Source relative exit plus
SOURCE_IDENTIFIER	EXIT –	Source relative exit minus
SOURCE_IDENTIFIER	DUR	Source duration

7.5 Device control element

The DEVICE_CONTROL_ELEMENT defines message strings and their execution times. It consists of the following fields:

- EDIT_NUMBER (with a DEVICE_CONTROL_ELEMENT character);
- DEVICE_IDENTIFIER_FIELD;
- TIME_CODE_SRC_IDENTIFIER;
- TIME_CODE_TYPE;
- MESSAGE_TIME_FIELD;
- MESSAGE_COUNT_FIELD;
- MESSAGE_FIELD.

The TIME_CODE_SRC_IDENTIFIER and TIME_CODE_TYPE fields have the same syntax and function as the fields with the same names in a TRIGGER_ELEMENT.

7.5.1 Device identifier field

The DEVICE_IDENTIFIER_FIELD contains an identifier which specifies the device to which this message should be sent. The device identifier is a symbolic name assigned by the user. Device identifiers are limited to a maximum of 8 characters.

7.5.2 Message time field

The MESSAGE_TIME_FIELD contains either a time code or the keyword ‘‘SETUP.’’ The ‘‘SETUP’’ keyword indicates that this message should be sent prior to execution of the edit. If this field contains a time code, it specifies when the message should be sent to the device. This time code is interpreted according to the contents of the TIME_CODE_SRC_IDENTIFIER and TIME_CODE_TYPE fields. See table 1 for a summary of the possible options.

7.5.3 Message count field

The MESSAGE_COUNT_FIELD contains the length, in bytes, of the message string which will be sent. The message count is expressed as a hexadecimal number.

7.5.4 Message field

The MESSAGE_FIELD defines the actual message string which will be sent. Each byte in the actual message is represented by encoding it into ASCII using two hexadecimal digits per byte. Separators within the message field will be ignored.

7.6 Source element

The SOURCE_ELEMENT defines the audio/video source, its related parameters, and its placement on the record medium. It consists of the following fields:

- EDIT_NUMBER;
- SOURCE_IDENTIFICATION_FIELD;
- MODE_FIELD;
- EFFECT_FIELD;
- EFFECT_PARAMETER_FIELD;
- SOURCE_ENTRY_FIELD;
- SOURCE_EXIT_FIELD;
- SYNC_ENTRY_FIELD;
- SYNC_EXIT_FIELD.

7.6.1 Source identification field

The SOURCE_IDENTIFICATION_FIELD consists of one to eight alphanumeric characters. A SOURCE_IDENTIFICATION_FIELD may be qualified by a SOURCE_AUDIO_SUBFIELD.

7.6.1.1 Special source identifiers

The following source identifiers have unique significance:

BLACK indicates black video and silent audio;
BARS indicate video color bars and audio test tone.

7.6.1.2 Virtual source identifier

A virtual edit can be referenced as a source by the use of a VIRTUAL_SOURCE_IDENTIFIER, which consists of the EDIT_NUMBER_SUBFIELD of the virtual edit followed by the less-than character (<). The edit number in the virtual source identifier refers to the most immediately preceding virtual edit with that edit number. Each virtual edit must precede any edit which references that virtual edit.

Note that a virtual source identifier can refer to an edit which itself contains virtual source references. Note also that the source times in a SOURCE_ELEMENT containing a virtual source reference refer to the sync times in the edit being referenced. Thus, an important restriction on both the source entry time and the source exit time is that these times are bounded by the sync entry time and sync exit time of the referenced edit.

Example 1 in annex C gives an example of the use of virtual source identifiers.

7.6.1.3 Source audio subfield

The optional SOURCE_AUDIO_SUBFIELD identifies the source audio tracks to be used. It is separated from the source identifier by a decimal point (.). There are two components in a SOURCE_AUDIO_SUBFIELD: audio sources and ranges of audio tracks. Combinations of audio sources and audio ranges can be linked by commas (,).

An audio source can be either a single audio track, or it can be a mix of two or more audio tracks. A single audio track is specified by one or two digits (0 through 9), and can have any value from 1 to 99. An audio mix is specified by two or more audio sources, linked by a plus sign (+). Successive audio tracks to be mixed can be specified by a notation consisting of the smallest track number followed by two plus signs (“++”) and the largest track number. In this case, all audio tracks from the smallest to the largest, inclusive, are to be mixed.

Each track number in the SOURCE_AUDIO_SUBFIELD that is not part of a range specification may be followed by a mix level in parentheses. The mix level is a real number, positive or negative, specifying the gain of that track in dB, where 0 dB represents unity gain, and negative values represent less than unity gain. If a mix level is not specified for a given track, the editing system will not attempt to control the gain for that track.

An AUDIO_RANGE is a notation for denoting successive audio tracks. It consists of the smallest track number followed by a hyphen (-) and the largest track number. All audio tracks from the smallest to the largest, inclusive, are to be used.

The mapping between source audio tracks and record audio tracks is described in 7.6.2.2 which defines the AUDIO_TRACK_SUBFIELD. An example of a SOURCE_AUDIO_SUBFIELD is shown in example 2 of annex C.

7.6.2 Mode field

The MODE_FIELD specifies the editing mode (video and audio track selection) of the record medium, and consists of a MODE_INDICATOR_SUBFIELD, optionally followed by an AUDIO_TRACK_SUBFIELD. If there is no AUDIO_TRACK_SUBFIELD, the audio track specification defaults to all available audio tracks, or, if an AUDIO system directive has been used, the default audio track specification follows the AUDIO system directive.

7.6.2.1 Mode indicator subfield

The MODE_INDICATOR_SUBFIELD consists of one alpha character. The character 'V' indicates video only, 'A' indicates audio only, and 'B' indicates both video and audio.

7.6.2.2 Audio track subfield

The AUDIO_TRACK_SUBFIELD identifies the record audio tracks to be enabled. It immediately follows the MODE_INDICATOR_SUBFIELD. There are two possible components in an AUDIO_TRACK_SUBFIELD: audio track numbers and audio ranges. There can be any combination of audio track numbers and audio ranges linked by commas (,). Note that there can be no spaces in the AUDIO_TRACK_SUBFIELD.

A single audio track is specified by one or two digits (0 through 9), and can have any value from 1 to 99. An AUDIO_RANGE is a notation for denoting successive audio tracks. It consists of the smallest track number followed by a hyphen (-) and the largest track number. All audio tracks from the smallest to the largest, inclusively, are to be used. Two mappings of audio sources to record audio tracks are allowed. If the number of audio tracks specified in the AUDIO_TRACK_SUBFIELD matches the number of audio sources specified in the SOURCE_AUDIO_SUBFIELD, the order of each subfield specifies a one-to-one correspondence of audio source to record track. Note that the track numbers indicated by the AUDIO_TRACK_SUBFIELD must be exclusive; i.e., record track numbers
must not repeat.

The only alternative to a one-to-one mapping is to specify one audio source and more than one record track. In this case, the single audio source is recorded on each of the specified record tracks.

Use of the AUDIO_TRACK_SUBFIELD is shown in example 3 of annex C.

7.6.3 Effect field

The EFFECT_FIELD specifies the switcher transition or effect to be performed. It consists of either a SINGLE_SOURCE_EFFECT or a MULTIPLE_SOURCE_EFFECT.

7.6.3.1 Single source effect

A SINGLE_SOURCE_EFFECT is used for switcher effects which specify the involvement of the source in the audio/video product without dependencies on earlier SOURCE_ELEMENTs. The first character in the SINGLE_SOURCE_EFFECT must be one of the following type identifiers:

C – CUT. Audio/video cut to the new source;

R – SYNC ROLL. Although no switcher effect is indicated by this effect type, the 'R' in the EFFECT_FIELD indicates a source synchronization element. No direct effect on video or audio.

A source synchronization element indicates that the specified source is to be rolled synchronously with the audio/video product. The SOURCE_ENTRY_FIELD and SYNC_ENTRY_FIELD define the synchronous relationship. If an edit consists of no other type of SOURCE_ELEMENT than sync roll elements, the first of these elements specifies the duration of the edit.

7.6.3.2 Multiple source effect

A MULTIPLE_SOURCE_EFFECT is used for switcher effects which specify how the source is combined into the audio/video product as specified by earlier SOURCE_ELEMENTs. A MULTIPLE_SOURCE_EFFECT cannot be used unless earlier SOURCE_ELEMENT(s) in the same EDIT specify the other source(s) for the effect.

An edit may contain at most one DISSOLVE, WIPE, KEY, MATTE, or NONADDITIVE MIX per channel.

The first character in the MULTIPLE_SOURCE_EFFECT must be one of the following type identifiers:

D – DISSOLVE. Video and audio dissolve from the source specified by an earlier SOURCE_ELEMENT (often referred to as the FROM source) to the new source.

W – WIPE. This must be followed by a pattern number. Video wipes from the FROM source to the new source and audio dissolves from the FROM source to the new source.

K – KEY. The associated source indicates the key source (cutter). A key requires earlier SOURCE_ELEMENT(s) to specify the background over which the key will appear. If no foreground source is specified in the following SOURCE_ELEMENT, the foreground is the same as the key source. A pattern number following this type identifier and its modifiers, if any, indicates a wipe key. The key effect does not affect audio and thus the element's MODE_FIELD should be video only.

M – MATTE. The associated source indicates the key source (cutter). A matte requires earlier SOURCE_ELEMENT(s) to specify the background over which the key will appear. The foreground source is a matte (or color) generator. A pattern number following this type identifier and its modifiers, if any, indicates a wipe matte. The matte effect does not affect audio and thus the element's MODE_FIELD should be video only.

F – FOREGROUND FILLER. The associated source indicates the filler for the immediately preceding foreground effect (e.g., fill hole cut by key signal or fill border of a wipe). A filler requires earlier SOURCE_ELEMENTs to specify both the background and the key source. No direct effect on audio.

