Real-Time Video Compression for Driver Assistance Camera Systems

Mehrnoush Rahmani, Holger Kloess, Wolfgang Hintermaier
BMW Group Research and Technology
80788 Munich, Germany
{firstname.lastname}@bmw.de

Eckehard Steinbach
Technische Universitaet Muenchen
Media Technology Group
80290 Munich, Germany
eckehard.steinbach@tum.de

ABSTRACT

In-vehicle communication systems have become quite complex and costly mainly because of the growing number of in-car audio and video applications and the different interconnected network technologies. In order to reduce the complexity and the production cost, an IP-based network realized by the Fast-Ethernet and Wireless LAN (WLAN) technologies has been proposed in our previous work that connects all audio and video transmitting devices in the car. However, in order not to exceed the link capacities while transmitting several video streams simultaneously, video compression is necessary. Video streams from driver assistance services have strict delay and quality requirements for the underlying network system. In this work, we compare and analyze applicable video codecs for real-time video applications in the car and define metrics to evaluate the performance of the algorithms in wired and wireless networks. Concepts for the hardware realization of IP cameras with hardware based video codecs are finally described to further improve the coding performance.

Keywords

Driver assistance, Camera systems, Video compression, In-vehicle video transmission

1. INTRODUCTION

1.1 Motivation

Today's premium cars incorporate a multitude of interconnected multimedia and infotainment devices such as the telephone system that is linked to the radio and CD/MP3 player, digital television, DVD player, and the navigation system that updates route recommendations with real-time congestion warnings from the radio device. In order to assist the driver with various tasks, car manufacturers have equipped cars with real-time camera systems that serve for both convenience and safety purposes. Cameras are used for two major applications: viewing and processing. The purpose of viewing applications is to provide the driver with a view of the areas in the surroundings of the car that are outside of his field of vision. In processing applications, camera systems are employed together with image processing algorithms that evaluate the outer-car environment with the help of object recognition and other image analysis tools. In this work, we mainly focus on the viewing applications.

The current in-vehicle video transmission system consists of not only a large number of electronic control units (ECUs) such as different cameras, but also of several network technologies, e.g., CAN [1], MOST [2], LIN [3] and also point-to-point analogue FBAS and LVDS (low-voltage differential signaling) wirings. The latter two connections are mainly applied for camera systems to transmit raw video streams. The video data from cameras is transmitted to the receivers as analogue data streams via FBAS cables. At the receiver side, it is converted into a digital signal and transmitted via LVDS wires to the displays which means additional cost and complexity at the receiver. A transmission to more than one receiver becomes even more complex due to the additional wiring effort. Accordingly, the in-vehicle communication system has become inflexible, complex and costly.

In order to reduce the costs and complexity, a new network architecture based on the IP, Fast-Ethernet and WLAN technologies has been proposed in [4] to accommodate all audio and video applications in the car while guaranteeing the required quality of service (QoS). In order to simultaneously transmit several high-resolution video streams from driver assistance cameras over the proposed network with a limited link capacity, video compression is required to reduce the data rates. The application of codecs for compression and decompression of media content adds delay to the transmission and introduces distortion to the media content as a consequence of quantization. However, if the codecs are implemented and configured accordingly, the additional delay and the introduced distortion can be kept small enough and within the acceptance bounds of driver assistance services.

1.2 Objectives

This work is concerned with the real-time compression of video streams from driver assistance cameras for viewing applications at low driving speeds, e.g., rear- and side-view cameras for the parking use case. In a real-time system, the correctness of an operation does not only depend on the logical correctness, but it also has to be performed within a given time period (deadline). In this work, the real-time criterion for a camera application implies that each of the processing steps, particularly encoding and decoding, has to be completed before the next frame is available, i.e., within one frame interval. The objective is to define constraints and requirements for an efficient and low delay video compression in the future in-vehicle convergent communication network. Applicable software realized video compression algorithms are evaluated for their

ISVCS 2008 July 22 - 24, 2008, Dublin, Ireland

performance by using several specific metrics. The most appropriate compression schemes with the best settings are determined and implemented in a prototypical testbed. Evaluation results are presented for both, wired and wireless networks. In order to improve the coding performance even further, we present hardware realization concepts for IP cameras that include hardware based video codecs.
Similar investigations have been carried out in the literature such as in [22]. However, previous studies have not considered the in-vehicle communication requirements. In the present work, delay, quality and complexity of applicable software codecs are analyzed for in-vehicle real-time applications.

