

Reprint

Disparity Codec Architecture for 3D Teleconferencing Systems

D. Papadimatos, T. Antonakopoulos and V. Makios

International Journal of Electronics

VOL. 89 No. 5, 2002, pp. 347-363

Copyright Notice: This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted or mass reproduced without the explicit permission of the copyright holder.

Disparity codec architecture for 3D teleconferencing systems

DIONYSIOS PAPADIMATOS†,
THEODORE ANTONAKOPOULOS*† and VASSILIOS MAKIOS†

This paper presents the architecture and the design methodology of an advanced and flexible codec optimized for disparity information compression and transmission over ATM constant bit rate channels. For this development we used a three-stage procedure, which benefits from the utilization of formal design methods and the reuse of its functional blocks for a more efficient and flexible implementation at both the encoder and the decoder side. The codec design provides the basis for defining and testing improved functionality that increases the performance of teleconferencing systems while allowing a migration path to other hardware implementations depending on which functionality needs to be optimized.

1. Introduction

The aim of the new generation of 3D videoconferencing systems is to provide the user with an enhanced illusion of true contact. Various means of achieving this make use of disparity information inherent in stereoscopic images i.e. images of the same scene taken by two cameras at slightly different positions. A 3D teleconferencing system using ATM as its communications link was built and tested successfully during the EU PANORAMA project (Ohm *et al.* 1998). The main components of this system, as shown in figure 1, are the disparity estimator with two camera inputs, two MPEG2 encoders, the disparity encoder, the synchronizer and the ATM multiplexer on the transmitter side, and two MPEG2 decoders, the disparity decoder the ATM demultiplexer, the 3D intermediate view interpolator and a special type of display called autostereoscopic display, which reproduces the stereoscopic illusion, at the receiving side. An ATM constant bit rate (CBR) channel is used for data transmission. To provide a sensation of natural contact, the stereoscopic display must emulate the slight changes in the visual scene the eyes see due to the difference in position of the head. The stereoscopic cameras, by virtue of their slightly different position, provide two image streams and the initial disparity data derived from these scenes. A person sitting in front of a teleconferencing set-up cannot be expected to sit totally still. The intermediate view interpolator uses the image information, the disparity information and the position of the viewer's head to generate the correct views.

Disparity information processing is an extremely processing power intensive task, so it is carried out on the transmitter side in the interest of reducing receiver complexity (Ohm *et al.* 1997). The transmission of the disparity information highlights the need for using compression techniques for efficient use of the communication channel, which is also used for compressed audio and video transmission.

Received 20 September 2000. Accepted 25 February 2002.

† Department of Electrical Engineering and Computers Technology, University of Patras, 26500 Rio-Patras, Greece.

* Corresponding author. e-mail: antonako@ee.upatras.gr

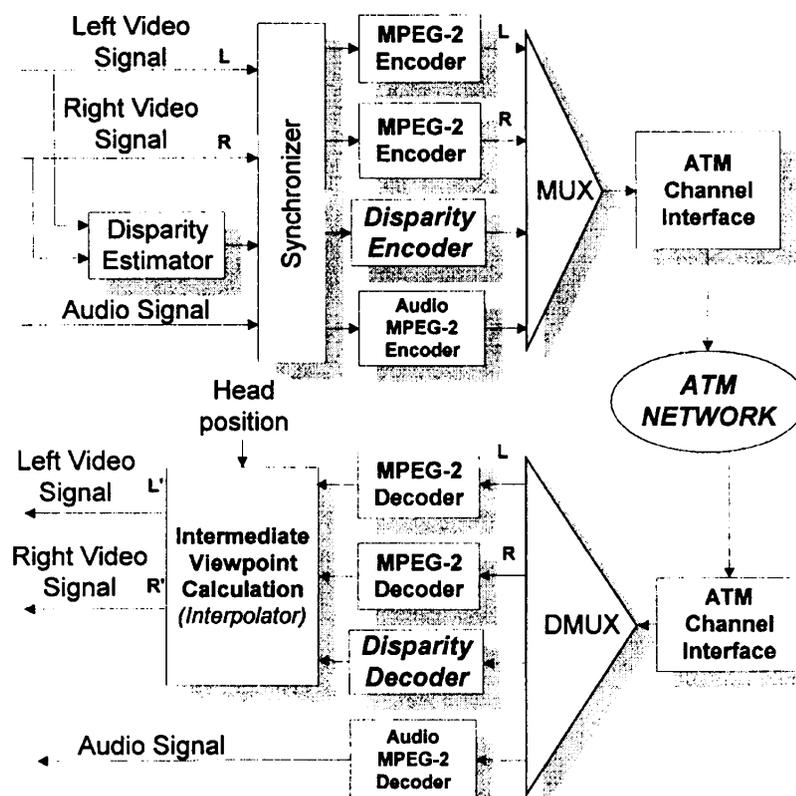


Figure 1. The 3D videoconferencing system set-up.

Various methods have been proposed for the efficient coding of disparity parameters (Tsovaras *et al.* 1996, Ziegler 1997). Some methods, such as DPCM, use prediction even though their vectors are associated with regions and not with blocks. In this case no spatial vector fields exist, resulting in inefficient coding. Lossy coding cannot be used since this would generate large synthesis errors and even the loss of the 3D impression (Ohm *et al.* 1977). The originally produced video and audio information of the PANORAMA system is compressed by commercially available MPEG2 codecs, but the specialized nature of the disparity information does not allow the use of the same hardware for disparity compression without introducing visible artefacts. In Ziegler (1997) the use of 2D block-based or 3D object-based methods has been investigated. These methods require the availability of motion information as a by-product of intensity image sequence coding. Such information was not available in the MPEG2 codecs and would have increased hardware complexity to a disproportionate extent. So a dedicated hardware platform was developed in order to process the disparity information in real-time. The technology developed in this work can easily be adapted to future stereoscopic consumer products using xDSL or cable modems for entertainment and educational purposes.

