NAME

Cell – Cell compressor/decompressor for compressed image sequences

DESCRIPTION

The Cell image compression technology, which was developed by Sun, has been optimized for rapid decompression and display on simple hardware. Cell compression is able to achieve reasonable display quality on indexed color frame buffers. The initial focus of the Cell technology is for Sun-to-Sun communications, where the benefits of fast decode performance outweigh the benefits of standards.

The Cell encoding process transforms individual video frames into a bytestream that can be displayed with the Cell decompressor. In the first step of the encoding process, the synthetic (or filtered) video images are analyzed to produce an appropriate colormap to represent the frames to be encoded. This step allows the specification of the colormap size, in order to leave colors unused. This enhances cooperation with the window manager and other applications. Cell also provides a dynamic colormap strategy in which a new colormap is generated after each frame is compressed. This map can be used in subsequent frames.

Choosing a Colormap

The compressor chooses which colormap to encode the current image in one of three ways. If Adaptive Colormap Selection (ACS) is enabled, and a new colormap has not been associated with the compressor since the last call to **xil_compress**(3), the adapted colormap is used. When ACS is disabled, the compressor always uses the colormap given by the COMPRESSOR_COLORMAP attribute, if it has been set. If the compressor does not have a colormap, either via the COMPRESSOR_COLORMAP attribute or ACS, the compressor calls **xil_choose_colormap**(3) to generate an optimal colormap for the image. To reset ACS, give the compressor a new colormap via the COMPRESSOR_COLORMAP attribute.

Image Types

The Cell compressor and decompressor, respectively, accept and produce 3-band images in RGB color space. The width and height of the images must be divisible by 4.

Creating a Cell CIS

To compress a compressed image sequence (CIS) with the XIL Cell compressor, specify "Cell" for the *compressorname* argument in **xil_cis_create**(3).

Getting and Setting Cell Attributes

Use **xil_cis_get_attribute**(3) and **xil_cis_set_attribute**(3) to get and set Cell CIS attributes. These attributes are described in the following sections. Refer to the example section for additional information.

Cell Compression Attributes

The following paragraphs describe the Cell CIS attributes available with the XIL library. All structures and enumerations are defined via **xil.h.** Note that some attributes are "set-only" and others are "get-only." This is noted under the *Access* heading for each attribute.

Note that if you are setting an attribute and that attribute is a structure, you must pass the address of that structure. If you are getting an attribute, you always pass its address.

ENCODING_TYPE

Description	Specifies encoding algorithm
Access	get and set
Type	typedef enum { BTC, DITHER } XilCellEncodingType;
Values	*DITHER:* Use the dither encoding technique, which chooses two colors and a mask that produces the least amount of error when dithered across the 4x4 region. By selecting dither encoding, Adaptive Colormap Selection (ACS) is disabled. The current value of the COLORMAP_ADAPTION attribute is ignored. *BTC:* Use Block Truncation Coding to selection the two colors and the mask. This is much faster than dither encoding and produces good results.
Default	*BTC*

TEMPORAL_FILTERING

Description	Turns on or off a form of temporal filtering that helps with compression interframe encoding.
Access	get and set
Type	Xil_boolean
Values	*TRUE:* Filtering turned on *FALSE:* Filtering turned off
Default	TRUE

COMPRESSOR_COLORMAP

Description	Associates a colormap with the compressor for encoding images.
Access	set-only
Type	XilLookup
Default	*NULL*

COLORMAP_ADAPTION

Description	Enables or disables Adaptive Colormap Selection (ACS). ACS selects a colormap for the next image that is visibly indistinguishable from the colormap used with the current image. Thus, ACS continually adapts the colormap so it does not change enough between two images to cause colormap flashing when the bytestream is decoded.

	ACS detects when an adapted colormap has too much error and encodes new colormaps until the colormaps closely match the optimal colormap for the image. So, when ACS is enabled, every frame may have a new colormap associated with it.
Access	get and set
Type	Xil_boolean
Values	*TRUE/FALSE*
Default	*TRUE*
Notes	ACS is disabled when using dither encoding.

KEYFRAME_INTERVAL

Description	Specifies the interval for encoding key frames in the bytestream. A key frame has a bytestream information header, a repeated colormap, and uses no interframe escape codes. If KEYFRAME_INTERVAL is set to 0, then no key frames are encoded in the resulting Cell bytestream, and bit-rate control is disabled.
Access	get and set
Type	int
Default	*6*

BITS_PER_SECOND

Description	The bit rate of the resulting Cell bytestream. The rate is guaranteed over a single frame group. If BITS_PER_SECOND is set to 0, then bit rate control is disabled; this is the default. If BITS_PER_SECOND is set to a rate lower than the compressor can produce, then an error is generated, and bit rate control is disabled.
Access	get and set
Type	int *
Default	*0*

COMPRESSOR_MAX_CMAP_SIZE

Description	Sets the maximum colormap size that will be encoded in the Cell bytestream. If COLORMAP_ADAPTION is enabled, this attribute limits the size of the colormaps produced by the compressor. If COLORMAP_ADAPTION is disabled, this attribute limits the size of the colormaps with the COMPRESSOR_COLORMAP attribute. If the compressor is given a colormap that is larger than COMPRESSOR_MAX_CMAP_SIZE , it will be truncated to this length.

The value of this attribute is passed in the Cell bytestream for retrieval with the DECOMPRESSOR_MAX_CMAP_SIZE attribute as an aid to X colormap management.

This attribute can only be set before the first **xil_compress**(3) call. After **xil_compress**(3) has been called or COMPRESSOR_MAX_CMAP_SIZE has been set, it cannot be changed for the life of the XilCis.