Q – QUAD SPLIT. One SOURCE_ELEMENT is required for each of the four quadrants, and the optional QUAD_MODIFIERS must be identical on each element. A QUAD_SPECIFIER must follow to identify which quadrant is filled by the video source of this SOURCE_ELEMENT. There is no direct effect on audio.

N – NONADDITIVE MIX. Video does a nonadditive mix, audio dissolves.

X - AUDIO MIX. Specifies a mix of audio channels. This effect type can appear in one or more SOURCE_ELEMENTs, each of which specifies a source to be included in the mix. The effect type is followed by an optional level specifier. The level specifier consists of a REAL_NUMBER enclosed within parentheses, representing a gain, in decibels, to be applied to each of the record audio tracks specified by the AUDIO_TRACK_SUBFIELD.

7.6.3.3 Wipe pattern number codes

An extensive set of basic wipe patterns has been assigned codes. Variations on these basic patterns are available via pattern modifiers which specify horizontal and vertical replication of the basic pattern. The pattern codes have been grouped by basic geometry as follows:

Pattern description	Code range
Combination of horizontal, vertical, and diagonal edges	1 – 99
Enclosed shapes, expanding from the picture's center	101 – 199
Combinations of horizontal and vertical edges rotating about various center points (clock wipes)	201 – 299
Wipes consisting of sequential revelation of picture squares (matrix wipes)	301 – 399

Pattern number zero is reserved to indicate no selection of pattern by the system. Gaps have been left in the numbering scheme to allow for future additions to the table of pattern codes. The basic patterns and their assigned codes are included in figure 1.

7.6.3.4 Pattern modifiers

Wipe patterns may be modified by appending one or more of the modifiers listed below. The modifiers directly follow the pattern number.

R – Reverse direction;
S – Soft edge;
B – Wipe border;
P – Position;
M – Pattern modulator;
A – Shadow;
T – Tumble or revolve pattern;
G – Spot light. The presence of this modifier indicates a wipe between the FROM source at reduced signal level and the TO source at full signal level;
H INTEGER – Horizontal replication factor. Specifies multiples of the basic pattern;
V INTEGER – Vertical replication factor. Specifies multiples of the basic pattern;
C – Checkerboard wipe. First pass creates a checkerboard, second pass fills it in (matrix wipes only);
X – eXternal wipe code, switcher specific. Specifies that this wipe pattern does not conform to the recommended wipe patterns. Note that many patterns not included in the basic set of patterns in figure 1 can be specified using horizontal and vertical replication factors.

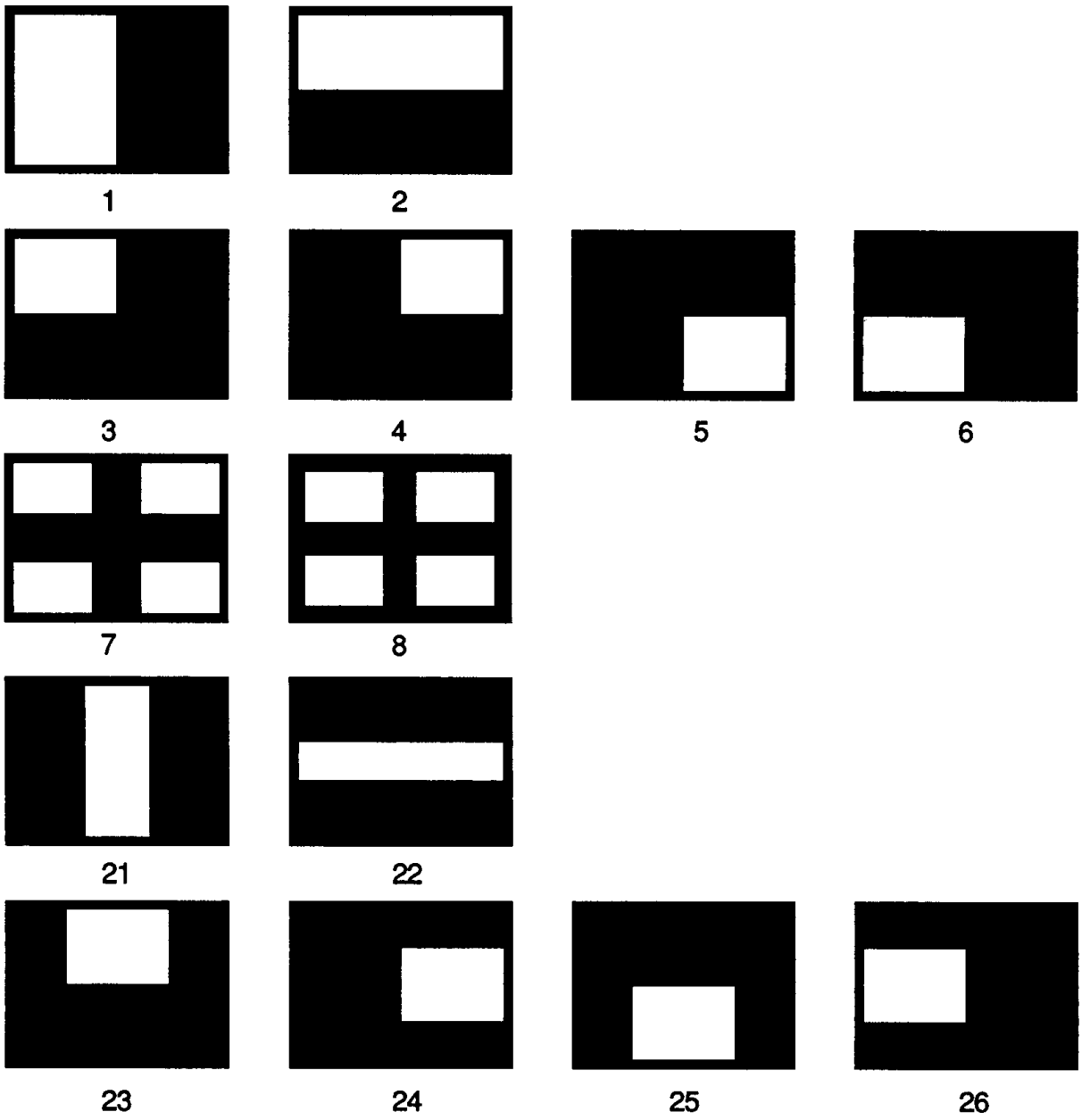
7.6.3.5 Key or matte modifiers

These modifiers must directly follow the key or matte type identifier.

H – Shadow;
O – Outline;
E – Edge (border);
I – Invert (inverts foreground and background);
T – Soft edge between foreground and background;
C – Chroma-key.

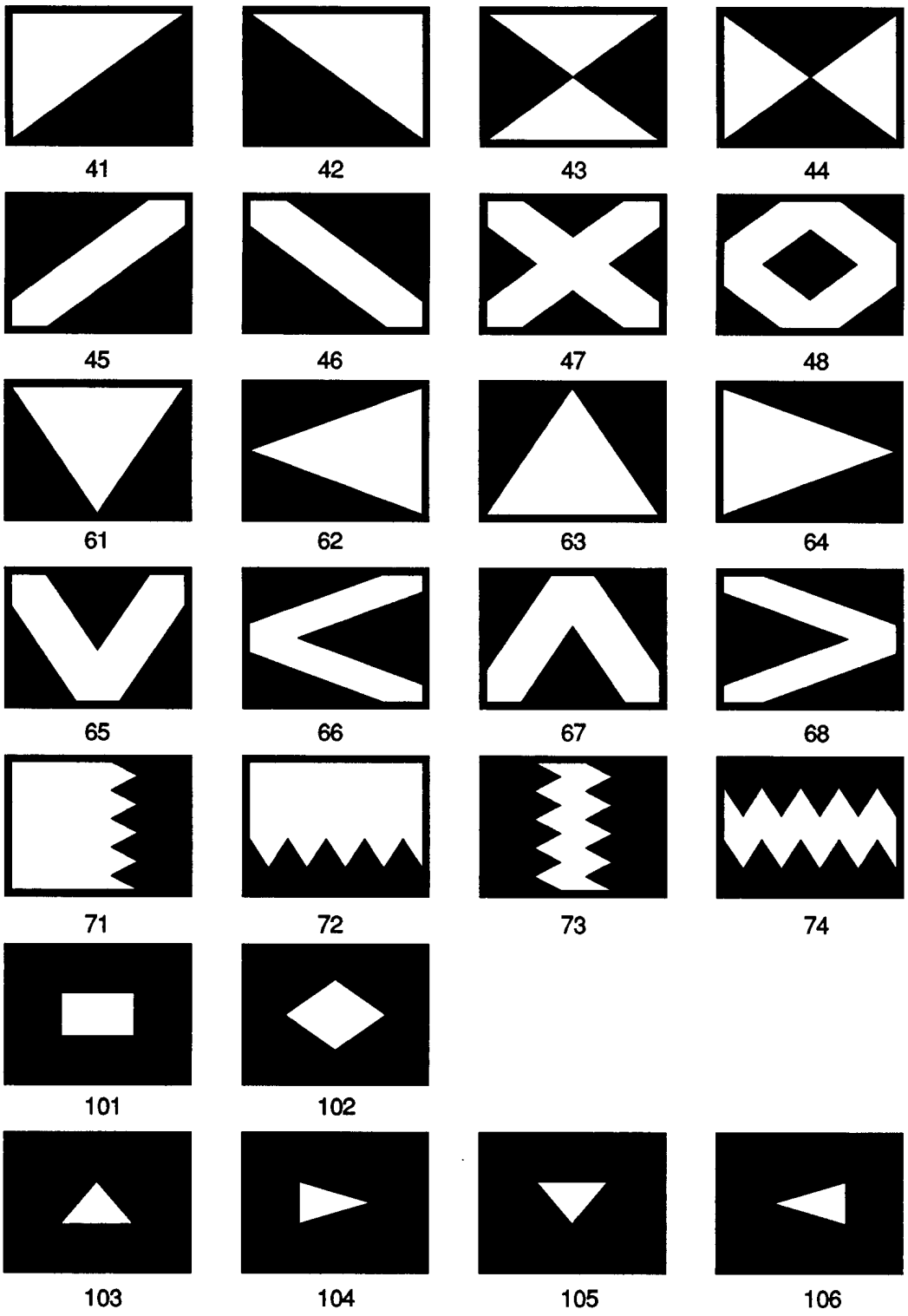
7.6.3.6 Quad split modifiers

D – Diagonal quad split. Default is rectangular;
S – Soft edge;
B – Bordered edge.



