High-throughput Video Data Transfer of Striping with SSDs for 8K Super Hi-Vision

Takeshi KAJIYAMA, Kodai KIKUCHI and Eiichi MIYASHITA

We have developed a high-throughput recording and playback method for full-featured 8K Super Hi-Vision (8K) video whose data rate is up to 2.4 GBps. The method fully exploits hardware with a specific design for handling large amounts of data, sequential transfer, and frame unit random reads including fast-forward, rewind and jog-shuttle. Steady recording and playback, including frame unit random reads, were demonstrated on an 8K recorder and player.

1. Introduction

8K Super Hi-Vision (8K)\(^1\)\(^-\)\(^5\) is an ultrarealistic video and audio system consisting of ultrahigh-definition (UHD) images with 33 megapixels, 16 times that of HD, and 22.2 multichannel sound. The throughput required for full-featured, uncompressed 8K video with a frame rate of 120 Hz, quantization of 12 bits, and a full resolution of 7,680 x 4,320 in green, and blue (RGB) is 18 GBps\(^6\) (144 Gbps). We have previously developed compressed recording equipment using video compression with high-speed signal recording\(^6\)\(^-\)\(^7\). This equipment used low compression rates (for high-image quality) so that it could be used for editing and other program production work within broadcast stations, resulting in after-compression transmission rates of 2.4 GBps (approx. 20 Gbps) for compressed video data and audio data in 4:4:4 format\(^2\) and 1.2 GBps (approx. 10 Gbps) for content in 4:2:0 format\(^3\). Thus, high-throughput recording and playback memory is required.

Solid-state drives (SSDs)\(^8\), which have higher throughput than hard disk drives (HDDs), are attractive as memory for video data at the above rates. SSDs use NAND flash memory\(^9\)\(^-\)\(^4\), a solid-state, non-volatile memory, as the recording medium. They are suitable for memory that needs to be impact-resistant and portable with removability.

However, ordinary SSDs are limited by the Serial ATA\(^5\)\(^-\)\(^9\) (SATA) interface specifications to a maximum throughput of 0.6 GBps, so they cannot be used individually to record compressed 8K.

A technology called redundant arrays of inexpensive disks (RAIDs)\(^1\)\(^0\)\(^-\)\(^1\)\(^1\), which parallelizes memory, can be used effectively to implement 8K compressed recording using SSDs. Several levels of RAID are defined, depending on the desired recording and playback throughput, recording capacity, and reliability. One of these levels (RAID 0) is called striping, and stores data partitioned into \(N\) parts on \(N\) separate memories. This has no redundancy, but it has \(N\) times the recording capacity and theoretically can achieve \(N\) times the recording and playback throughput.

However, when RAID is actually applied, it is difficult to achieve both sequential transfer\(^6\) and random transfer\(^7\) performance. In particular, ordinary products incorporating RAID have a strong tendency to emphasize performance for random transfer, and when used for video data recording, which requires high sequential transfer performance, the expected performance is not obtained. Here we propose striping technology and data transfer control methods that can improve the sequential throughput for video with high data rates. This article describes a high-throughput recording technology with a throughput of up to 2.4 GBps and up to 20 MB per frame, which considers various data access patterns including sequential writing for recording, sequential reading for playback, and random reading of individual frames for special playback modes such as fast forward, rewind, and using a jog shuttle.

2. Striping technology

This section first discusses examples of striping configurations and performance evaluation. It then explains the causes of throughput bottleneck as the number of parallel SSDs increases, in light of the evaluation results, and de-

\(^{1}\) Bps is a unit indicating the amount of data, in bytes, that can be transmitted in one second.

\(^{2}\) A high-quality video format that records all color components of the video without chroma subsampling.

\(^{3}\) A recording format that subsamples color components of the video. The red and blue components have only 1/4 of the original amount of information.

\(^{4}\) A type of non-volatile solid-state memory that is inexpensive and has high capacities as compared to other non-volatile semiconductor memories. Writing and erasing is fast but overwriting individual bytes is slower due to the required complicated procedure inside.

\(^{5}\) A high-speed serial interface for connecting computer and memory. Gen2, the second-generation specification, is capable of transmission up to 0.375 GBps (0.3 GBps in practice), and Gen3, the third-generation specification, is capable of transmission up to 0.75 GBps (0.6 GBps in practice).