2. SYSTEM DESCRIPTION AND METRICS FOR CODEC ANALYSIS

2.1 Transmission Scenario and Testbed Description

In the considered in-car transmission system, each frame is preprocessed after being captured at the source and compressed. Then, the bit stream is packetized and transmitted. Upon reception, the stream is depacketized, decoded, and eventually post-processed before being displayed at the sink. The underlying core network is a wired network realized by the Fast Ethernet technology according to [4]. A peripheral wireless network realized by the WLAN IEEE 802.11g has been applied to extend the wired core network and increase the application flexibility for the car occupants. Rear seat monitors or external sinks can, for example, be connected via WLAN to the Ethernet network as shown in Fig.1. We consider the Ethernet network as error-free due to its very low bit error rates [5] while the WLAN network is an error-prone network due to reflections, signal interferences, etc.
Several codecs have been implemented in software in a prototypical testbed. They include M-JPEG, MPEG-2 and MPEG-4 (part 2) through the libavcodec library [9] and the Xvid library [18] and the H.264/AVC (MPEG-4 part 10) codec realized by the x264 library [10]. For decoding, the libavcodec decoder for H.264/AVC has been used. The basic components of the prototype are as follows.

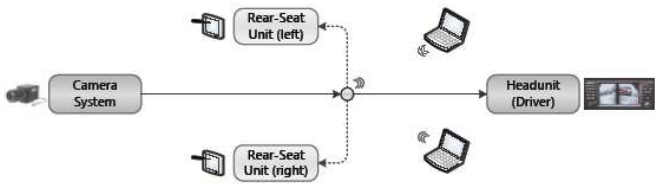


Figure 1: A scenario for the transmission of camera data for viewing purposes

- **Test Sequences:**
  1. Highway sequence [14]: The test sequence Highway is subsampled to CIF resolution and 4:2:0 format. It shows a drive on a highway at a moderate speed. Among the 2000 frames in the original video, frames 550 to 1349 were used, since the content of the frames before and after the selected sequence is very similar. Therefore, the sequence length is 800 frames.
  2. VGA sequence: The second video sequence was produced from typical scenes captured by automotive cameras. Different scenes were recorded with the camera that was used in the prototype design and combined into a sequence of 715 frames of VGA resolution. The test sequence contains a scene in which another car drives by close to the camera, a scene in which another car is approached while reversing (typical parking situation), a scene in which a person walks in front of the camera, a reversing scene at low speed in a darkened environment to reflect dimness, as well as reversing toward a wall with high spatial details in which the characteristic distortions of a wide-angle lens become visible.
- **FireWire Camera:** For frame acquisition, the camera DFK 21AF04 from The Imaging Source [11] was used. It captures YUV 4:2:2 frames and is connected to a PC (test computer 1) via a FireWire (IEEE 1394) cable. In the implementation, it is controlled via the unicap library [12].
- **Server and Client:** The server and the client are implemented upon two computers connected via Fast Ethernet using a layer-2 switch. Test computer 1 (server) was a Dell Precision notebook with an Intel Dual Core CPU running at 2 GHz. Test computer 2 (client) was a desktop PC with an Intel 3.4 GHz CPU. On both test systems, the Linux distribution open SUSE 10.1 with kernel 2.6.16 working with 1000 Hz Kernel cycle frequency was used.
- **Data Transmission and Session Control:** For the transmission, the UDP protocol without any additional higher layer protocols was employed. Session control was implemented through the SNMP framework.