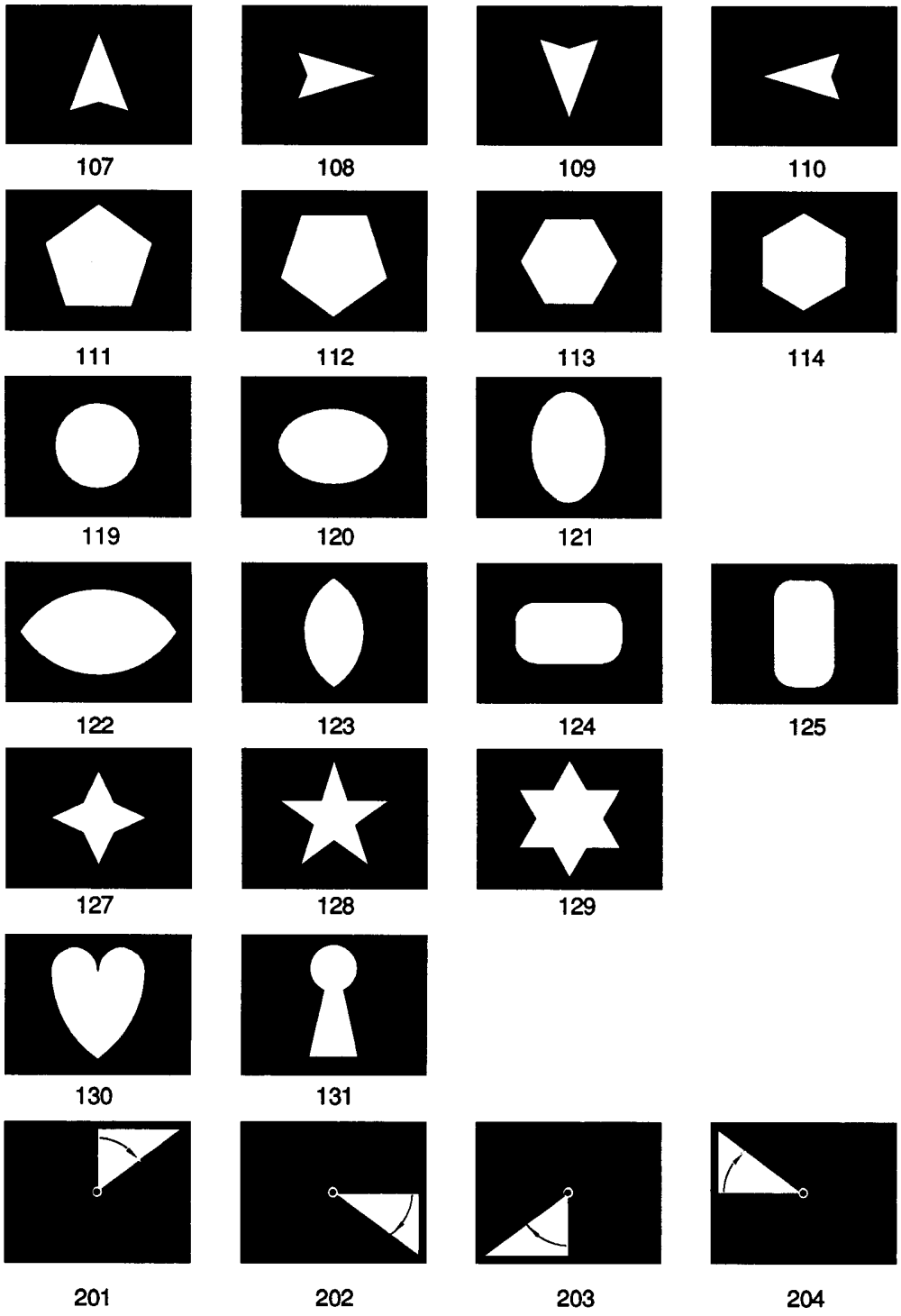
NOTE – Sketch shows effect. Normal direction of change is toward increasing white area.

Figure 1 – Wipe patterns



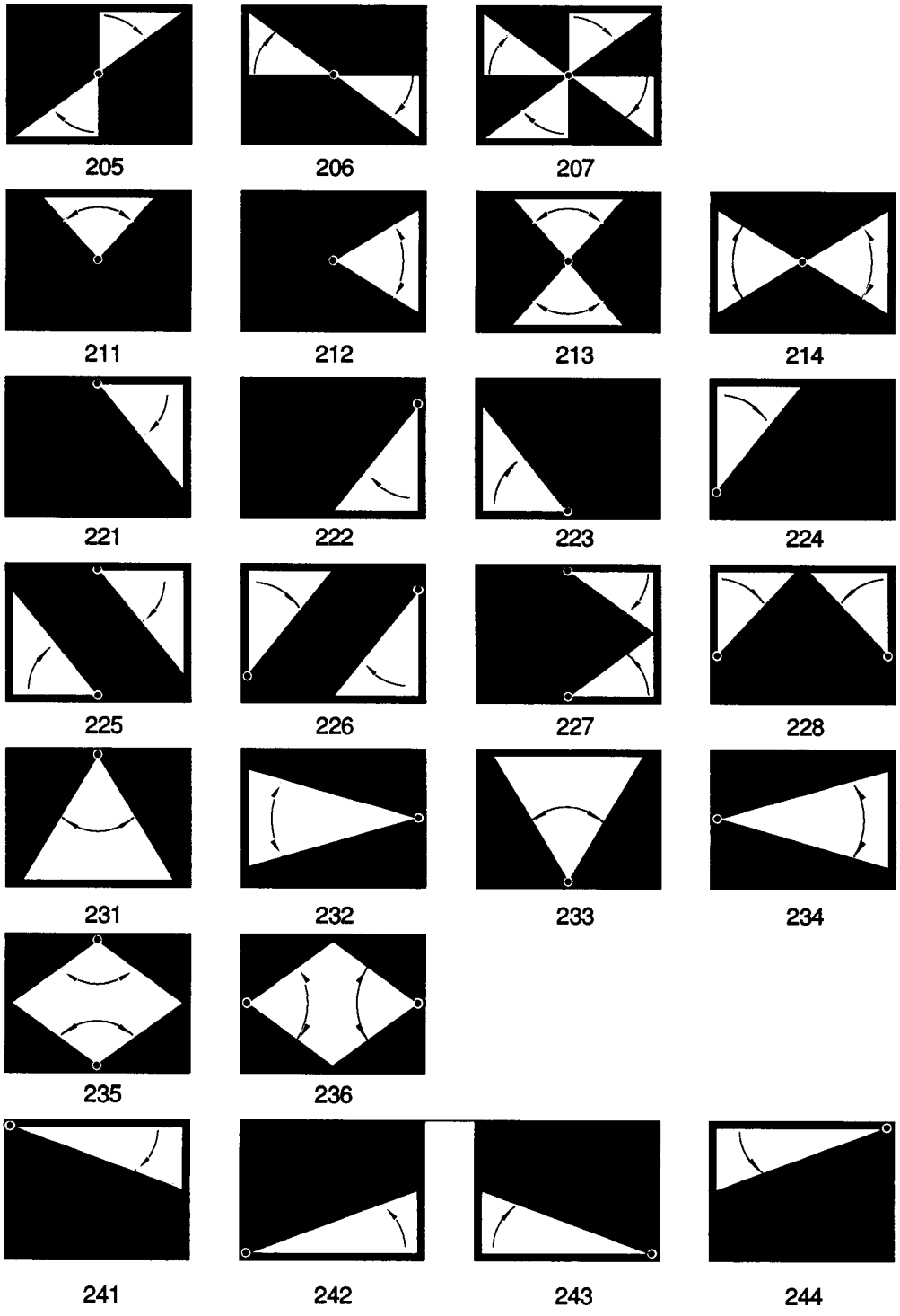
NOTE – Sketch shows effect. Normal direction of change is toward increasing white area.

Figure 1 (continued)