\(^{6}\) A data access pattern in which data is written or read from consecutive addresses.

\(^{7}\) A data access pattern in which data is written or read from addresses that are not consecutive.
scribes our basic concept for improving throughput.

2.1 Examples of striping configurations

Striping control can be divided into hardware striping and software striping.

An example of a hardware striping configuration is shown in Fig. 1. Hardware striping is provided by a dedicated RAID board and can provide high throughput through a dedicated large-scale integration (LSI) RAID 0 controller, but the maximum number of SSDs that can be used is determined by the product. Generally, the host interface is PCIe 12*8 and the SSD interface is SATA or Serial Attached SCSI (SAS)*9.

An example of a software striping configuration is shown in Fig. 2. With software striping, the RAID 0 controller operates on the host in the software, so throughput depends on the CPU performance and load, but the number of SSDs connected can be increased by using expansion boards that convert the PCIe interface to several SATA or SAS interfaces.

In the next section, we discuss the results of investigating the types of transfer bottlenecks that occur and the different characteristics of these two types of striping control.

2.2 Evaluating striping performance

The theoretical throughput expected with the striping of \( N \) parallel SSDs is \( N \) times the individual throughput, but in fact, there is a tendency toward saturation as \( N \) increases.

To examine this phenomenon in detail, we measured the throughput for the hardware striping configuration shown in Fig. 1 using three different SSDs (labeled A, B, and C) from different manufacturers and with different capacities and connector types. We used a RAID board that can accommodate up to eight SSDs. The interface between the host personal computer (PC) and the RAID board was PCIe with the Gen2 x 8 lane throughput bandwidth (i.e., the maximum throughput) of 4 GBps. The interfaces between the RAID board and SSDs were SATA with the bandwidth per SSD of Gen3, i.e., 0.6 GBps. The total throughput bandwidth for eight SSDs was thus 4.8 GBps. The specifications of the hardware used for the measurements are shown in Table 1.

The relationships between the number of parallel SSDs, \( N \), and the sequential write and sequential read throughputs for hardware striping are shown in Figs. 3 and 4, respectively. The solid lines are the measured values, while the dotted lines are the theoretical values.

![Figure 1: Example of hardware striping configuration](image1)

![Figure 2: Example of software striping configuration](image2)

<table>
<thead>
<tr>
<th>Table 1: Hardware specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host PC</td>
</tr>
<tr>
<td>RAID board</td>
</tr>
<tr>
<td>Expansion board</td>
</tr>
<tr>
<td>SSD A</td>
</tr>
<tr>
<td>SSD B</td>
</tr>
<tr>
<td>SSD C</td>
</tr>
</tbody>
</table>

*1 A smaller SATA connector than that usually used for SATA interfaces.
lines are the theoretical values, derived by multiplying the individual SSD throughput by $N$. The results in Fig. 3 show that for all three SSD brands, the throughput increased up to $N=4$ but then saturated for $N=8$, yielding results much lower than the theoretical value. Figure 4 shows similar results for reading, with saturation when $N=8$. The maximum throughput for both writing and reading was under 2 GBps, which is less than half of the maximum PCIe throughput of 4 GBps (8 x 0.5 GBps). Also, for SSD B, the throughput for both writing and reading when $N=8$ was actually less than the throughput for $N=4$, suggesting that the cause was not only the performance limits of the RAID board hardware but was also related to the number of parallel devices, $N$.