This paper presents the hardware architecture of such a high-speed disparity codec. The codec consists of a disparity encoder unit and a disparity decoder unit. The disparity encoder compresses the disparity data and conditions them for

multiplexing with the compressed video data streams and for transmission over the ATM network. The disparity codec uses a lossless compression algorithm which produces a variable bit-rate data stream, so a traffic-shaping algorithm has to be applied in order to limit the bit-rate to the rate available by the network (Papadimitos *et al.* 1998, 1999). The disparity decoder accepts the compressed data from the ATM demultiplexer and recreates the original disparity bit-stream.

In order to derive efficient hardware architecture, we carried out a detailed analysis of the requirements using the OMT formal design methodology (Rumbaugh *et al.* 1991). This methodology allows us to describe the static and dynamic aspects of the disparity codec graphically with the use of object models, SDL (specification and description language) notations and MSCs (message sequence charts). The steps we took to get to a successful development were the following:

- system requirements;
- system analysis and partitioning;
- hardware and microcode implementation.

The main goal of each step was to extract as much information as was needed for supporting the next step. All work was oriented to making the final design achieve the required functionality while utilizing the most suitable solutions in terms of hardware and software. A design methodology must be flexible with regard to the prioritization of requirements in order for it to apply if this prioritization changes. Since the design under study would serve as a vehicle for further study of compression methods concerning disparity data, flexibility had a higher priority than factors such as cost. However, by having an abstracted and organized way of describing the codec functionality, other implementations would become feasible having different design criteria.

Section 2 of this paper presents the basic performance requirements that were used for designing the codec architecture, while §3 describes in detail the codec architecture and its functional partitioning. Section 4 presents details on the implementation of various system functions, and §5 gives some experimental results.

2. System requirements

Disparity data are organized into data entities called disparity maps, which correspond to video frames captured by a camera, and are output from the disparity estimator at a rate of 25 mps (maps per second). Part of the ATM channel data rate is allocated for disparity information transmission. This rate is determined during the initialization of the whole system and remains constant during each session. The allocated channel data rate can be equal to or smaller than the uncompressed bit-rate of the disparity estimation unit and is divided into fixed-size time slots. In the absence of extensive studies on the effect of disparity map errors on image quality, a method called controlled data loss (CDL) was chosen to limit the output bit-rate to a threshold value during decreased compression. In CDL, the compression ratio is evaluated on a map-to-map basis. Each compressed disparity map is allocated a slot. If a compressed map size is larger than its uncompressed map, then this map is transmitted uncompressed. When an output slot cannot carry a complete disparity map (either compressed or uncompressed), the transmission of this map continues into the next slot; meanwhile the next map is not transmitted. At the decoder side, the previously received map replaces a rejected map. In slow-moving scenes, like

those that occur in teleconferencing applications, disparity data do not change significantly from map to map. The above scheme was chosen since its implementation is independent of the compression algorithm and allows progressive degradation of the whole quality of service.

The function of the encoder unit is to compress the disparity data stream losslessly and to transmit it over a constant bit-rate transmission link. In this application, the meaning of lossless compression is as follows. The information contained in each video frame must be compressed without any loss of information, while some complete frames may be lost. Thus, an appropriate module, which implements a lossless compression algorithm, is required in the disparity codec. The algorithm selection process involved the evaluation of the following parameters:

- achieved compression rate (coding efficiency);
- algorithm complexity;
- latency.

We evaluated the performance of three categories of general-purpose lossless compression algorithms: Huffman coding, arithmetic coding and Ziv–Lempel (LZ) coding (Papadimitos *et al.* 1998). The evaluation was mainly based on the first two parameters. Software implementations of algorithms from the above categories were ported to and optimized for an evaluation platform based on a high-performance floating point DSP. Simulations carried out on this evaluation board showed that the Lempel–Ziv algorithms combine good compression ratio and high throughput. A commercially available ASIC, implementing an LZ-78 type algorithm, was also included in the evaluation and was found to offer satisfactory compression and high throughput, and thus it was selected to be used in the final system configuration. As noted above, entropy-coding algorithms produce a variable bit-rate output, while the ATM multiplexer allocates a constant bit-rate channel (CBR) for the transmission of the compressed disparity data. The use of the LZ-78 algorithm satisfies the lossless compression requirements on each frame, but another technique has to be used for the rest of the system functionality. So we chose to use re-programmable logic for this additional functionality.

The proper operation of the intermediate viewpoint interpolator is based on the synchronization of the received video streams (left and right) with the disparity information data stream. The proper operation of the interpolator is achieved by bounding the delay introduced by the video and the disparity codecs and by guaranteeing the same mean delay to all three compressed data streams over the ATM network. The disparity encoder and decoder units introduce user-programmable delay, so that they can be used to match the delay introduced by the video compression units.

The disparity encoder and decoder units use unidirectional synchronous interfaces conforming to the CCIR 601/656 standards. The video data are encapsulated by appropriate control byte sequences, which enable the exact positioning of the data on the video display. According to CCIR (1990), the data streams are divided into entities called frames. Each frame is further subdivided into two fields. The frame consists of scan-lines, which contain the video data. The active video and the blanking data are delineated by appropriate byte sequences. The interface is clocked at 27 MHz. The duration of each frame is constant, so timing signals for synchronization purposes can be extracted by searching for the frame boundaries. This standard is used to transmit the image data and to convey the disparity data even though their

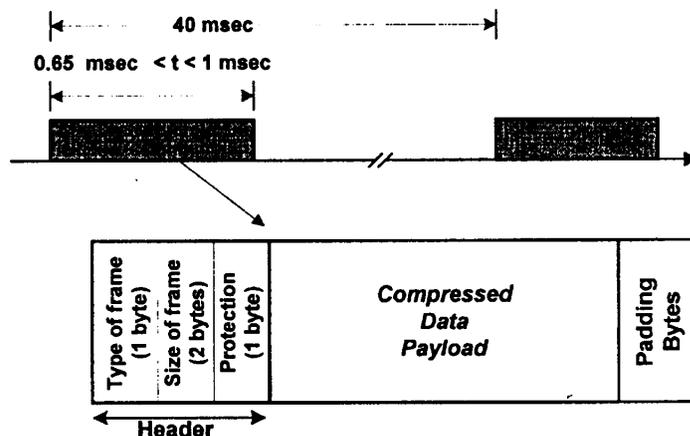


Figure 2. ATM interface format and timing.

data rate equals 5.184 Mbps. The disparity data derived from CCIR 601/656 images are present in the active video data positions of every fourth scan-line.