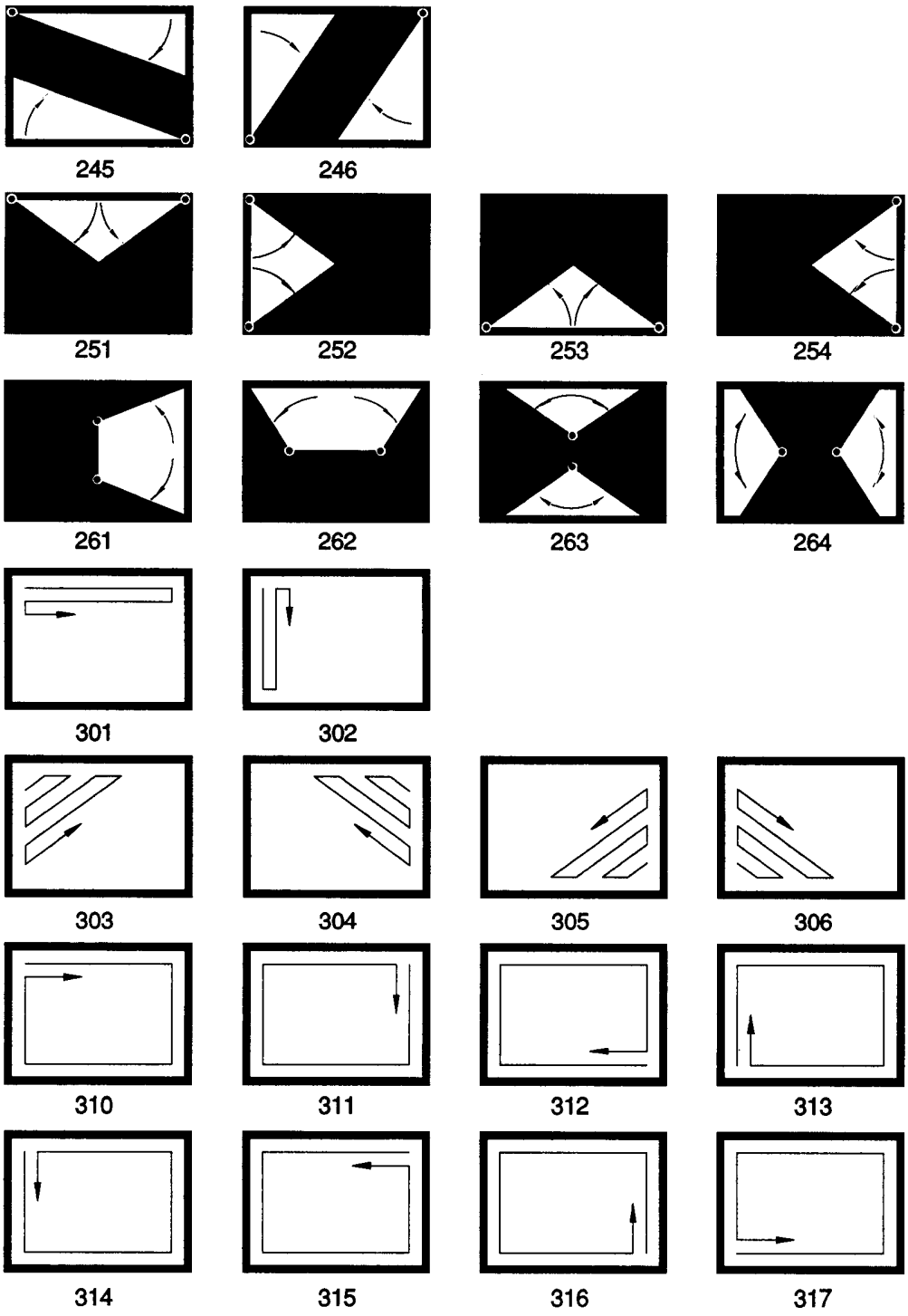
NOTES
1 Sketch shows effect. Normal direction of change is toward increasing white area.
2 Center of rotation is shown by ⊙ .

Figure 1 (continued)



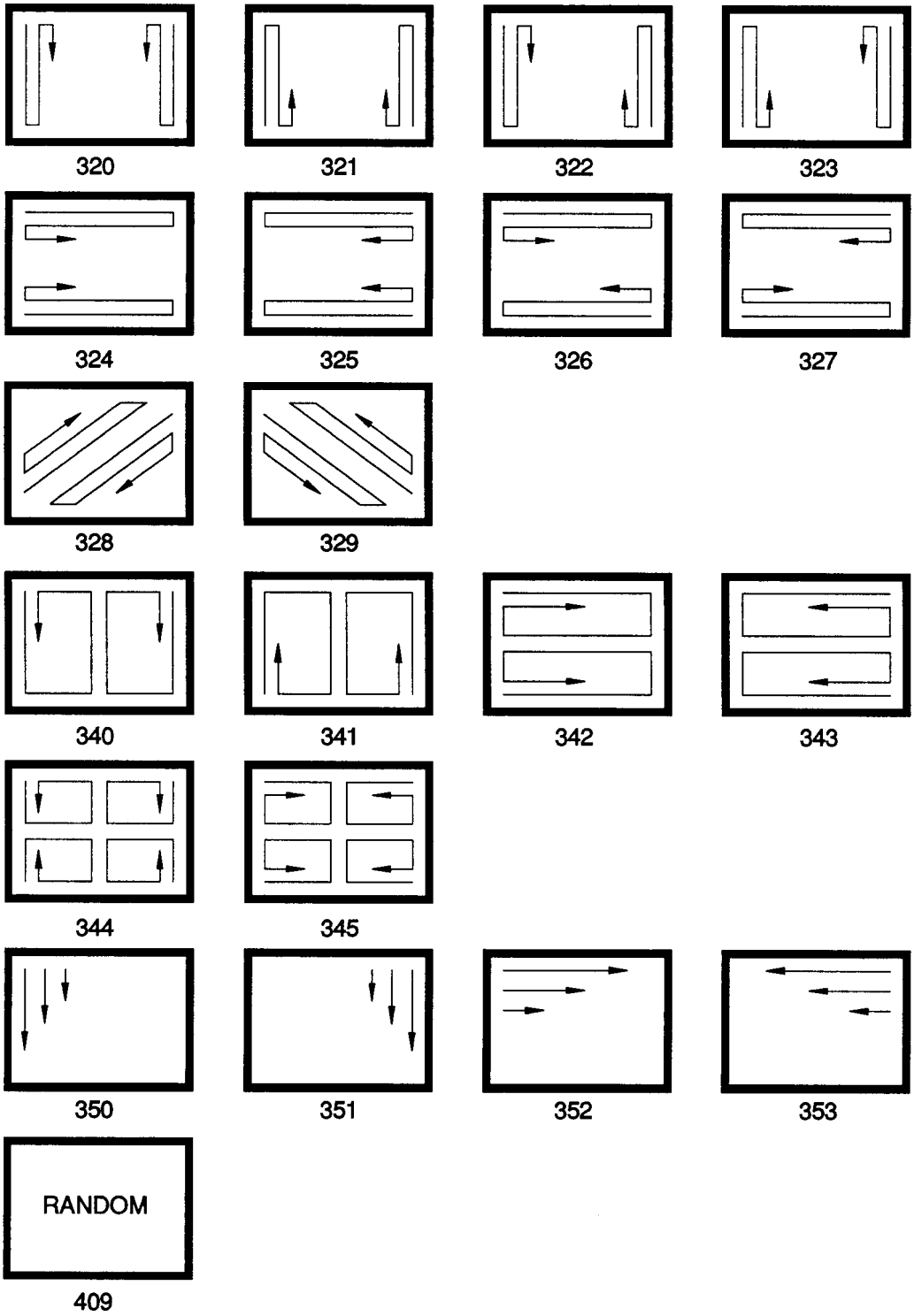
NOTES
1 Sketch shows effect. Normal direction of change is toward increasing white area.
2 Center of rotation is shown by ⊙ .

Figure 1 (continued)



NOTES
1 Sketch shows effect. Normal direction of change is toward increasing white area.
2 Center of rotation is shown by ⊙ .

Figure 1 (continued)



NOTE – Sketch shows effect. Normal direction of change is toward increasing white area.

Figure 1 (concluded)

7.6.3.7 Quad specifier

The QUAD_SPECIFIER is a single digit, from 1 to 4 inclusively, which specifies the quadrant of the quad split to which this SOURCE_ELEMENT refers. The quadrants are laid out as shown in figure 2.

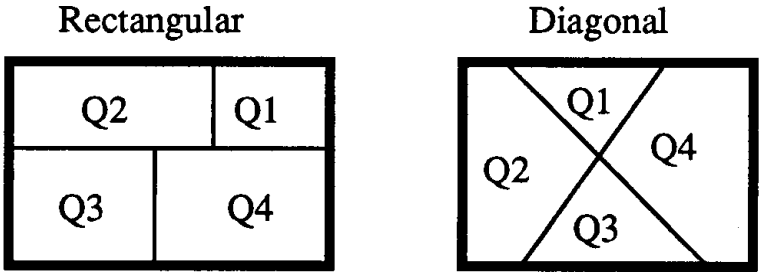


Figure 2 – Quad-split quadrants

7.6.4 Effect parameter field

Some effects require an EFFECT_PARAMETER_FIELD following the EFFECT_FIELD and separated from it by a field separator. These effects are dissolve, wipe, key, matte, and nonadditive mix. The first parameter is the effect duration which consists of a REAL_NUMBER indicating the duration of the effect in frames. Following the duration there is an optional EFFECT_LIMIT_SUBFIELD.

7.6.4.1 Effect limit subfield

The EFFECT_LIMIT_SUBFIELD is optional. When it is missing, the effect transitions from its minimum level to its maximum level. When present, the EFFECT_LIMIT_SUBFIELD consists of a left parenthesis, an effect limit specifying the start level, a comma, and an effect limit specifying the end level followed by a right parenthesis. The effect limits consist of REAL_NUMBERs, where 100.0 indicates the maximum effect level and 0 indicates the minimum effect level. The effect starts at the level indicated by the first limit, and ends at the level indicated by the second limit. When an EFFECT_LIMIT_SUBFIELD is used, the effect duration represents the time in frames required for the effect to transition from the start level to the end level.

7.6.5 Source entry field

The SOURCE_ENTRY_FIELD consists of the source entry TIME_CODE followed by an optional PLAYBACK_SUBFIELD. The source entry time code designates the first field taken from the indicated source, field 1 or 2 being differentiated by the seconds-frames separator, as described in clause 8.

7.6.5.1 Playback subfield

The PLAYBACK_SUBFIELD, when present, directly follows the source entry time and consists of a left parenthesis, followed by a SOURCE_SPEED_SPECIFIER or a VIRTUAL_SPEED_SPECIFIER, and ending with a right parenthesis.

7.6.5.2 Source speed specifier

The SOURCE_SPEED_SPECIFIER consists of a REAL_NUMBER followed by an optional VIDEO_FIELD_SEQUENCE_SPECIFIER. The REAL_NUMBER represents source speed as a percentage of normal play speed. Thus, the value 100 indicates full play speed and 0 indicates still frame. Negative values indicate reverse motion.

7.6.5.3 Video field sequence specifier

The VIDEO_FIELD_SEQUENCE_SPECIFIER allows specification of frame versus field mode, and field sequencing. This specifier consists of one of the following options:

F1 – Field 1, interlaced with field 2;
F2 – Field 2, interlaced with the following frame's field 1;
V – Vertical mode. Each field is presented as it is encountered during playback of the source material.