2.2 Applied Metrics for Codec Comparison

- **End-to-End Delay:** Low delay has been identified as one of the major performance criteria for the transmission of real-time video streams from automotive camera systems. In this context, the end-to-end delay, i.e., the time from image acquisition to the reception and presentation at the client, is the main measure. It is an accumulation of the individual delays that occur in a video acquisition and transmission system. For viewing applications such as for the rear-view camera a maximum end-to-end delay of 45 ms is acceptable. The end-to-end delay time is reduced to 33 ms (for 30 frames/s) when image processing is applied to reserve sufficient time for the image processing algorithm.
- **Video Quality:** The video quality at the sink is another important metric for the comparison of video compression systems. Besides being a function of the data rate and the video sequence properties, the perceived video quality also depends on environmental conditions, such as lighting, the sampling density in the form of display size vs. resolution, the viewing distance, and others. Yet, for further analysis of this work, the environmental conditions are considered to be static for all compression systems. Two well-known objective metrics, the Peak Signal-to-Noise Ratio (PSNR)[1] and the Structral Similarity Index (SSIM) [6], [7] have been applied for the

[1] $PSNR = 10 \log_{10} \frac{L^2}{MSE}$, where MSE defines the mean squared error and $L = (2^n - 1)$ is the dynamic range of the pixel values with n being the number of bits used to represent the value of a pixel per component, is a mathematical function that evaluates the effects of distortion introduced during compression and transmission.

analysis of the compression algorithms. While PSNR cannot be directly translated into perceived image quality when comparing different kinds of distortion, as it is a mere mathematical function [8], the perceptual approach SSIM is able to predict the perceived quality of an image or a video automatically based on properties of the human visual system. SSIM extracts structures from an image or video. Thus, it interprets a scene based on its structures and the changes in the structures. A higher SSIM value corresponds to higher similarities between the two compared video sequences with a maximum value of 1 representing two identical sequences.

- **Complexity:** Video standards only define the bit stream syntax and the decoding process. Thus, developers have degrees of freedom in their choices when implementing an encoder. The algorithmic complexity of an encoder is then affected by the specific implementation architecture, its data and memory structures, and optimizations. Further, when looking at the number of features and options that can be combined for a standard like H.264/AVC, it becomes clear that the complexity of both encoder and decoder is affected by the feature choices. In any case, the algorithmic or computational complexity is often based on the processing time for a given sequence and on a particular platform. Despite the implementation uncertainties, the processing time still provides useful and valuable indications and is defined as the metric of the computational complexity here.

3. COMPARISON RESULTS OF VIDEO COMPRESSION ALGORITHMS

3.1 Computational Complexity

In the following, the compression efficiency[2] as well as the processing times of Motion JPEG (libavcodec), MPEG-2 (libavcodec), MPEG-4 (libavcodec), MPEG-4 (Xvid), and H.264/AVC (x264) are presented. For the sake of simplicity, the codecs are denoted by M-JPEG, MPEG-2, MPEG-4 (L), MPEG-4 (X), and H.264/AVC, respectively. They were determined such that the resulting video quality was the same for all codecs. The SSIM value for the Highway sequence is 0.95 (condition a). For the VGA sequence, two conditions were defined: an SSIM value of also 0.95 (b) and a PSNR value of 40 dB (c). Table 1 summarizes the results for the three conditions. The table specifically includes the measurements