According to the system requirements, the compressed disparity data are transmitted over an ATM CBR channel since it preserves the timing relationship between the end points of the system (Onvural 1994). The requirement that the disparity and the video bit-streams are delayed equally during transmission is achieved by multiplexing the data into one CBR transport stream, and by introducing the same delay on all compression procedures running concurrently. The output interface of the encoder unit and the input interface of the decoder unit use a synchronous interface to transfer the compressed disparity bit-stream to and from the ATM multiplexing/demultiplexing equipment. This interface is clocked at 13.5 MHz, a frequency that is derived from the CCIR 601/656 interface. The output data are organized into equally spaced, fixed-size blocks called ATM blocks, each of which occupies an output slot. As shown in figure 2, each block contains a header with information on the size of its payload and its status. This header is protected using a CRC byte, implementing single bit error correction (Maniatopoulos *et al.* 1996). Padding bytes are used to keep the size of the block constant. The CBR channel allocated for the disparity data has a fixed capacity that is determined during the call set-up. In all cases the rate of the ATM blocks is set to 25 blocks per second, thus having 40 ms inter-block time, while the duration of each block depends on the allocated bit-rate.

3. System partitioning

After outlining the requirements of the disparity encoder and decoder units imposed by the system functionality, we attempted to discern the main functional blocks of these two units. These blocks are considered as logical 'containers' of similar functions, and the partitioning of their functionality gives us valuable clues on possible methods of implementation. From the system requirements we can discern the main modules that should constitute the disparity encoder unit. These are the following:

- The *disparity data reception module* (DDRM). This module filters the disparity data from the CCIR 601/656 standard input bit-stream and transfers the data to the compression module.
- The *compression module* (CM). This module undertakes the compression of the disparity data using the lossless algorithm.
- The *data framer* (DFr). This module implements the mapping of the variable bit-rate compressor output to the constant bit-rate transmission channel.

The disparity decoder unit should correspondingly have the following modules:

- The *data deframer* (DDFr). This module extracts the compressed data from the ATM data link and transfers them to the decompression module.
- The *decompression module* (DM). This module decompresses and reproduces the original disparity data.
- The *disparity data transmission module* (DDTM). This module accepts the uncompressed data and adds the appropriate format bytes so that the original CCIR 601/656 frame is generated.

Deciding whether these modules should be implemented in hardware (ASIC or FPGA) or whether a microprocessor-based solution should be used is not a trivial task. However, by understanding all aspects of the required functionality, we can carry out the partitioning with the lowest risk possible. In Papadimatos *et al.* (1999) we described the object model that we used as a guide for further analysis. The object model provides a static description of the system, of which the following parts are of interest to us: the disparity data module, the compression/decompression engine, the FIFO, the CDL data framer/deframer and the system control (microprocessor).

The final system was designed in order to take advantage of an off-the-shelf available compression solution with the flexibility of reprogrammability in both hardware and software. Thus, keeping in line with the various simulations on existing disparity maps, we decided to use a general-purpose ASIC compressor augmented by FPGAs and controlled by a microcontroller with on-chip timer and debugging facilities. Figure 3 shows the architecture of the complete codec. The compression/decompression engine attains the required processing rate while achieving compression rates comparable to those of other methods. It also undertakes both compression and decompression tasks through software reconfiguration. It implements a high-speed lossless dictionary-based algorithm with a 2.5 Mbyte/s maximum throughput, while various performance parameters can be changed by the microcontroller. This algorithm belongs to the Lempel–Ziv family of algorithms (LZ78). Compression rates on test sequences ranged from 45% to 75%, resulting in effective bit-rates in the range of 1.30 Mbps to 2.86 Mbps. By using an advanced 16-bit microprocessor the unit operates either as stand-alone with on-site programming of coding parameters or as controlled by a host computer.

The dynamic behaviour of the system can be described by using MSC charts. Figure 4 shows the interactions required to achieve a smooth flow of data through the disparity encoder. During normal system operation the microprocessor initializes the compression engine that begins processing the data. The compressor sends an acknowledgment when a frame of disparity data has been successfully compressed. Then the compression ratio is calculated and is transmitted along with other status information to the data framer. The data framer then outputs the frame according to the CDL scheme.

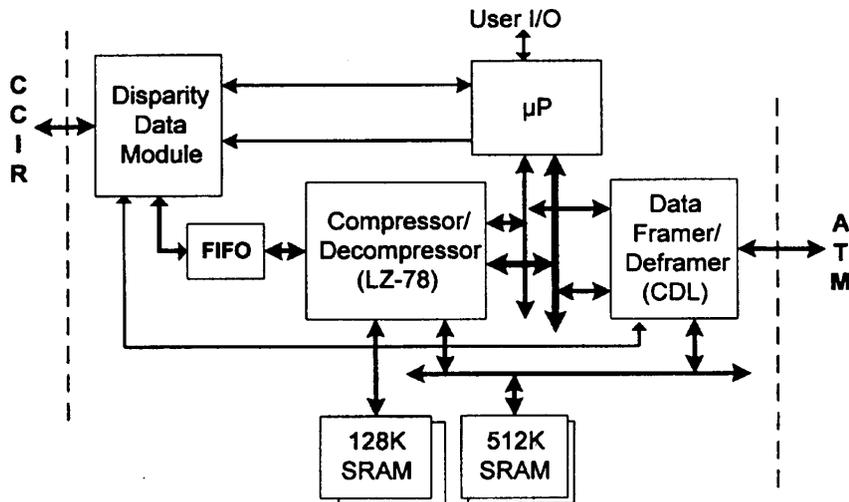


Figure 3. Disparity codec architecture.