Similarly, we measured the throughput for software striping with the configuration shown in Fig. 2. We used an expansion board that can connect up to 16 SSDs. The interface between the host PC and expansion board was PCIe Gen2 x 8 lanes, with a bus bandwidth of 4 GBps, the same as in the hardware striping measurements. The interfaces between the expansion board and SSDs were SATA Gen3 (bus bandwidth 0.6 GBps) with a maximum transfer bandwidth of 9.6 GBps when 16 SSDs are connected. The relationships between the numbers of parallel SSDs, $N$, used with software striping and the sequential write and read throughputs are shown in Figs. 5 and 6, respectively. Solid lines are measured values and dotted lines are theoretical values, derived by multiplying the individual SSD throughput by $N$. Figure 5 shows that the measured write throughput for $N=8$ was much lower than the theoretical value, and for $N=16$ it was mostly saturated. The maximum write throughput for SSD B was about 2.2 GBps when $N=16$ but was less than 2 GBps for both SSD A and SSD C; all these values were approximately half of the maximum throughput for PCIe of 4 GBps. These results, and the fact that the saturation rates for each case were different, suggest that the cause of the bottleneck was not simply hardware performance limitations. Figure 6 also shows that the read throughput was much less than the theoretical value for $N=8$ and that the maximum throughput for $N=16$ under saturation was below 2 GBps for all three drive brands.

### 2.3 Factors causing throughput bottlenecks

There are two possible factors resulting in the bottlenecks that cause the throughput saturation discussed in the previous section.

The first factor is a limitation on the hardware operating speed, which is the operating speed of the RAID board for hardware striping, and the operating speeds of the expansion board and PC CPU for software striping. These factors depend on specifications such as the hardware operating frequency and bus bandwidth, so such bottlenecks need to be eliminated at the hardware design stage to achieve the desired performance.

The second factor is the transmission overhead for the host interface (PCIe) and SSD interfaces (SATA), and is
common to both hardware and software striping. One way to deal with this factor is to increase the hardware performance to 1.5 times or double, so that the transfer is possible even with the overhead, but when processing high-speed video data such as 8K, increasing the operating frequency and other aspects is not practical. Thus, to avoid bottlenecks, it is important to eliminate the transmission overhead.

3. Analyzing video data transmission characteristics and improving transmission speed

In this section we give the results of analyzing the relationship between throughput and data size, assuming a very large amount of video data, with the goal of eliminating PCIe and SATA interface bottlenecks. Such bottlenecks have been thought to be a cause of the drop in throughput for both hardware and software striping. From the results of this analysis, we describe a data transmission control method for eliminating the transmission overhead and improving throughput.

To investigate the relationship between data size and throughput, we built the test board shown in Fig. 7. To eliminate bottlenecks due to the speed of the host PC CPU in our analysis, the test board uses hardware striping. The test board has three field-programmable gate arrays (FPGA), and the low-voltage differential signaling (LVDS) on the internal bus maintains a total bandwidth of 4 GBps. The host interface is the same as that in Figs. 1 and 2, PCIe Gen2 x 8 lanes. The SSD interfaces are SATA Gen2 with up to 0.3 GBps bandwidth and there are connections for up to 16 SSDs. The test board had mSATA connectors, which are smaller than regular SATA connectors, and SSD C devices with mSATA connectors were used in the remainder of this analysis.

3.1 Analysis of PCIe transmission characteristics

PCIe provides direct memory access (DMA) for the efficient transfer of large-volume data such as video. Figure 8 is a diagram explaining the transmission time for a single frame of video data. $R$ in the figure is the number of DMA transfers needed to transfer one frame of video data, $T_{\text{start}}$ is the latency for starting a DMA transfer, $T_{\text{hd}}$ is the packet header transfer time, and $T_{\text{data}}$ is the packet data transfer time. Figure 8 suggests that we can improve the overall transmission time by reducing $T_{\text{start}}$, but $T_{\text{start}}$ depends on the hardware performance of the host controlling the PCIe bus and the board, so this approach would be difficult. Thus, we attempted to increase the data rate by increasing the data size of each DMA transfer, thereby reducing $R$ and the number of times $T_{\text{start}}$ is needed.

To study this effect, we measured the relationship between data size and throughput, we built the test board shown in Fig. 7. To eliminate bottlenecks due to the speed of the host PC CPU in our analysis, the test board uses hardware striping. The test board has three field-programmable gate arrays (FPGA), and the low-voltage differential signaling (LVDS) on the internal bus maintains a total bandwidth of 4 GBps. The host interface is the same as that in Figs. 1 and 2, PCIe Gen2 x 8 lanes. The SSD interfaces are SATA Gen2 with up to 0.3 GBps bandwidth and there are connections for up to 16 SSDs. The test board had mSATA connectors, which are smaller than regular SATA connectors, and SSD C devices with mSATA connectors were used in the remainder of this analysis.

