

## RTP Payload Format of Sun's CellB Video Encoding

### Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

### Abstract

This memo describes a packetization scheme for the CellB video encoding. The scheme proposed allows applications to transport CellB video flows over protocols used by RTP. This document is meant for implementors of video applications that want to use RTP and CellB.

### 1. Introduction

The Cell image compression algorithm is a variable bit-rate video coding scheme. It provides "high" quality, low bit-rate image compression at low computational cost. The bytestream that is produced by the Cell encoder consists of instructional codes and information about the compressed image.

For further information on Cell compression technology, refer to [1]. Currently, there are two versions of the Cell compression technology: CellA and CellB. CellA is primarily designed for the encoding of stored video intended for local display, and will not be discussed in this memo.

CellB, derived from CellA, has been optimized for network-based video applications. It is computationally symmetric in both encode and decode. CellB utilizes a fixed colormap and vector quantization techniques in the YUV color space to achieve compression.

## 2. Network Packetization and Encapsulation

### 2.1 RTP Usage

The RTP timestamp is in units of 90KHz. The same timestamp value is used for all packet payloads of a frame. The RTP maker bit denotes the end of a frame.

### 2.2 CellB Header

The packetization of the CellB bytestream is designed to make the resulting packet stream robust to packet loss. To achieve this goal, an additional header is added to each RTP packet to uniquely identify the location of the first cell of the packet within the current frame. In addition, the width and height of the frame in pixels is carried in each CellB packet header. Although the size can only change between frames, it is carried in every packet to simplify the packet encoding.

0								1								2								3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Cell X Location								Cell Y Location																							
Width of Image								Height of Image																							
Compressed CellB Data																															
....																															

All fields are 16-bit unsigned integers in network byte order, and are placed at the beginning of the payload for each RTP packet. The Cell X and the Cell Y Location coordinates are expressed as cell coordinates, not pixel coordinates. Since cells represent 4x4 blocks of pixels, the X or Y dimension of the cell coordinates range in value from 0 through 1/4 of the of the same dimension in pixel coordinates.

### 2.3 Packetization Rules

A packet can be of any size chosen by the implementor, up to a full frame. All multi-byte codes must be completely contained within a packet. In general, the implementor should avoid packet sizes that result in fragmentation by the network.

### 3. References

1. "Cell Image Compression Byte Stream Description,"  
<ftp://playground.sun.com:/pub/multimedia/video/cellbytestream.ps.Z>
2. Turletti, T., and C. Huitema, "RTP Payload Format for H.261 Video Streams", RFC 2032, October 1996.
3. Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, January 1996.
4. Schulzrinne, H., "RTP Profile for Audio and Video Conferences with Minimal Control", RFC 1890, January 1996.

### 4 Authors' Addresses

Michael F. Speer  
Sun Microsystems Computer Corporation  
2550 Garcia Ave MailStop UMPK14-305  
Mountain View, CA 94043

Voice: +1 415 786 6368  
Fax: +1 415 786 6445  
EMail: [michael.speer@eng.sun.com](mailto:michael.speer@eng.sun.com)

Don Hoffman  
Sun Microsystems Computer Corporation  
2550 Garcia Ave MailStop UMPK14-305  
Mountain View, CA 94043

Voice: +1 415 786 6370  
Fax: +1 415 786 6445  
EMail: [don.hoffman@eng.sun.com](mailto:don.hoffman@eng.sun.com)

## Appendix A - Structure of the CellB Video Stream

The CellB bytestream consists of cell codes, skip codes and quantization-table specific codes. These are now described.

### A.1 CellB Cell Code

Cell codes are 4 bytes in length, and describe a 4x4 pixel cell. There are two possible luminance (Y) levels for each cell, but only one pair of chrominance (UV) values. The CellB cell code is shown below:

0	1	2	3						
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9						
+-----+-----+-----+-----+									
0 M M M M M M M M M M M M M M M M									
+-----+-----+-----+-----+									
U V U V U V U V   Y Y Y Y Y Y Y Y									
+-----+-----+-----+-----+									
4x4 Bitmask                      U/V Code                      Y/Y Code									

The first two bytes of the cell code are a bitmask. Each bit in the mask represents a pixel in a 16-pixel cell. Bit 0 represents the value of the upper right-hand pixel of the cell, and subsequent bits represent the pixels in row-major order. If a pixel's bit is set in the 4x4 Bitmask, then the pixel will be rendered with the pixel value <Y(1), U, V>. If the pixel's bit is not set in the bitmask, then the pixel's value will be rendered with the value <Y(0), U, V>. The bitmask for the cell is normalized so that the most significant bit is always 0 (i.e., corresponding to <Y(0), U, V>).

The U/V field of the cell code represents the chrominance component. This code is in index into a table of vectors that represents two independent components of chrominance.

The Y/Y field of the cell code represents two luminance values (Y(0) and Y(1)). This code is an index into a table of two-component luminance vectors.

The derivation of the U/V and Y/Y tables is outside the scope of this memo. A complete discussion can be found in [1].

## A.2 CellB Skip Code

The single byte CellB skip code tells the CellB decoder to skip the next S+1 cells in the current video frame being decoded. The maximum number of cells that can be skipped is 32. The format of the skip code is shown below.

```

      0 1 2 3 4 5 6 7
+---+---+---+---+---+
|1 0 0 S S S S S|
+---+---+---+---+---+

```

## A.3 CellB Y/Y Table Code

The single byte "new Y/Y table" code is used to tell the decoder that the next 512 bytes are a new Y/Y quantization table. The code and the representation of the table are shown below. The sample encoder/decoder pair in this document do not implement this feature of the CellB compression. However, future CellB codecs may implement this feature.

```

      0 1 2 3 4 5 6 7
+---+---+---+---+---+
|1 1 1 1 1 1 1 0|
+---+---+---+---+---+

```

The format of the new Y/Y table is:

```

      0                               1                               2                               3
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Y1_000   |   Y2_000   |   Y1_001   |   Y2_001   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
.
.
.

```

```

      0                               1                               2                               3
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Y1_254   |   Y2_254   |   Y1_255   |   Y2_255   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

#### A.4 CellB U/V Table Code

The single byte "new U/V table" code is used to tell the decoder that the next 512 bytes represent a new U/V quantization table. The code is shown below. The sample encoder/decoder pair provided in this document do not implement this feature of the CellB compression. However, future CellB codecs may implement this feature.

```

      0 1 2 3 4 5 6 7
    +---+---+---+---+
    |1 1 1 1 1 1 1 1|
    +---+---+---+---+

```

The bytes of the new U/V quantization table are arranged as:

```

      0                               1                               2                               3
    0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   U_000   |   V_000   |   U_001   |   V_001   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
.
.
.

      0                               1                               2                               3
    0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   U_254   |   V_254   |   U_255   |   V_255   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

#### Appendix B - Availability of CellB

It is the viewpoint of Sun Microsystems, Inc, that CellB is publically available for use without any license.