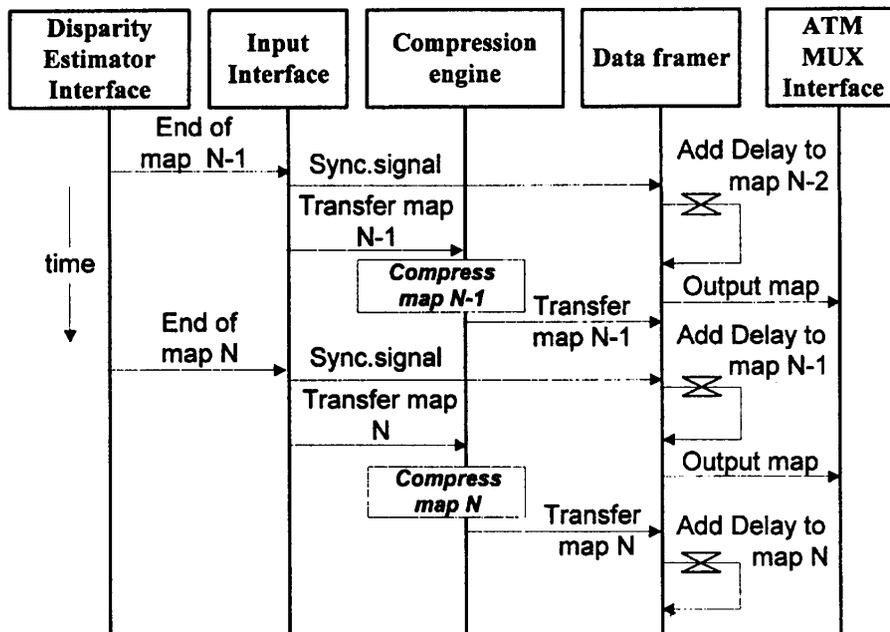


Figure 4. Control interactions of the disparity encoder during normal operation.

The disparity decoder control interactions are shown in figure 5. The data deframer synchronizes the CCIR transmitter interface at constant intervals. Owing to the CDL logic in the disparity encoder, if a compressed disparity map does not fit in one ATM block it is fragmented and transmitted in two ATM blocks. This entails re-assembly of this frame at the decoder unit. When the data deframer inputs a block from the ATM demultiplexer, it decodes the header and stores the payload in a

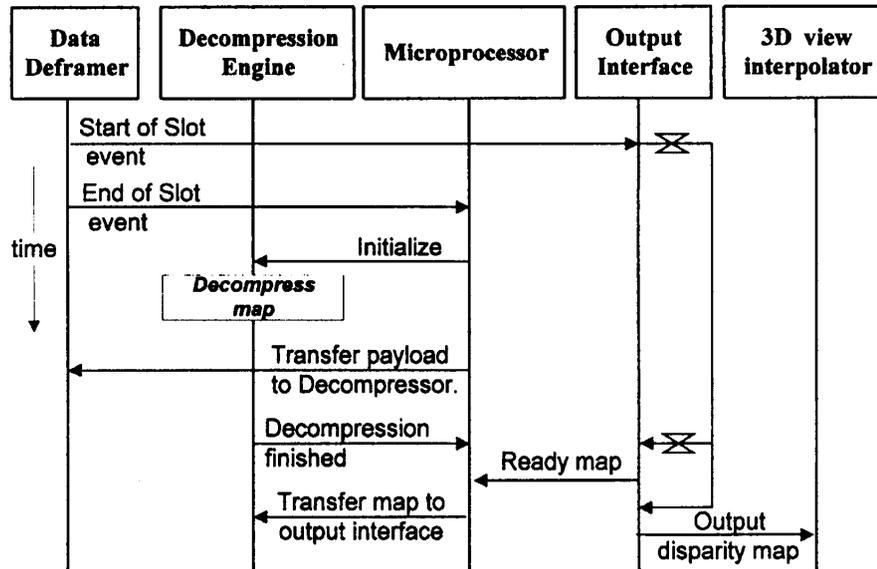


Figure 5. Control interactions of the disparity decoder during normal operation.

buffer. The header provides information on the status of the payload, i.e. whether it is compressed or uncompressed and whether it contains a whole disparity frame or part of it. This information is passed to the microprocessor, which configures the decompression engine accordingly. If the payload contains compressed data, then they are decompressed and stored in a FIFO. After a predefined delay, the output interface retrieves the unformatted disparity frames and outputs the decoded disparity maps.

The various state machines described in the previous sections have been implemented using a combination of FPGAs, FIFOs and SRAMs. By virtue of the design, one PCB was manufactured and is used as either an encoder unit or a decoder unit. The differentiation between encoder and decoder unit was achieved through different configuration files for the FPGAs and the microprocessor, and through the reversal of the data flow through the unidirectional devices such as the FIFO and the interfacing ICs.

4. System implementation

This section describes in detail the implementation of various functions at the encoder and the decoder units. The block diagram of the encoder unit is shown in figure 6.