		PSNR	SSIM	Conv. Time (msec)	Enc. Time (msec)	Transm. Time (msec)	Dec. Time (msec)	Display Time (msec)	Video Bit Rate (kbit/s)	IP Bit Rate (kbit/s)	Compr. Ratio	Bits per Pixel
(a) Highway Sequence	M-JPEG	39.4	0.950	-	1.61	0.86	0.82	0.22	2341	2398	0.064	0.25
	MPEG-2	39.7	0.950	-	2.40	0.44	0.65	0.23	1378	1416	0.038	0.15
	MPEG-4 (L)	39.6	0.950	-	2.98	0.33	0.92	0.24	1124	1157	0.031	0.12
	MPEG-4 (X)	39.6	0.950	-	3.79	0.31	0.93	0.17	1048	1079	0.029	0.11
	H.264/AVC	40.0	0.950	-	14.05	0.27	1.95	0.24	953	984	0.026	0.10
(b) VGA Sequence	M-JPEG	37.0	0.950	0.38	5.40	3.20	2.46	1.01	8416	8589	0.076	0.91
	MPEG-2	37.5	0.950	0.44	7.68	1.57	1.89	0.62	4323	4417	0.039	0.47
	MPEG-4 (L)	37.3	0.950	0.42	10.12	1.28	2.44	0.58	3563	3642	0.032	0.39
	MPEG-4 (X)	37.3	0.950	0.49	13.62	1.25	2.37	0.60	3475	3554	0.031	0.38
	H.264/AVC	37.2	0.950	0.50	39.90	0.91	5.71	0.59	2636	2698	0.024	0.29
(c) VGA Sequence	M-JPEG	40.0	0.969	0.40	6.84	4.92	3.03	0.61	13384	13651	0.121	1.45
	MPEG-2	40.0	0.965	0.45	8.98	2.83	2.46	0.59	7885	8047	0.071	0.86
	MPEG-4 (L)	40.0	0.967	0.44	11.46	2.62	3.14	0.60	7312	7463	0.066	0.79
	MPEG-4 (X)	40.0	0.966	0.51	15.52	2.60	3.09	0.60	7208	7357	0.065	0.78
	H.264/AVC	40.0	0.966	0.50	45.91	2.02	7.47	0.58	5680	5799	0.051	0.62

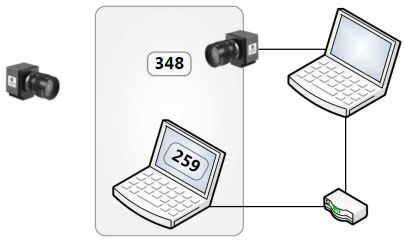
Table 1: Comparison of the processing times and the compression rates for M-JPEG, MPEG-2, MPEG-4, and H.264/AVC

[2]The compression ratio or efficiency can be computed by the size of the encoded sequence over the uncompressed size.

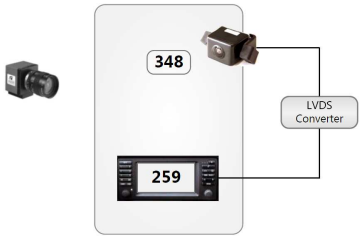
of the video quality, processing times, and the compression efficiency. The quality is given in terms of the PSNR and the SSIM. As expected, H.264/AVC requires the highest complexity in terms of encoding and decoding times in all measurements.

3.2 End-to-end Delay

The delay of a camera transmission system using compression ("digital system") was measured with a testbed as shown in Fig. 2(a) and compared with the delay of the current analogue rear-view camera system ("reference system" in Fig. 2(b)). In each system,



(a) Digital system (using compression)



(b) Reference system

Figure 2: Testbed for the measurement of the end-to-end delay

a counter showing the current time in milliseconds was captured by the respective camera. In the digital system, the data was compressed with Motion JPEG (M-JPEG), MPEG-2, MPEG-4 (X), and H.264/AVC. The compression parameters for the codecs were set such that a real-time compression was possible (even though, for x264 this still resulted in some skipped frames). The resulting video quality was not assessed. The data was then transmitted with UDP to the client computer over a 100 Mbit/sec Ethernet link using one switch. There, it was decompressed as soon as a frame was completely received (no additional buffering) and displayed on the screen. The counter in the digital system was displayed on that same screen. In the reference system, the video signal was converted into a digital signal before being displayed on the screen of the headunit. For both systems, the current time on the counter and the time on the display were captured with a second camera and stored as individual images. The difference in the counter readings represents the total delay of the system. Even though not all of the readings were clear (overlapping of the counter values due to the display response times and the shutter opening time of the camera), for each system and codec around 50 "clear" values were evaluated which showed a standard deviation of about 8 ms. Therefore, the outliers (set to 25% of the values for each test) were discarded. The results of the measurements with the most influencing values are shown in Fig.3. The averaged total delays are: 37.2 ms for the reference system, 84.4 ms for M-JPEG, 83.8 ms for MPEG-2, 92.5