When transmitting data in computers and other devices, processing and procedures are required, in addition to the transmitted data itself.

---

*10 When transmitting data in computers and other devices, processing and procedures are required, in addition to the transmitted data itself.

*11 The delay time needed from a data transmission request until the result is returned.
between the DMA transfer data size and PCIe throughput using the test board. The results are shown in Fig. 9. These measurements were taken between the test board and the main memory\textsuperscript{12} of the host PC. The DMA transfer data size is shown on the horizontal axis and the PCIe throughput for sequential transfer is shown on the vertical axis. Figure 9 shows that for a data size of 2 MB, the throughput did not reach 3 GBps, but as the data size increased, so did the throughput, until it surpassed 3.4 GBps at 20 MB. This shows that the throughput can be increased by increasing the DMA transfer data size.

3.2 Analysis of SATA transmission characteristics

We also studied how to improve the throughput for the SATA interfaces in the same way as for the PCIe throughput by increasing the transfer size. SSDs manage data internally in 512-byte units called sectors\textsuperscript{13}, and the number of sectors transferred through the SATA interface for each transfer command can be selected. Thus, we measured the relationship between the number of sectors transferred with each transfer command and the sequential write and read throughputs. The results are shown in Fig. 10. These measurements were carried out using the test board and SSD C devices from Section 2.2. The number of sectors is on the horizontal axis and the throughput for a single SSD is on the vertical axis. From Fig. 10, we see that for both writing and reading, the throughput increases as the number of sectors increases up to 1024 sectors, and thereafter it increases more slowly. In these tests, the highest throughput was achieved for 3,072 sectors, which was 0.215 GBps for writing and 0.262 GBps for reading. These results show that high throughput can be obtained by setting the number of sectors for each transfer command to greater than 1,024.

3.3 Evaluation of random read rates

In this section, we verify the relationship between the random read data size and throughput. When recording, video data is exclusively written with sequential transfer, but when reading, random transfer in frame units is also used. It is also known that the SSD performance is generally lower for random transfer than for sequential transfer. For these reasons, we evaluated the SSD random read performance.

The results of measuring the random read performance using SSD C are shown in Fig. 11. The data size of random read operations is shown on the horizontal axis and the throughput is shown on the vertical axis. The random read throughput increases quickly with the data size up to near 1,024 kB, and then continues to increase slowly. This result suggests that by ensuring data sizes of at least 1,024 kB, good random read performance can be attained.

3.4 Video data transmission control to improve transmission throughput

Below, we describe a video data transfer control method utilizing the results above to improve throughput.

As discussed earlier, the throughput on PCIe and SATA interfaces increases as the amount of transferred data increases, but with video data, random reads in single frame units must also be considered for special playback operations such as fast forward, rewind, and use of the jog shuttle. For these cases, video data must be managed in units that are the same size of or less than the video frame size. Accordingly, we propose a video data transfer method that can

\textsuperscript{12}Main memory. Extremely fast relative to HDD or SSD but volatile and smaller in capacity.

\textsuperscript{13}The smallest unit of recording in a recording medium. Used mainly for disk-based media but also used for SSDs, which have interfaces that are compatible with HDDs.
handle data in video frame units.

To maximize the PCIe throughput, we set the data size transferred between the host and the test board to equal the video frame size, and call it $D_f$. To maximize the SATA throughput, we set the data size transferred between the test board and each of the SSDs, $D_s$, by dividing $D_f$ by the number of parallel SSDs, $N$, so that $D_s = D_f/N$. Thus, the number of sectors corresponding to $D_s$ is given by $S = D_s/512$, which we take as the maximum number of sectors transferrable through the SATA interfaces in a single command.

The results of evaluating throughput using sample values for these parameters are summarized in Table 2. Parameters that yielded high throughput are indicated with a circle in Table 2.