4.1. The encoder unit

The disparity data reception module (DDRM) is the module that implements the CCIR 601/656 receiver functions. The receiver filters the disparity map information from the incoming bit-stream. The CCIR interface uses sequences of reserved bytes to delineate the payload data in each scan-line, so the DDRM initially searches for

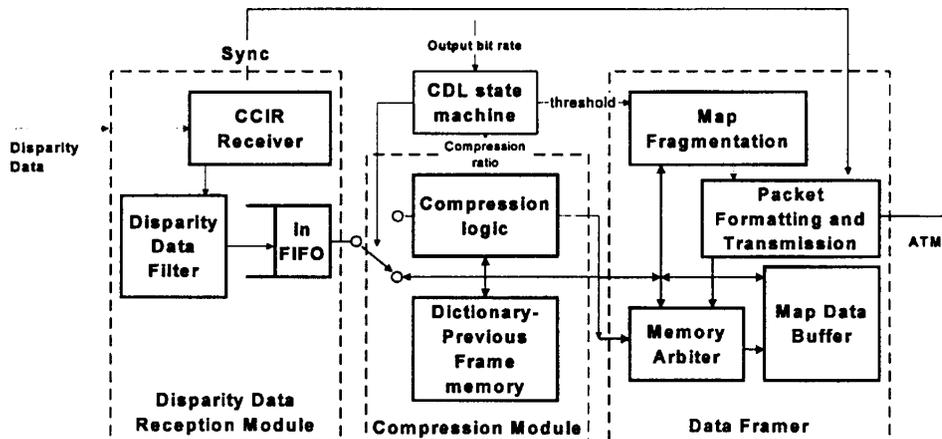


Figure 6. Block diagram of the encoder unit.

these sequences and then passes the data generated between these sequences to the next processing stage. It also produces a synchronization signal every time it detects a frame boundary in the CCIR bit-stream. The control sequences, which delineate the video scan-lines, are used to generate the synchronization signal. The maximum synchronization resolution achieved in the system corresponds to one scan-line, i.e. 64 μ s. This synchronization signal is routed directly to the data framer so that it can synchronize the output interface with low skew.

Every fourth scan-line of the active video contains disparity data. Each byte of valid disparity data takes one of two different values: ML and MR, so it is coded using a single bit. Therefore the byte-packetizer at the output of the disparity data filter packs the command information of eight bytes into one and passes it to a FIFO.

The compression module uses a dictionary-based LZ algorithm. At the beginning of the compression process the dictionary of disparity values is created. It is updated continuously with the processing of each byte containing eight disparity values. Compression is achieved when sequences of disparity values are replaced by references to a dictionary entry. The dictionary is reset at the start of each map so that resynchronization is possible when maps are lost. The CDL logic decides which frames will be compressed and which will be forwarded intact to the following module, on the basis of information on the compression ratio achieved. When the size of a compressed map exceeds a certain threshold (the size of the uncompressed frame), an uncompressed copy is forwarded to the data framer.

The data framer encompasses the functionality for transferring the compressor output frames to the ATM multiplexer via a CBR channel by implementing the controlled data loss (CDL) scheme. The task of the data framer is to output each frame after a known delay from the reception of the EOF (end of frame) signal. In this context we consider the delay introduced to a disparity frame to be the time between the decoding of the EOF signal of the respective CCIR frame and the output of the first byte of the ATM block that contains the corresponding disparity map.

Figure 7 shows the interactions that take place in the data framer. A synchronization signal from the input interface controls the output of the ATM multiplexer blocks. When both interfaces are synchronized, the data framer decides which

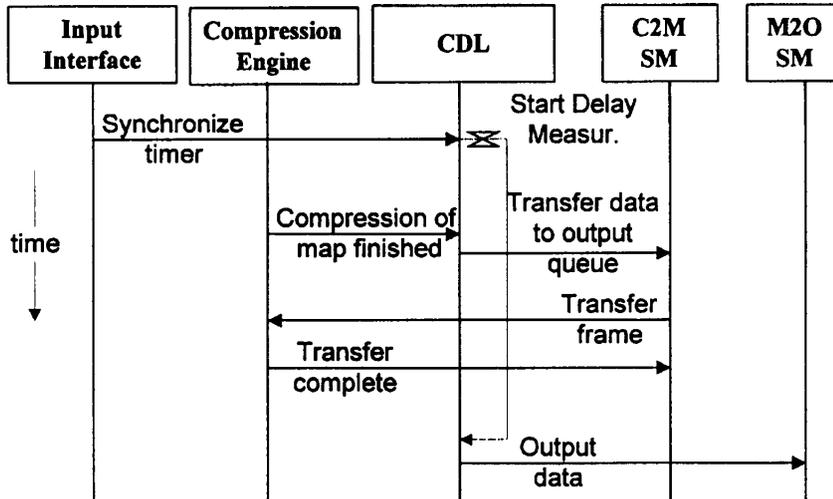


Figure 7. Data framer internal interactions.

compressed block should be inserted in the output stream. The capacity of each output slot is smaller than the size of an uncompressed disparity frame and sometimes is also smaller than the size of a compressed disparity frame. In this case, the CDL logic is used to reduce the output bit-rate. The data framer is an aggregate of three objects:

- the compressor-to-memory state machine (C2M);
- the memory-to-output state machine (M2O);
- the CDL logic (CDL).

The C2M state machine carries out the actual transfer of the compressed data from the compression engine to the delay buffer memory. It implements the actual handshaking between the compression engine and the buffer memory controller and allocates the necessary memory area. The M2O state machine creates the output block with the corresponding control information and the compressed data. This block is then transmitted when the required amount of delay has been introduced. The CDL logic decides when to transmit or reject a block on the basis of the corresponding disparity frame's compressed size.

The internal operation of the CDL logic is described efficiently using SDL (specification and description language) as shown in figure 8. We consider the CDL logic as a state machine composed of the compressed block (CB) and the uncompressed block (UB) states.

At start-up the state machine is in the CB state. When the EOF signal is decoded at the input interface, it starts a timer in CDL. Meanwhile, the compression engine compresses the corresponding disparity map and signals the data framer upon completion of the compression. The CDL then initiates the C2M state machine to transfer the compressed map to the buffer. It also stores the compression ratio of each stored map. As shown in the SDL diagram, these interactions occur in either state and the CDL finally returns to the same state.