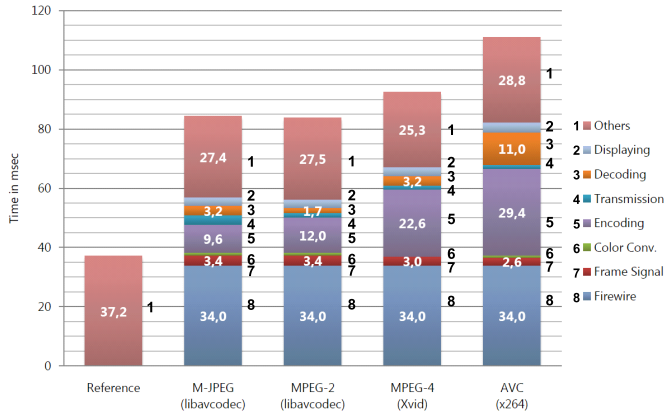


Figure 3: End-to-end delay values and the delay components for all considered codecs except of Dirac and Theora compared to the reference analogue system

ms for MPEG-4, and 110.7 ms for H.264/AVC. A large amount of the delay in the compressed transmission accounts from the use of a FireWire camera. There, the transmission of the data within one frame is spread over the whole frame period (isochronous transfer). With an assumed short processing time in the camera and the frame time of 33.3 ms, the image transfer from the camera to the computer accounts for about 34 ms of the total delay.

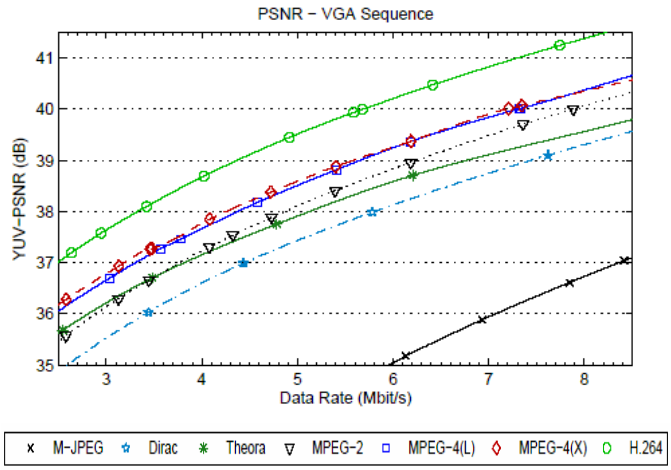
Delay components that could be measured are: the time until the Linux kernel signaled the availability of a new frame and the processing started (frame signal), the time for the YUV subsampling, the encoding, transmission, and decoding times, as well as the time that elapsed while copying the frame to the graphics buffer and issuing the command to update the display. The resulting delay contributions (assigned with ”Others” in Fig.3) that could not be explicitly measured mainly comprise the time from displaying the reference counter until it is actually captured by the camera and the time from issuing the display update command for the delay counter until the frame is actually displayed (depending on the display refresh rate, amongst others).

Consequently, in a fully IP-based in-vehicle network, the major delay component ”FireWire” will be eliminated when integrating the camera sensor and codec into one chip as explained in Section 4. By further adaptations to the in-vehicle requirements, the delay component ”Others” can also be significantly reduced. Thus, the end-to-end delay requirement of 45 ms for camera frames can be fulfilled for almost all mentioned software codecs.