Access	get and set
Type	int
Default	*256*

COMPRESSOR_FRAME_RATE

Description	Set the frame rate, in microseconds per frame, at which the images were captured. This value is passed in the Cell bytestream for retrieval with the DECOMPRESSOR_FRAME_RATE attribute. It is permissible to change this attribute in between calls to **xil_compress**(3).
Access	set-only
Type	Xil_unsigned32
Default	*33333 (30 frames/second)*

COMPRESSOR_USER_DATA

Description: Set the user data to be encoded with the next frame. This attribute clears itself after every call to **xil_compress**(3), so it only affects the very next call to **xil_compress** (). A copy of the data is made when setting this attribute, so no assumptions are made about the validity of the data pointer after the attribute is set. The given data is encoded into the Cell bytestream, making the data available to a decompressor via the DECOMPRESSOR_USER_DATA attribute. The attribute accepts a pointer to XilCellUserData, which is a structure containing a pointer to the data and the length of the data. The length of the data is limited to 8K (8192 bytes) per frame. It is permissible to change this attribute in between calls to **xil_compress**(3).

Access: set-only

Type:

```
typedef struct {
        Xil_unsigned8*   data;
        Xil_unsigned32   length;
} XilCellUserData;
```

Default: Not set

Cell Decompression Attributes

DECOMPRESSOR_COLORMAP

Description	In the case of set, give the Cell decompressor a look-up table with which to perform accelerated 8-bit display of the decompressed image when using **xil_nearest_color**(3). All colormap indices are assumed to be read-only by the decompressor (see RDWR_INDICES). In the case of get, it returns the look-up table associated with the Cell decompressor. This table could possibly have been modified by a call to **xil_decompress**(3). If this attribute has not been set, then it returns NULL.
Access	get and set
Type	XilLookup
Default	*Not set*

RDWR_INDICES

Description	Set the list of colormap indices in the DECOMPRESSOR_COLORMAP look-up table that the Cell decompressor can change for optimum display of decompressed images. The DECOMPRESSOR_MAX_CMAP_SIZE attribute can be used to determine the number of colormap entries needed for optimum display. Setting the list is not cumulative; the list from any previously set attribute call is discarded. Any indices outside the range of the DECOMPRESSOR_COLORMAP look-up table are discarded. Entries in the lookup are only changed on a call to **xil_decompress**(3). If you set this attribute, the Cell decompressor assumes that after each call to **xil_decompress**(3), you will check to see if the XilLookup has been changed via **xil_lookup_get_version**(3), and if so, that you will install the changed colormap before calling **xil_nearest_color**(3) with the XilLookup. Refer to the *XIL Programmer's Guide.* for more details.
Access	set-only
Type	typedef struct { 　　Xil_unsigned32* pixels; 　　Xil_unsigned16 ncolors; } XilIndexList;
Default	*Not set*

DECOMPRESSOR_MAX_CMAP_SIZE

Description	Get the maximum size of a colormap for this Cell bytestream. This assists in X colormap management when decompressing the bytestream. Refer to the example in the *XIL Programmer's Guide.* for more information.
Access	get-only
Type	int
Default	*256*

DECOMPRESSOR_FRAME_RATE

Description	Get the frame rate, in microseconds per frame, at which the images were captured. This value is stored in the Cell bytestream via the COMPRESSOR_FRAME_RATE attribute, and is useful only when the compressed image sequence represents a movie. This attribute may have different values at various points in the Cell bytestream if the COMPRESSOR_FRAME_RATE attribute was changed during the creation of the compressed image sequence. If the Cell bytestream does not contain a frame rate, the default value (33333) is returned.
Access	get-only
Type	Xil_unsigned32
Default	*33333 (30 frames/second)*

DECOMPRESSOR_USER_DATA

Description	Get the user data that may be encoded with the most-recently decompressed frame. This attribute clears itself after every call to **xil_decompress**(3), so the returned data is only valid until the next call to **xil_decompress** (). The data decoded from the Cell bytestream was encoded via the COMPRESSOR_USER_DATA attribute. A *pointer* to XilCellUserData is returned.
Access	get-only
Type	XilCellUserData*
Default	Not set

EXAMPLES

The following example opens and closes a Cell CIS using the XIL library:

```
XilSystemState State;
XilCis cis;
State = xil_open();
cis = xil_cis_create(State, "Cell");

-- calls to Cell-specific compression routines --

xil_cis_destroy(cis);
xil_close(State);
```

The following example sets a Cell CIS attribute called *TEMPORAL_FILTERING* to TRUE. Note that because this attribute is not a structure, it is not necessary to pass the address of *TEMPORAL_FILTERING* when setting it.

```
XilCis cis;

xil_cis_set_attribute(cis,"TEMPORAL_FILTERING", (void *) TRUE);
```

The following example returns the value of a Cell CIS attribute called *TEMPORAL_FILTERING.* Note that when getting an attribute it is always necessary to pass the address.

```
Xil_boolean encode_type;
XilCis cis;

xil_cis_get_attribute(cis, "TEMPORAL_FILTERING", (void **) &encode_type);
```

NOTES

The **xil_cis_set_attribute** () and **xil_cis_get_attribute** () calls are used to modify the default behavior of a specific compressor. Generic attributes of compressors are set by individual function calls.

SEE ALSO

xil_cis_create(3), **xil_cis_get_attribute**(3), **xil_cis_get_bits_ptr**(3), **xil_compress**(3), **xil_decompress**(3), **xil_choose_colormap**(3), **xil_lookup_get_version**(3), **xil_nearest_color**(3).