Upon expiration of the timer for the specific frame, the CDL reads the compression ratio and computes the slot size required to transmit the compressed map. If

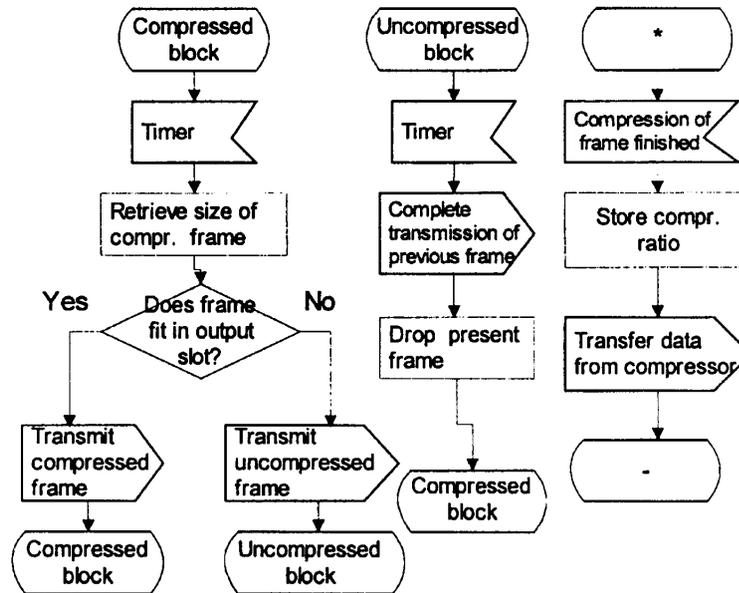


Figure 8. SDL diagram of the CDL state machine at the encoder unit.

the capacity of an ATM slot is sufficient to carry the compressed map, the CDL signals the M2O state machine to output this map. An appropriate header is inserted at the start of the slot. Upon completion of the transmission the CDL returns to the CB state. On the other hand, if two ATM blocks are required to transmit the map, the CDL starts the transmission at the current ATM slot, seizes its operation when the end of the slot arrives, and continues at the beginning of the next slot. The next disparity map is discarded and the procedure restarts at the next ATM slot.

The data framer (DFr) has been implemented by partitioning its functionality into reprogrammable logic and microcode. This module handles the data transfers from the compression engine to the SRAM and also controls the logic required for the output interface. DFr performs the formation of the output frame with header, payload and padding bytes; it implements the M2O state machine and part of the C2M state machine, concerning the transfer of data to the buffer memory. The storage area in the SRAM is organized as a circular buffer with read and write pointers. The C2M state machine controls the write pointer while the M2O state machine controls the read pointer. The SRAM is accessed by the compression engine and the data framer, so a memory arbiter is required to decide which unit should have access. In order to minimize deviation from the programmed delay value, the M2O state machine has the highest priority.

The CDL logic sets the read pointer to the memory area that holds the payload data and informs the M2O state machine of the kind of payload the frame will carry in order to create the correct header information. The header is then output to the multiplexer interface. M2O gains control of the SRAM bus and starts reading the payload data and passing them to the output. After reaching the last payload byte, it pads the output frame with bytes so that it has a constant size. Finally the SRAM bus is relinquished and the compressor completes any interrupted data transfer.

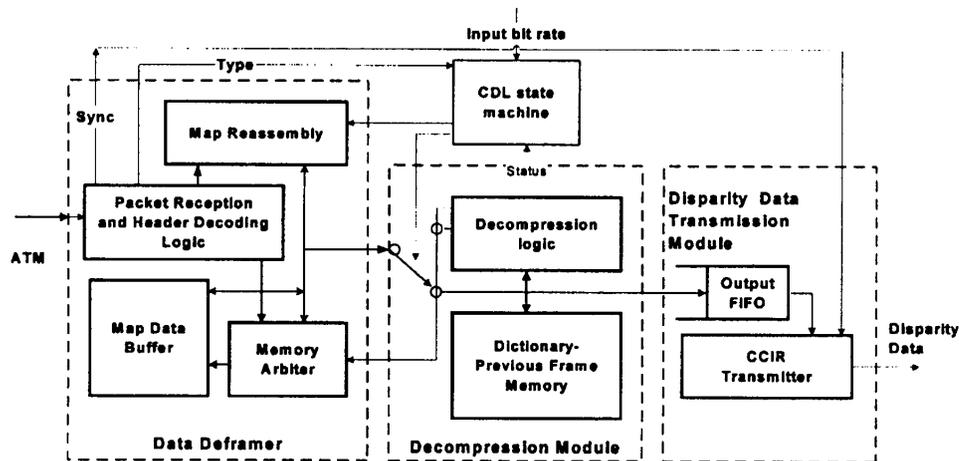


Figure 9. Block diagram of the decoder unit.

DDRM and DFr were implemented using XC4006 FPGAs. The DDRM required approximately 50% of the FPGA resources, while the DFr needed 85%.

4.2. The decoder unit

The block diagram of the decoder unit is shown in figure 9. The data deframer logic controls the ATM demultiplexer interface, extracts the data from the ATM blocks and stores the data in a FIFO. It also undertakes the reassembly of fragmented frames. Its architecture is similar to the DFr of the encoder unit. The synchronization signal is now generated by the header decoding logic and is transferred to the CCIR transmitter. The CCIR transmitter encapsulates the uncompressed disparity data in CCIR frames. The depacketizer module maps each bit of the raw disparity data to the appropriate CCIR byte. The CCIR frame generator inserts the control bytes to form the original 1720 byte scan-lines, as the CCIR 601/656 standard specifies.

The data deframer extracts the compressed disparity maps as they arrive from the ATM interface. The SDL diagram shown in figure 10 describes this functionality. When the data deframer inputs data from the ATM interface, it decodes the header and stores the payload in a buffer. The header provides information on the status of the payload, i.e. whether it is compressed or uncompressed and whether it contains a whole compressed disparity map or part of a frame. This information is passed on to the microcontroller, which configures the decompression engine accordingly. The decompression module produces the original disparity maps from the compressed data input. The CDL logic controls whether the input data will be decompressed or be forwarded to the disparity data transmission module. LZ schemes do not require explicit transmission of the dictionary since it is automatically recreated from the incoming data. Data errors are confined to a single map, owing to the resetting of the dictionary at the beginning of each transmitted map.