7.6.5.4 Virtual speed specifier

The VIRTUAL_SPEED_SPECIFIER is an EDIT_NUMBER_SUBFIELD followed by the character '<.' The VIRTUAL_SPEED_SPECIFIER references a PLAYBACK_SEQUENCE following a strategy similar to that of virtual edits. The PLAYBACK_SEQUENCE consists of a series of PLAYBACK_ELEMENTs. In a PLAYBACK_ELEMENT, a position and speed relationship between source and audio/video product is specified consisting of a SOURCE_OFFSET, SOURCE_SPEED, SYNC_OFFSET and optional MOTION_TYPE. By chaining several PLAYBACK_ELEMENTs, any motion sequence may be fully specified and reproduced exactly as it originally happened. Transition from one source speed to the next will be accomplished according to the MOTION_TYPE, if any; then according to the preceding MOTION_DIRECTIVE, if any; then according to the rules described in 7.7.

7.6.6 Source exit field

The SOURCE_EXIT_FIELD consists of a TIME_CODE that will designate the first field of source material not included in the audio/video product via the effect specified by the current SOURCE_ELEMENT. Note that the source exit and sync exit time codes do not include involvement of the current source in any effect specified in the following element(s) of the current edit.

7.6.7 Sync entry field

The SYNC_ENTRY_FIELD consists of a TIME_CODE that designates, for each SOURCE_ELEMENT, the video field in the audio/video product that is synchronous with the video source field specified by the SOURCE_ENTRY_FIELD. Initiation of any indicated effect will also be synchronous with the sync entry time code. Every SOURCE_ELEMENT within an edit must have a sync entry time code greater than or equal to the sync entry time code of the first SOURCE_ELEMENT in the edit.

7.6.8 Sync exit field

The SYNC_EXIT_FIELD consists of a TIME_CODE that designates the first field of the audio/video product not specified by the current SOURCE_ELEMENT. The SYNC_EXIT_FIELD and SOURCE_EXIT_FIELD also designate a synchronous relationship between the video source and the audio/video product.

7.7 Playback element

The PLAYBACK_ELEMENT is used to specify dynamic tape motion in a manner which allows speed variation to be precisely duplicated. A PLAYBACK_SEQUENCE consists of an unlimited number of PLAYBACK_ELEMENTs. Each PLAYBACK_ELEMENT consists of the following fields:

– EDIT_NUMBER;
– SOURCE_OFFSET;
– SOURCE_SPEED;
– SYNC_OFFSET;
– MOTION_TYPE.

The EDIT_NUMBER field of the PLAYBACK_ELEMENT (with a PLAYBACK_ELEMENT character), is followed by three values which define a position and speed relationship between source and audio/video product. These three values specify all information available on the condition of the source transport at any point in time. Very rough motion sequences may be defined by providing only a few PLAYBACK_ELEMENTs and more precisely repeatable motions may be generated by providing more PLAYBACK_ELEMENTs for any given period of time. The optional MOTION_TYPE field gives further information on the type of interpolation desired between PLAYBACK_ELEMENTs.

In the event that precise duplication of the original PLAYBACK_SEQUENCE cannot be obtained because insufficient PLAYBACK_ELEMENTs have been defined, or because the motion ballistics of the EDL-originating source transport are unreproducible by the EDL-reproducing source transport, and in the absence of MOTION_TYPE specifications or prior MOTION_DIRECTIVEs, interpretation of each PLAYBACK_ELEMENT's values shall be resolved by assuming constant acceleration between the speed values of the specified PLAYBACK_ELEMENTs. Any residual value-interpretation errors shall then be resolved in favor of the accuracy of the positional relationships between the source and the audio/video product, rather than in the accuracy of speed interpretation. Such interpretation of the PLAYBACK_ELEMENTs' values shall be deemed compliant with this specification.

7.7.1 Source offset

The SOURCE_OFFSET field specifies an offset in frames through which the source transport moves during the time that the PLAYBACK_ELEMENT is active. This offset is relative to the SOURCE_ENTRY of the SOURCE_ELEMENT which referenced the PLAYBACK_SEQUENCE of which the PLAYBACK_ELEMENT is a part. This offset is a REAL_NUMBER, allowing subframe accuracy, and can be greater than zero, zero, or less than zero.

7.7.2 Source speed

The SOURCE_SPEED field consists of a left parenthesis, followed by a SOURCE_SPEED_SPECIFIER (including an optional VIDEO_FIELD_SEQUENCE_SPECIFIER), followed by a right parenthesis. The SOURCE_SPEED_SPECIFIER specifies the speed of the source material to be attained by the end of the PLAYBACK_ELEMENT and the VIDEO_FIELD_SEQUENCE_SPECIFIER specifies the sequencing of the source fields. SOURCE_SPEED_SPECIFIER and VIDEO_ FIELD_ SEQUENCE_SPECIFIER are defined above in 7.6.5.2 and 7.6.5.3.

7.7.3 Sync offset

The SYNC_OFFSET field specifies the offset in frames through which the audio/video product moves during the time that the PLAYBACK_ELEMENT is active. This offset is relative to the SYNC_ENTRY of the SOURCE_ELEMENT which referenced the PLAYBACK_SEQUENCE. This offset is a REAL_NUMBER, allowing subframe accuracy, but must be greater than zero.

7.7.4 Motion type

The optional MOTION_TYPE field specifies the type of interpolation desired from the previous PLAYBACK_ELEMENT to the current PLAYBACK_ELEMENT. It may be BEST, indicating that the editing equipment and EDL-producing source are free to choose the best method; or SPLINE, indicating that cubic B-spline interpolation is to be used; or a single digit indicating the maximum order of polynomial curve to be used. If SPLINE is specified, all subsequent MOTION_TYPE fields in the current PLAYBACK_SEQUENCE will be ignored, and a cubic B-spline will be used for all interpolation from the first use of SPLINE to the end of the sequence.

If no MOTION_TYPE is specified on a PLAYBACK_ELEMENT, interpolation will follow the type specified for the previous PLAYBACK_ELEMENT in the same PLAYBACK_SEQUENCE. If no such MOTION_TYPE is specified, interpolation will follow the type specified by a preceding MOTION_DIRECTIVE. If no MOTION_DIRECTIVE is given, interpolation will follow the rules given in 7.7.

8 Time code

Time code is expressed in terms of four two-digit groups representing hours (HH), minutes (MM), seconds (SS), and frames (FF), separated by punctuation marks (p), in the form HHpMMpSSpFF, followed by an optional SUB_FRAME_SPECIFIER. The separator between the seconds and frames groups is significant in that it is used to denote drop-frame/nondrop-frame time code and to indicate field 1 or field 2, as follows:

. (period) Nondrop-frame code, field 1;
: (colon) Nondrop-frame code, field 2;
, (comma) Drop-frame code, field 1;
; (semicolon) Drop-frame code, field 2.

For video standards which do not support drop-frame time code, the nondrop-frame indicators shall be used. Allowance is made for the expression of time code in compact form by the suppression of leading zeroes and associated separators, up to but not including the seconds-frames separator.

8.1 Sub frame specifier

The SUB_FRAME_SPECIFIER is an option which can be appended to any time code to specify subframe accuracy. The SUB_FRAME_SPECIFIER consists of a percent sign (%), followed by a REAL_NUMBER specifying a percent of a frame. This percent of a frame is relative to the start of the field specified by the time code. This percent must be greater than zero and less than one hundred.

Annex A (normative)
ASCII character values

ASCII character set
(American National Standard Code for Information Interchange)

Character values in octal

000 NUL	020 DLE	040 SP	060 0	100 @	120 P	140 ‘	160 p
001 SOH	021 DC1	041 !	061 1	101 A	121 Q	141 a	161 q
002 STX	022 DC2	042 “	062 2	102 B	122 R	142 b	162 r
003 ETX	023 DC3	043 #	063 3	103 C	123 S	143 c	163 s
004 EOT	024 DC4	044 $	064 4	104 D	124 T	144 d	164 t
005 ENQ	025 NAK	045 %	065 5	105 E	125 U	145 e	165 u
006 ACK	026 SYN	046 &	066 6	106 F	126 V	146 f	166 v
007 BEL	027 ETB	047 ‘	067 7	107 G	127 W	147 g	167 w
010 BS	030 CAN	050 (	070 8	110 H	130 X	150 h	170 x
011 HT	031 EM	051)	071 9	111 I	131 Y	151 i	171 y
012 LF	032 SUB	052 *	072 :	112 J	132 Z	152 j	172 z
013 VT	033 ESC	053 +	073 ;	113 K	133 [	153 k	173 {
014 FF	034 FS	054 ,	074 <	114 L	134 \	154 l	174 |
015 CR	035 GS	055 -	075 =	115 M	135]	155 m	175 }
016 S0	036 RS	056 .	076 >	116 N	136 ^	156 n	176 ~
017 SI	037 US	057 /	077 ?	117 O	137 _	157 o	177 DEL

Character values in hexidecimal

00 NUL	10 DLE	20 SP	30 0	40 @	50 P	60 ‘	70 p
01 SOH	11 DC1	21 !	31 1	41 A	51 Q	61 a	71 q
02 STX	12 DC2	22 “	32 2	42 B	52 R	62 b	72 r
03 ETX	13 DC3	23 #	33 3	43 C	53 S	63 c	73 s
04 EOT	14 DC4	24 $	34 4	44 D	54 T	64 d	74 t
05 ENQ	15 NAK	25 %	35 5	45 E	55 U	65 e	75 u
06 ACK	16 SYN	26 &	36 6	46 F	56 V	66 f	76 v
07 BEL	17 ETB	27 ‘	37 7	47 G	57 W	67 g	77 w
08 BS	18 CAN	28 (	38 8	48 H	58 X	68 h	78 x
09 HT	19 EM	29)	39 9	49 I	59 Y	69 i	79 y
0A LF	1A SUB	2A *	3A :	4A J	5A Z	6A j	7A z
0B VT	1B ESC	2B +	3B ;	4B K	5B [	6B k	7B {
0C FF	1C FS	2C ,	3C <	4C L	5C \	6C l	7C |
0D CR	1D GS	2D -	3D =	4D M	5D]	6D m	7D }
0E S0	1E RS	2E .	3E >	4E N	5E ^	6E n	7E ~
0F SI	1F US	2F /	3F ?	4F O	5F _	6F o	7F DEL