Bottlenecks in the PCIe and SATA interfaces can be reduced by controlling transfers according to the parameters shown in Table 2. That is, as shown by the results in Fig. 9, the larger the PCIe DMA transfer size, the more the bottleneck is eliminated and the higher the throughput achieved. By making the DMA transfer size close to the frame size, $D_f$, the bottleneck was eliminated. Specifically, with $D_f=20$ MB, over 3.4 GBps was achieved, and with $D_f=10$ MB, over 3 GBps was achieved. The results in Fig. 10 show that the bottleneck can be avoided by setting the number of sectors transferred per SATA command to greater than 1,024. Note here that the number of sectors transferred, $S$, must be inversely proportional to the number of parallel memories, $N$, and under the conditions shown in Table 2, the number of transferred sectors, $S$, is 1,024 or greater, so bottlenecks can be eliminated.

For random reads, we can set the parallelism, $N$, such that the $D_f$ value does not drop below 1,024 kB to achieve high throughput. For the parameters shown in Table 2, when $D_f = 20$ MB with parallelism $N = 16$, and when $D_f = 10$ MB with parallelism $N = 8$, $D_s$ was 1,024 or greater, resulting in good performance, but when $D_f = 10$ MB and $N = 16$, the random read performance was lower.

4. Evaluating the transmission speed of striping with the proposed method

We applied the proposed method using the test board in Fig. 7 and up to 16 SSD C devices, and evaluated the throughput between the host PC and the striped SSDs.

Figure 12 shows results of measuring the relation between the striping parallelism, $N$, and write throughput for a video frame size of $D_f = 20$ MB. The dotted line shows the theoretical values, obtained by multiplying the individual SSD transfer rate by $N$, and the solid line shows the measured results. The write throughput increased almost proportionally with $N$, achieving 3.211 GBps when $N = 16$. Figure 13 shows the results of measuring the read throughput under the same conditions. The read throughput also increased with $N$, achieving 3.118 GBps at $N = 16$. These results show that the proposed method reduced the transfer overhead, achieving write throughput nearly proportional to $N$ up to $N = 16$. Similarly, the method achieved read throughput comparable to the write values, which, while dropping somewhat below the theoretical throughput, continued to increase up to $N = 16$.

One possible reason that the read throughput in Fig. 13 dropped below the theoretical value is as follows. Generally

<table>
<thead>
<tr>
<th>Video frame size $D_f$ and PCIe DMA transfer rate</th>
<th>Number of parallel SSDs ($N$)</th>
<th>$D_f$ (kB)</th>
<th>Throughput</th>
<th>$S$ (no. of sectors)</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_f$ (MB)</td>
<td>Throughput</td>
<td>$D_f$ (kB)</td>
<td>Throughput</td>
<td>$S$ (no. of sectors)</td>
<td>Throughput</td>
</tr>
<tr>
<td>20</td>
<td>○</td>
<td>16</td>
<td>1,280</td>
<td>○</td>
<td>2,560</td>
</tr>
<tr>
<td>10</td>
<td>○</td>
<td>8</td>
<td>2,560</td>
<td>○</td>
<td>5,120</td>
</tr>
</tbody>
</table>

Table 2: Transmission control parameters and transmission throughput

Figure 12: Number of parallel SSDs and write transmission throughput using the proposed method

Figure 13: Number of parallel SSDs and read transmission throughput using the proposed method
an upper limit for the throughput for $N$ striped SSDs is $N$ times the throughput of the slowest of the $N$ SSDs. Thus, as $N$ increases, the probability that the transfer rate will drop increases. In particular, SSDs have sufficient cache memory to maintain stable write transfer rates, but many products cannot use this cache for reading, and this tendency may be quite noticeable.

For these measurements, we used SSD C because we were constrained by the type of connector on the test board, but see similar results are expected for SSDs A and B, which have similar or better individual device performance.

5. Implementation and evaluation in an 8K recording system

We implemented the proposed method in a prototype 8K recording equipment and evaluated its recording and playback performance. We compressed full-featured 8K video with a data rate of 1.2 GBps, a video frame size of 10 MB, and 4:2:0 format. To record and play back this video data, we used the test board and eight SSD C devices in parallel. Previously, we measured the performance of this configuration using the host PC as shown in Fig. 7 with parallel. We also confirmed that the proposed method is useful for recording, regular playback, and special types of playback for 1.2 GBps compressed video data using 8K recorder with eight parallel SSD devices.

This article was edited and revised on the basis of the following article, which appeared in ITE Transactions on Media Technology and Applications.


References