The recovered unformatted disparity data are used by the disparity data transmission module to reconstruct the originally transmitted CCIR frame.

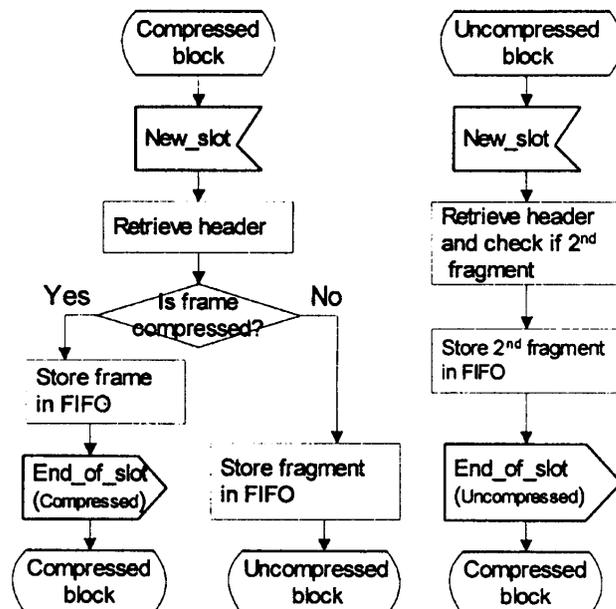


Figure 10. SDL diagram of the CDL state machine at the decoder unit.

Uncompressed data, and compressed data requiring more than a slot, are fragmented and transmitted in two slots. This entails reassembly at the decompression unit. The reassembly consists of gathering the uncompressed data from two slots and storing them in the FIFO. To maintain a constant flow of disparity maps to the next stage, the decompression unit introduces variable delay. An automatic delay control module introduces the user-defined constant delay. The decoder modules were also implemented using XILINX XC4006 FPGAs. The data deframer needs around 80% of the available resources, while the CCIR transmitter uses around 75%.

5. Experimental evaluation

During this development, we have built and tested two fully operational units, an encoder unit and a decoder unit. Such a unit is shown in figure 11. The compression engine achieved a compression rate of between 45% and 50%, so a 3 Mbps channel was sufficient for the lossless transmission of the disparity data. The introduced delay ranged from 170 m to 1 s for both units. This high delay is required for delay matching with commercially available MPEG2 codecs. In real-life sequences, the CDL mechanism can limit the bit-rate required according to the desired quality. Statistics from six typical videoconferencing sequences, each lasting 20 min, were collected for evaluation purposes and the results are shown in table 1. The CDL logic causes the loss of disparity maps and the interpolator has to operate on disparity information that was extracted from the previous frame. Degradation is introduced since the depth information used in some disparity maps differs from the depth information contained in the original images. The viewer notices this as an unnatural scene.

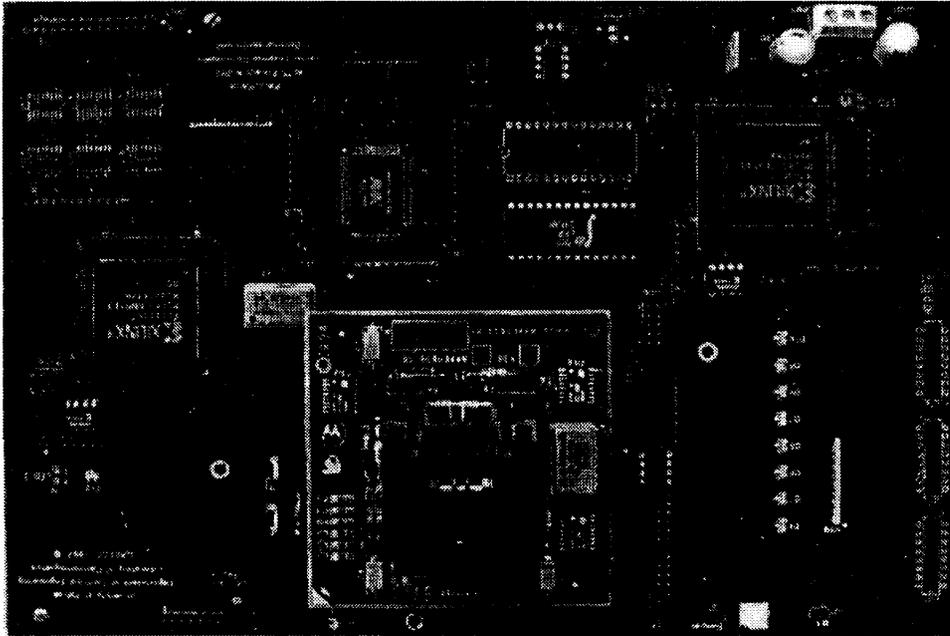


Figure 11. The codec board.

Disparity maps containing great changes in disparity information are much harder to compress, as is evident for sequence 5 when the CDL is configured for 90% pass rate. The CDL is adaptive in the sense that only the most recent disparity information is used to replace dropped maps. This ensures that the decoder uses the most recently available disparity information, which does not exhibit discontinuities in space during the duration of a 40 ms frame. During subjective tests, the use of the disparity maps of image frames extracted 80 ms earlier was virtually unnoticeable when movement was not extreme.

For initial evaluation we used various image sequences such as the 'MAN' sequence available from the AC092 PANORAMA project. In figure 12 the first frame of this sequence, its corresponding disparity vector map and derived disparity command map are shown. The command format consists of sending two different command sequences for controlling the frames sent to each eye. By adding the number of each command sequence in each frame and comparing this value at

Sequence	Mean compression ratio (%)	Required bit-rate for 10% map drop rate from CDL (Mbps)	Required bit-rate for 30% map drop rate from CDL (Mbps)
1	81	1.35	1.19
2	81	2.38	1.40
3	85	1.09	0.88
4	77	1.56	1.40
5	73	2.90	1.92
6	77	2.07	1.97

Table 1. Achieved mean compression rates and bit-rate requirement for a predefined quality setting.