Annex B (normative)
Edit decision list syntax

The following is a formal definition of the syntax of the edit decision list:

```
EDL:
    INITIAL_DIRECTIVES EDIT_LIST 'SUB'
|   INITIAL_DIRECTIVES EDIT_LIST
;

INITIAL_DIRECTIVES:
    SMPTE_DIRECTIVE FRAME_RATE_DIRECTIVE
|   SMPTE_DIRECTIVE INITIAL_DIRECTIVES
;

SMPTE_DIRECTIVE:
    "SMPTE" SEPARATOR "RP" NUMBER
|   "SMPTE" SEPARATOR "RP" NUMBER ['-' NUMBER]
|   "SMPTE" SEPARATOR "EG" NUMBER
|   "SMPTE" SEPARATOR "EG" NUMBER ['-' NUMBER]
|   "SMPTE" SEPARATOR "ST"' NUMBER "M"
|   "SMPTE" SEPARATOR "STD" NUMBER "M" ['-' NUMBER]
;

FRAME_RATE_DIRECTIVE:
   "TIME_CODE_MODULUS" SEPARATOR INTEGER
;

EDIT_LIST:
    EDIT_SEQUENCE
|   PLAYBACK_SEQUENCE
|   EDIT_LIST EDIT_SEQUENCE
|   EDIT_LIST PLAYBACK_SEQUENCE
;

EDIT_SEQUENCE:
    [ SEPARATOR ] TERMINATOR
|   EDIT_ELEMENT [ SEPARATOR ] TERMINATOR
|   EDIT_SEQUENCE EDIT_ELEMENT [ SEPARATOR ] TERMINATOR
;

PLAYBACK_SEQUENCE:
    [ SEPARATOR ] TERMINATOR
|   PLAYBACK_ELEMENT [ SEPARATOR ] TERMINATOR
|   PLAYBACK_SEQUENCE PLAYBACK_ELEMENT
        [ SEPARATOR ] TERMINATOR
;

ELEMENT:
    EDIT_ELEMENT
|   PLAYBACK_ELEMENT
;

PLAYBACK_ELEMENT:
    EDIT_NUMBER_SUBFIELD '=' SEPARATOR SOURCE_OFFSET
        [ SEPARATOR ] SOURCE_SPEED
        [ SEPARATOR ] SYNC_OFFSET
        [ SEPARATOR MOTION_TYPE ]
;
```

```
SOURCE_OFFSET:
    REAL_NUMBER
;

SOURCE_SPEED:
    '(' SOURCE_SPEED_SPECIFIER ')'
;

SOURCE_SPEED_SPECIFIER:
    REAL_NUMBER [ VIDEO_FIELD_SEQUENCE_SPECIFIER ]
;

VIDEO_FIELD_SEQUENCE_SPECIFIER:
    "V"         /* Vertical mode                    */
|   "F1"        /* Frame mode, field 1 dominant */
|   "F2"        /* Frame mode, field 2 dominant  */
|               /* Empty field                       */
;

SYNC_OFFSET:
    REAL_NUMBER
;

EDIT_ELEMENT:
    SYSTEM_DIRECTIVE
|   COMMENT_ELEMENT
|   USER_ELEMENT
|   TRIGGER_ELEMENT
|   DEVICE_CONTROL_ELEMENT
|   SOURCE_ELEMENT
;

SYSTEM_DIRECTIVE:
    INFORMATIONAL_DIRECTIVE
|   AUTO_ASSEMBLE_DIRECTIVE
|   CONTROL_DIRECTIVE
|   MISCELLANEOUS_DIRECTIVE
|   MOTION_DIRECTIVE
;

INFORMATIONAL_DIRECTIVE:
    '*' PRINTABLE_STRING
|   "TITLE" SEPARATOR PRINTABLE_STRING
;

AUTO_ASSEMBLE_DIRECTIVE:
    "WAIT"
|   "SKIP"
|   "BELL"
;

CONTROL_DIRECTIVE:
    "RECORD" SEPARATOR MODE_FIELD SEPARATOR
        MEDIUM_IDENTIFIER '=' TIME_CODE
|   "NORECORD" SEPARATOR MODE_FIELD SEPARATOR
        MEDIUM_IDENTIFIER
|   "SYNC" SEPARATOR { SOURCE_IDENTIFIER '=' TIME_CODE }
|   "NOSYNC" { SEPARATOR SOURCE_IDENTIFIER }
|   "SLAVE" SEPARATOR MASTER_SRC_ID '=' TIME_CODE
        { SEPARATOR SLAVE_SRC_ID '=' TIME_CODE }
|   "NOSLAVE" SEPARATOR MASTER_SRC_ID
        { SEPARATOR SLAVE_SRC_ID }
;
```

```
MISCELLANEOUS_DIRECTIVE:
    “AUDIO” SEPARATOR MODE_AUDIO_SUBFIELD
|   “INCLUDE” SEPARATOR FILENAME ’=’ TIME_CODE
|   “MEDIUM” SEPARATOR SOURCE_IDENTIFIER SEPARATOR
        MEDIA_TYPE
|   WIPE_DIRECTIVE
;

MOTION_DIRECTIVE:
    “MOTION CURVE” SEPARATOR MOTION_TYPE
;

MEDIA_TYPE:
    “LIVE” [ SEPARATOR PRINTABLE_STRING ]
|   “TAPE” [ SEPARATOR PRINTABLE_STRING ]
|   “OTHER” [ SEPARATOR PRINTABLE_STRING ]
;

MOTION_TYPE:
    “BEST”              /* Best curve for this VTR. */
|   “SPLINE”            /* Cubic B-spline curve.    */
|   NON_ZERO_DIGIT      /* Nth order curve fit.     */
;

WIPE_DIRECTIVE:
    “WIPES” SEPARATOR “SMPTE”
|   “WIPES” SEPARATOR PRINTABLE_STRING
;

MASTER_SRC_ID:
    SOURCE_IDENTIFIER
;

SLAVE_SRC_ID:
    SOURCE_IDENTIFIER
;

SOURCE_IDENTIFICATION_FIELD:
                        /* Note the special significance */
                        /* of “BLACK” and “BARS.” */
SOURCE_IDENTIFIER
|   SOURCE_IDENTIFIER ’.’ SOURCE_AUDIO_SUBFIELD
;

SOURCE_IDENTIFIER:  /* Note 8-character maximum. */
    IDENTIFIER
|   VIRTUAL_SOURCE_IDENTIFIER
                        /* Virtual source reference. */
;

VIRTUAL_SOURCE_IDENTIFIER:
    EDIT_NUMBER_SUBFIELD ’<’
;

MEDIUM_IDENTIFIER:      /* Note 8-character maximum. */
    IDENTIFIER
;
```

```
IDENTIFIER:
    ALPHA
|   DIGIT
|   ‘ _ ’
|   IDENTIFIER ALPHA
|   IDENTIFIER DIGIT
;

FILENAME:
    ’ “ ’ PRINTABLE_STRING ’ “ ’
;

COMMENT_ELEMENT:
    EDIT_NUMBER ’*’ PRINTABLE_STRING
;

USER_ELEMENT:
        /* Format is user defined. */
    EDIT_NUMBER ’?’ PRINTABLE_STRING
;

TRIGGER_ELEMENT:
    EDIT_NUMBER ’#’ SEPARATOR TRIGGER_IDENTIFIER
    [ SEPARATOR TIME_CODE_SRC_IDENTIFIER ]
    [ SEPARATOR TIME_CODE_TYPE ]
    SEPARATOR TIME_CODE
;

TRIGGER_IDENTIFIER:   /* Note 8- character maximum. */
    IDENTIFIER
;

TIME_CODE_SRC_IDENTIFIER:
    SOURCE_IDENTIFIER
;

TIME_CODE_TYPE:
    “ENTRY+”
|   “ENTRY-“
|   “EXIT+”
|   “EXIT-“
|   “DUR”
;

DEVICE_CONTROL_ELEMENT:
    EDIT_NUMBER ’%’
        SEPARATOR DEVICE_IDENTIFIER_FIELD
        [ SEPARATOR TIME_CODE_SRC_IDENTIFIER ]
        [ SEPARATOR TIME_CODE_TYPE ]
        SEPARATOR MESSAGE_TIME_FIELD
        SEPARATOR MESSAGE_COUNT_FIELD
        SEPARATOR MESSAGE_FIELD
;

DEVICE_IDENTIFIER_FIELD:      /* Note 8-character maximum. */
    IDENTIFIER
;

MESSAGE_TIME_FIELD:
    “SETUP”
|   TIME_CODE
;
```

```
MESSAGE_COUNT_FIELD:
    HEXADECIMAL_NUMBER
;

MESSAGE_FIELD:
    MESSAGE_BYTE
|   MESSAGE_BYTE MESSAGE_FIELD
|   MESSAGE_BYTE SEPARATOR MESSAGE_FIELD
;

MESSAGE_BYTE:
    HEXADECIMAL_DIGIT HEXADECIMAL_DIGIT
;

SOURCE_ELEMENT:
    EDIT_NUMBER
        SEPARATOR SOURCE_IDENTIFICATION_FIELD
        SEPARATOR MODE_FIELD
        SEPARATOR EFFECT_FIELD
        SEPARATOR SOURCE_ENTRY_FIELD
        SEPARATOR SOURCE_EXIT_FIELD
        SEPARATOR SYNC_ENTRY_FIELD
        SEPARATOR SYNC_EXIT_FIELD
;

SOURCE_ENTRY_FIELD:
    TIME_CODE
|   TIME_CODE PLAYBACK_SUBFIELD
;

SOURCE_EXIT_FIELD:
    TIME_CODE
;

SYNC_ENTRY_FIELD:
    TIME_CODE
;

SYNC_EXIT_FIELD:
    TIME_CODE
;

PLAYBACK_SUBFIELD:
    '(' SOURCE_SPEED_SPECIFIER ')'
|   '(' VIRTUAL_SPEED_SPECIFIER ')'
;

VIRTUAL_SPEED_SPECIFIER:
    EDIT_NUMBER_SUBFIELD '<'
;

SOURCE_AUDIO_SUBFIELD:
    AUDIO_SOURCE
|   AUDIO_RANGE
|   SOURCE_AUDIO_SUBFIELD ',' AUDIO_SOURCE
|   SOURCE_AUDIO_SUBFIELD ',' AUDIO_RANGE
;
```

```
AUDIO_SOURCE:
    AUDIO_TRACK_NUMBER AUDIO_MIX_LEVEL
|   AUDIO_MIX_RANGE
|   AUDIO_SOURCE '+' AUDIO_TRACK_NUMBER
        AUDIO_MIX_LEVEL
|   AUDIO_SOURCE '+' AUDIO_MIX_RANGE |
;

AUDIO_MIX_RANGE:                /* Audio mix, successive channels      */
                                /* The second track number must        */
                                / * be greater than the first number.   */
    AUDIO_TRACK_NUMBER '+' '+' AUDIO_TRACK_NUMBER
;

AUDIO_RANGE:                    /* The second track number must        */
                                /* be greater than the first           */
                                /* track number.                       */
    AUDIO_TRCK_NUMBER '-' AUDIO_TRACK_NUMBER
;

AUDIO_TRACK_NUMBER:
    DIGIT
|   DIGIT DIGIT
;

AUDIO_MIX_LEVEL:                /* Audio mix level (dB)                */
    '(' REAL_NUMBER ')'
|   /*  Empty field  */
;

MODE_FIELD:
    'V'                         /* Video only                          */
|   'A'                         /* All audio trcks                     */
|   'B'                         /* Video and all audio tracks          */
|   'A' MODE_AUDIO_SUBFIELD     /* Only audio tracks                   */
|   'B' MODE_AUDIO_SUBFIELD     /* Video and audio tracks              */
;

MODE_AUDIO_SUBFIELD:
    AUDIO_TRACK_NUMBER
|   AUDIO_RANGE
|   MODE_AUDIO_SUBFIELD ',' AUDIO_TRACK_NUMBER
|   MODE_AUDIO_SUBFIELD ',' AUDIO_RANGE
;

EFFECT_FIELD:
    SINGLE_SOURCE_EFFECT
|   MULTIPLE_SOURCE_EFFECT SEPARATOR
        EFFECT_PARAMETER_FIELD
;

SINGLE_SOURCE_EFFECT:
    'C'                         /* Cut                                 */
|   'R'                         /* Sync roll                           */
;

MULTIPLE_SOURCE_EFFECT:
    'D'                         /* Dissolve                            */
|   'W' WIPE_SPEC               /* Wipe                                */
|   KEY_TYPE                    /* Key                                 */
|   KEY_TYPE KEY_SPEC           /* Key                                 */
|   KEY_TYPE WIPE_SPEC          /* Wipe key                            */
```

```
|    KEY_TYPE KEY_SPEC WIPE_SPEC     /* Wipe key                        */
|    'Q' QUAD_MODIFIER DIGIT         /* Quad split                      */
|    'F'                             /* Foreground filler               */
|    'N'                             /* Nonadditive mix                 */
|    'X'                             /* Audio mix                       */
|    'X' '(' REAL_NUMBER ')'         /* Audio mix (w DB level)          */
;

WIPE_SPEC:
     NUMBER
|    WIPE_SPEC WIPE_MODIFIER
;

WIPE_MODIFIER:
     'R'                           /* Reverse                          */
|    'S'                           /* Soft                             */
|    'B'                           /* Border                           */
|    'P'                           /* Position                         */
|    'M'                           /* Pattern modulator                */
|    'A'                           /* Shadow                           */
|    'T'                           /* Tumble (revolve) pattern         */
|    'G'                           /* Spot light                       */
|    'H' INTEGER                   /* Horizontal replication           */
|    'V' INTEGER                   /* Vertical replication             */
|    'C'                           /* CKBD (matrix wipes only)         */
|    'X'                           /* eXternal pattern number          */
;

QUAD_MODIFIER:
     QUAD_MODIFIER QUAD_DESCRIPTOR
|    /*  Empty field  */
;

QUAD_ DESCRIPTOR:
     'D'                           /* Diagonal quad split              */
|    'S'                           /* Soft                             */
|    'B'                           /* Border                           */
;

KEY_TYPE:
     'K'                           /* Key                              */
|    'M'                           /* Matte                            */
;

KEY_SPEC:
     KEY_MODIFIER
|    KEY_SPEC KEY_MODIFIER
;

KEY_MODIFIER:
     'H'                           /* Shadow                           */
|   'O'                           /* Outline                          */
|    'E'                           /* Edge                             */
|   'I'                           /* Invert foreground background     */
|    'T'                           /* Soft edge                        */
|    'C'                           /* Chroma key                       */
;

EFFECT_PARAMETER_FIELD:
     REAL_NUMBER                   /* Effect parameter field           */
|    REAL_NUMBER EFFECT_LIMIT_SUBFIELD
;
```

```
EFFECT_LIMIT_SUBFIELD:
      '(' REAL_NUMBER ',' REAL_NUMBER ')' /* (Start, end) */
;

TIME_CODE:
     TIME_CODE_FIELD_SPECIFIER
|    TIME_CODE_FIELD_SPECIFIER SUB_FRAME_SPECIFIER
;

TIME_CODE_FIELD_SPECIFIER:
     LEADING_RADIX_NUMBER
          TIME_CODE_SEPARATOR RADIX_NUMBER
          TIME_CODE_SEPARATOR RADIX_NUMBER
          TIME_CODE_SEPARATOR RADIX_NUMBER
|    LEADING_RADIX_NUMBER
          TIME_CODE_SEPARATOR RADIX_NUMBER
          TIME_CODE_SEPARATOR RADIX_NUMBER
|    LEADING_RADIX_NUMBER
          TIME_CODE_SEPARATOR RADIX_NUMBER
|    TIME_CODE_SEPARATOR RADIX_NUMBER
;

SUB_FRAME_SPECIFIER:
     '%' REAL_NUMBER
               /* Specifies a percentage of a frame to be added to the start of the field specified by the time code. */
;

LEADING_RADIX_NUMBER:
     DIGIT
|    DIGIT DIGIT
;

RADIX_NUMBER:
     DIGIT DIGIT
;

PRINTABLE_STRING:
     PRINTABLE_CHARACTER
|    PRINTABLE_STRING PRINTABLE_CHARACTER
;

PRINTABLE_CHARACTER:
     '!'  |  ' " '  |  '#'  |  '$'  |  '%'  |  '&'  | '''
|    '('  |  ')'  |  '*'  |  '+'  |  '-'  | '/'
|    '<'  |  '='  |  '>'  |  '?'
|    '@'
|    '['  |  '\'  ']'  |  '^ '  |  '_'
|    '"'
|    '{'  |  '|'  |  '}'  |  '~'
|    ALPHA
|    DIGIT
|    TIME_CODE_SEPARTOR
|    SEPARATOR
;

TIME_CODE_SEPARATOR:
     ':'
|    '.'
|    ','
|    ' '
|    '-'
;
```

```
EDIT_NUMBER:
    EDIT_NUMBER_SUBFIELD
        VIRTUAL_EDIT_INDICATOR_SUBFIELD
        RECORDED_INDICATOR_SUBFIELD
;


EDIT_NUMBER_SUBFIELD:
    DIGIT                                       /* Note the limit to 6 digits. */
| DIGIT ALPHA
;

VIRTUAL_EDIT_INDICATOR_SUBFIELD:
    '>'
|   /*  Empty field  */
;

RECORDED_INDICATOR_SUBFIELD:
    '!'
|   /*  Empty field  */
;

ALPHA:
    UPPER_ALPHA
|   LOWER_ALPHA
;

UPPER_ALPHA:
    'A'  |  'B'  |  'C'  |  'D'  |  'E'  |  'F'  |  'G'  |  'H'  |  'I'  |  'J'
|   'K'  |  'L'  |  'M   |  'N'  |  'O'  |  'P'  |  'Q'  |  'R'  |  'S'  |  'T'
|   'U'  |  'V'  |  'W'  |  'X'  |  'Y'  |  'Z'
;

LOWER_ALPHA:
    'a'  |  'b'  |  'c'  |  'd'  |  'e'  |  'f'  |  'g'  |  'h'  |  'i'  |  'j'
|   'k'  |   'l'  |  'm'  |  'n'  |  'o'  |  'p'  |  'q'  |   'r'  |   's' | 't'
|   'u'  |   'v' |  'w'  |  'x'  |   'y'  |  'z'
;

REAL_NUMBER:
    INTEGER
|   INTEGER ',' NUMBER
;

INTEGER:
    NUMBER
|   '+' NUMBER
|   '-' NUMBER
;

NUMBER:
    DIGIT
|   NUMBER DIGIT
;


HEXADECIMAL_NUMBER:
    HEXADECIMAL_DIGIT
|   HEXADECIMAL_NUMBER HEXADECIMAL_DIGIT
;
```

```
HEXADECIMAL_DIGIT:
    DIGIT
|   'A'  |  'B'  |  'C'  |  'D'  |  'E'  |  'F'
;

DIGIT:
    '0'
|   NON_ZERO_DIGIT
;

NON_ZERO_DIGIT:
    '1'  |  '2'  |  '3'  |  '4'  |  '5'  |  '6'  |  '7'  |  '8'  |  '9'
;

SPACE:
    ' '
;

TERMINATOR:
    'CR'    'LF'
;

OBSOLETE_TERMINATOR:
    'ETB'
;

SEPARATOR:      /* One or more actual separators */
    ACTUAL_SEPARATOR
|   SEPARATOR ACTUAL_SEPARATOR
;

ACTUAL_SEPARATOR:
    SPACE
|   'HT'
|   OBSOLETE_TERMINATOR
|   '\'  'CR'  'LF'
;
```

Annex C (informative)
Examples

Example 1, Virtual source identifiers:

1> REEL1	B	C	00:05:00	00:07:00	01:00:00	01:02:00
1> TITL1	V	K 30(0,1000)	00:00:00	00:02:00	01:00:00	01:02:00
2> 1< B C			01:00:00	01:02:00	02:00:00	02:02:00
2> TITL2	V	K 30(0,1000)	00:00:00	00:02:00	02:00:00	02:02:00
3> REEL2	B	C	02:00:00	02:02:00	03:00:00	03:02:00
3> TITL3	V	K 0(1000,1000)	00:00:00	00:02:00	03:00:00	03:02:00
17A 2< B	C		02:00:00	02:01:00	04:00:00	04:01:00
17A 3< B	D 30		03:00:00	03:02:00	04:01:00	04:03:00

Edit 17A is a delayed dissolve between two virtual source references. The first virtual source reference is to virtual edit 2, which is a key of TITL2 over virtual edit 1. Virtual edit 1 is itself a key of TITL1 over REEL1. Thus, virtual edit 2 consists of TITL2 keyed over the key of TITL1 over REEL1.