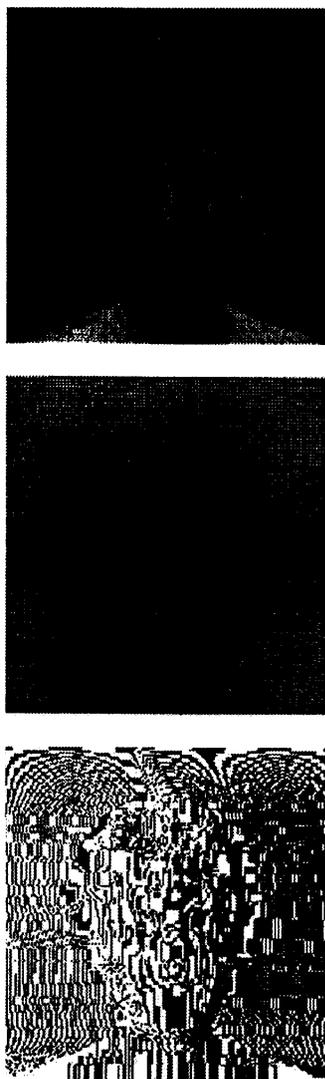


Figure 12. First frame of the MAN sequence and its disparity and command maps.

each command position with values from nearby disparity maps, we can estimate how different each disparity map is.

The system chain was successfully demonstrated in October 1998 in Berlin under real-life conditions. The experimental set-up included all the required hardware blocks for disparity extraction, transmission and image recreation. The disparity estimator hardware processed images captured by two calibrated cameras. The generated disparity maps were processed by the disparity codec hardware. Two commercial MPEG2 codecs compressed the images in real time introducing approximately 200ms delay, which was noticeable but not annoying. The disparity codec was configured to introduce the same amount of delay so that all bit-streams were synchronized on image and map boundaries on the codec interfaces, i.e. each

image and its corresponding disparity map were synchronized on camera-MPEG2 codecs and estimator-disparity codec interfaces. This eased the synchronization of the bit-streams with the same timestamps being inserted on the image frames and the corresponding disparity maps by the ATM multiplexer. All bit-streams were transmitted over the ATM network to another location in Berlin. On the receiver side the MPEG2 codecs and the disparity codec reproduced the original bit-streams and forwarded them to the interpolator, which produced the required bit-streams for the auto-stereoscopic display. The stereoscopic display sent the correct image for each eye with the help of prisms embedded in front of the screen. The stereoscopic impression was changed dynamically as the subject in front of the screen moved his head. We tested with various allocated bit-rate allocations for the disparity, reducing the bit-rate down to 1 Mbps. Disparity maps derived from one image frame could be used to regenerate the depth impression for other image frames that did not have more than approximately 150 ms time difference from the originating image frame. In slow-moving scenes the depth errors were barely noticeable at the displayed object edges. Depth errors became gradually quite annoying, especially when delay differences between disparity and image frame were larger than 200 ms.

6. Conclusions

In this paper we have presented the design and architecture of a lossless compression unit and the corresponding decompression unit capable of compressing disparity data for videoconferencing applications. The design process was divided into three phases: system requirements, system analysis and hardware/microcode development. This approach allows us to gain a thorough understanding of the task at hand, and to integrate commercially available ASICs and custom-designed logic using a flexible hardware architecture. The compression and decompression units satisfy the application requirements, offering real-time compression and decompression of disparity data while introducing constant delay.

References

- CONSULTATIVE COMMITTEE FOR INTERNATIONAL RADIO (CCIR), 1990, Recommendation 656: Interface for digital component video signals in 525-line and 625-line television systems.
- MANIATOPOULOS, A., ANTONAKOPOULOS, T., and MAKIOS, V., 1996, Implementation issues of the ATM cell delineation mechanism. *Electronics Letters*, **32**, 963–964.
- OHM, J., GRONEBERG, K., HENDRIKS, E., IZQUIERDO, E., KALIVAS, D., KARL, M., PAPADIMATOS, D., and REDERT, A., 1997, A real-time hardware system for stereoscopic videoconferencing with viewpoint adaptation. *International Workshop on Synthetic-Natural Hybrid Coding and Three-Dimensional Imaging*, Rhodes, Greece, September.
- OHM, J., GRONEBERG, K., HENDRIKS, E., IZQUIERDO, E., KALIVAS, D., KARL, M., PAPADIMATOS, D., and REDERT, A., 1997, A real-time hardware system for stereoscopic videoconferencing with viewpoint adaptation. *Signal Processing: Image Communication*, **14**, 147–171.
- ONVURAL, O., 1994, *Asynchronous Transfer Mode Networks* (Norwood, MA: Artech House), pp. 28–31.
- PAPADIMATOS, D., ANTONAKOPOULOS, T., and MAKIOS, V., 1998, Real-time disparity information compression in 3D teleconferencing systems. *24th EUROMICRO Conference*, Vasteras, Sweden, August.

- PAPADIMATOS, D., ANTONAKOPOULOS, T., and MAKIOS, V., 1999, Performance analysis of a selective rejection algorithm on compressed disparity data. *7th International Conference on Advances in Communications and Control*, Athens, Greece, July.
- RUMBAUGH, J., BLAHA, M., PREMIERLANI, W., EDDY, F., and LORENSON, W., 1991, *Object-Oriented Modelling and Design* (New York: Prentice Hall).
- TZOVARAS, D., GRAMMALIDIS, N., and STRINTZIS, M. G., 1996, Disparity field and depth map coding for multiview image sequence compression. *International Conference on Image Processing (ICIP)*, Lausanne, Switzerland, 6–19 September.
- ZIEGLER, M., 1997, *Region-based Analysis and Coding of Stereoscopic Video* (München: Akademischer Verlag), pp. 102–103.