The second virtual source reference in edit 17A is to edit 3, a virtual edit, which is a key of TITL3 over REEL2. Note that the source times in edit 17 refer to the sync times in virtual edits 2 and 3.

A good way to visualize the use of a VIRTUAL SOURCE IDENTIFIER is to consider that the referenced edit was recorded on a submaster. Thus, the source times in virtual source references refer to the sync times on the submasters.

Edit 17A can be created by first recording virtual edit 1 on a submaster. That submaster would be used to create virtual edit 2 on a second submaster. Virtual edit 3 would be recorded on a third submaster. The submasters can then be used as sources from which to assemble edit 17A.

Alternatively, edit 17A could be recorded in one pass by making use of a multiple mix effect switcher to create the appropriate keys and dissolve between them.

Example 2, Source audio subfield:

– A.1.2	Source ID is A. The two audio sources are tracks 1 and 2.
– 5A.1-2	Source ID is 5A. The two audio sources are tracks 1 and 2.
– 2.3,4(-3.4)+9(-5.2)	Source ID is 2. The two audio sources are track 3 and a mix of track 4 at -3.4 dB and track 9 at -5.2 dB.
– REEL3.3++8	Source ID is REEL3. The audio source is a mix of tracks 3 through 8.
– STUDIO.4,4++15+22++24	Source ID is STUDIO. The two audio sources are track 4 and a mix of tracks 4 through 15 and tracks 22 through 24.
– 35B<.3,5)	Source ID is edit 35B. The two audio sources are track 3 and track 5. Note that the tracks specified by this SOURCE_AUDIO_SUBFIELD must be a subset of the track specified by the AUDIO TRACK SUBFIELD of edit 35B.
– 15A.1,7-9,3-6,17	Source ID is 15A. The nine audio sources are tracks 1, 7 through 9, 3 through 6, and 17.

Example 3, Audio track subfield:

– V Record video only;
– A Record all available audio tracks. Note that this mode may have limitations placed on it by the“AUDIO” system directive;
– B Record video and all available audio tracks;
– A3 Record audio track number 3 only;
– B5 Record video and audio track number 5 only;
– A1,7 Record audio tracks 1 and 7;
– A2-5 Record audio tracks 2 through 5;
– B1,3 Record video and audio tracks 1 and 3;
– B2-5 Record video and audio tracks 2 through 5;
– B7,3-6 Record video and audio track 7, and audio tracks 3 through 6;
– A1,7-9,3-6,17 Record audio tracks 1, 7 through 9, 3 through 6, and 17.

Example 4, Source elements:

01> BKG1	V	C	01:00:00(50V)	01:02:15	00:00:00	00:05:00
01> BKG2	V	D 60	02:05:00(0F2)	02:05:00	00:05:00	00:10:00
12 01<	V	C	00:02:00	00:10:00	15:00:00	15:08:00
12 MAT1	V	K 0	10:00:00	10:05:00	15:03:00	15:08:00
12 FOG	V	F	05:00:00(20)	05:01:00	15:03:00	15:08:00
12 MUS.5,6	A1,2	C	30:00:00	30:20:00	15:00:00%75	15:20:00

Virtual edit 01 specifies BKG1 dissolving to BKG2 after 5 seconds. BKG1 is at 50% play speed with vertical field sequencing, i.e., fields presented sequentially. BKG2 is in still frame playback with field 2 dominant; i.e., field 2 alternating with the next frame’s field 1. The SOURCE_EXIT is the time reached in 5 seconds at 50% speed.

The last 8 seconds of this edit are used in edit 12, with source MAT1 cutting a key 3 seconds in. Source FOG fills the foreground at 20% play speed.

In the same edit, roughly 20 seconds of audio tracks 5 and 6 of source MUS are recorded on tracks 1 and 2 with an entry subframe offset of 75% of a frame.

Note that EDLs that predominantly specify field 2 time codes can specify subframe accuracy based on field 2 even if the offset results in a point in the following frame’s field 1; i.e., 15:00:00%75 is equivalent to 15.00.01%25.

Example 5, Playback element:

The following is an example of a specified motion curve:

10= 90	(50.0)	105					
10= 120.0	(25.0)	180	3				
10= 129.0	(0F2)	225.0					
11 REEL1	V	C		00:00(10<)	04:10:09	0:00	10:00

Edit 11 is a mid-interval slow-down sequence. The source plays at normal play speed, but must achieve 50% play speed by 105 real time frames into the playback sequence, at which time the source will have traveled 90 frames. The source will then continue at 50%, but must achieve 25% play speed and a 120 frame source offset by 180 frames into the playback sequence. Finally, the source will freeze on field 2 with a 129 frame source offset, 225 real time frames into the playback sequence.

Interpolation from 50% to 25% and from 25% to the freeze will be by third order polynomial curve.

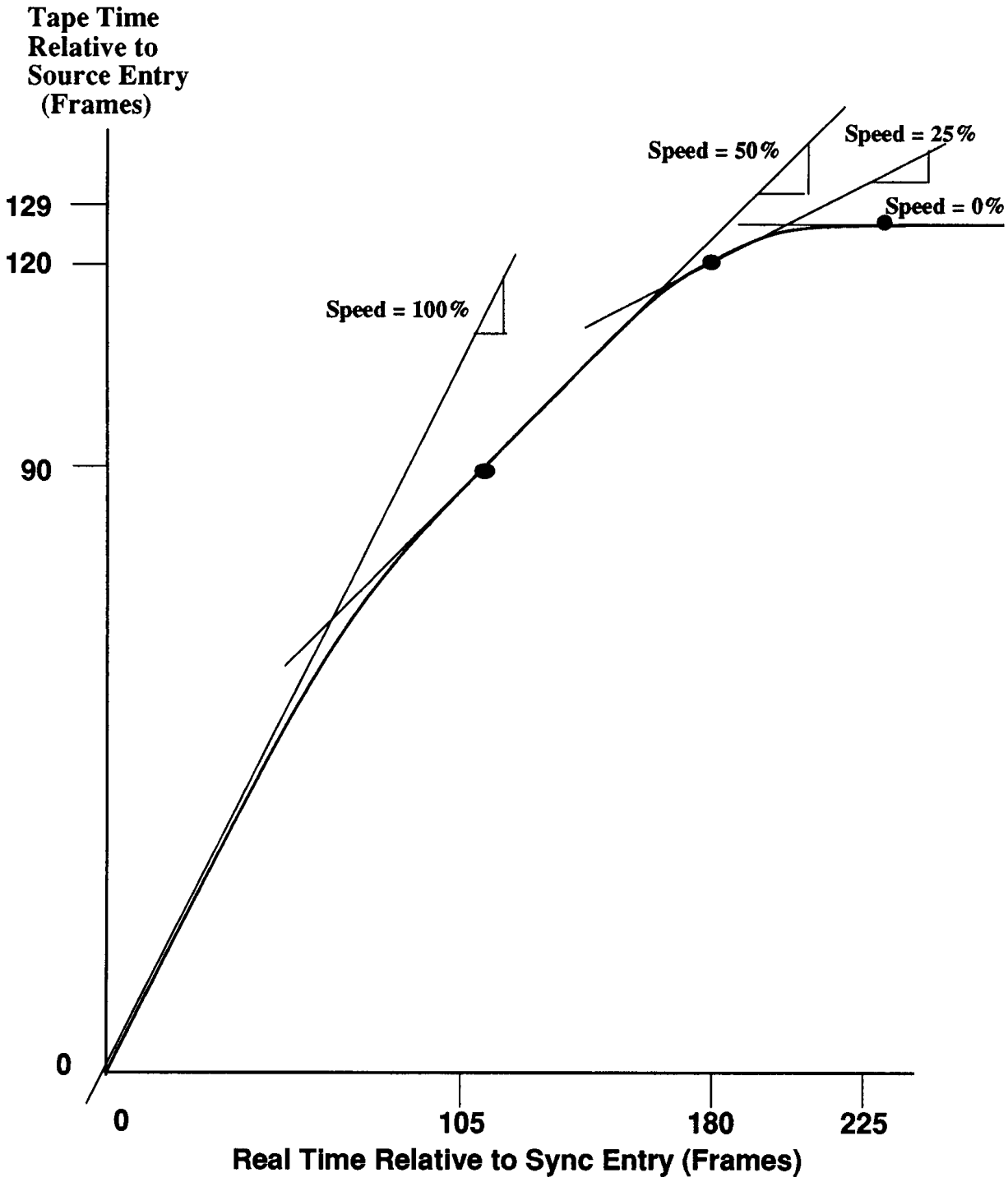


Figure C.1 – Third order polynomial curve