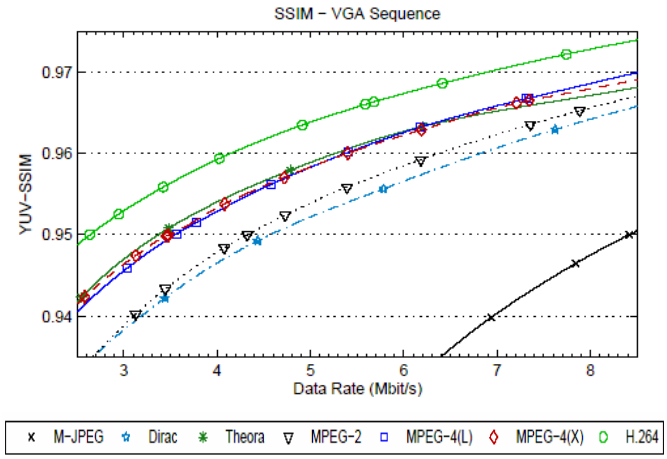
3.3 Video Quality

3.3.1 Error-Free Wired Networks

In the following, the video quality of the compression systems is assessed for different bit rates using the test sequences from Subsection 2.1. For the analysis, i.e., a comparison in terms of rate-distortion performance, two open source codecs Theora [15] and Dirac [16] are also applied besides the codecs mentioned before. The PSNR and SSIM values of the VGA sequence are shown in Fig. 4(a) and Fig. 4(b) for bit rates of 2.5 to 8.5 Mbit/s. The quality measurements for the Highway sequence correspond qualitatively to the results of the VGA sequence and are therefore not presented here. The objective quality metrics PSNR and SSIM are approximations of the perceived quality. For example, to obtain an SSIM index of 0.95 for the VGA sequence, the data rate of an



(a) PSNR values



(b) SSIM values

Figure 4: Video Quality Evaluation: PSNR and SSIM values for the VGA sequence for data rates 2.5 to 8.5 Mbit/s

M-JPEG coded sequence has to be more than three times higher than with H.264/AVC and more than twice in comparison to the MPEG-4 coded sequence. The difference is higher in the lower bit rate range. Yet, in our analysis the superiority of H.264/AVC in terms of quality is not as clear as it might have been expected. This is due to several reasons. First, B frames were not used for prediction due to the stringent delay requirements and the I frame interval of 15 frames is rather low[3]. Second, due to the low complexity requirements, the motion estimation process was carried out with algorithms that offer a trade-off between complexity and efficiency. Third, some advanced coding tools, such as CABAC for H.264/AVC, were not used due to restrictions to the features of the Baseline profile. Also, parameters such as the content of the video sequences and artifacts in the sequences might have impacted the motion estimation efficiency.
Although H.264/AVC is superior to all other compression systems, its significantly higher processing requirements for encoding and decoding do not justify its employment as software codec in the discussed real-time scenario. As expected, the performance of MPEG-4 lies between that of MPEG-2 and H.264/AVC. Also, the two MPEG-4 implementations show very similar results. While the wavelet based Dirac codec performs below any of the other compression schemes and its encoding and decoding times are about a factor ten higher than those for the DCT based compression schemes, the performance of the Theora codec is almost equal to that of MPEG-4 in terms of perceived quality. Its encoding time was in the range of the x264 implementation of H.264/AVC. Since the current release is only an alpha version, Theora shows high potential as a (free) alternative to MPEG-4. However, due to their insufficient timing performance Dirac and Theora are not further analyzed in the present work.

3.3.2 Error-Prone Wireless Networks

For the tests in an error-prone network, the following scenarios and transmission strategies have been considered:

U: Simple UDP transmission without a sequence header field and with packet loss. Each frame was split into packets with the maximum payload size (1472 bytes) and sent to the client in the order of encoding. The receiver was configured to separate the frames within the bit stream upon reception of a packet with a length less than the maximum packet size. Additionally, a packet of length eight bytes containing only "00" values was sent after each frame, to help the receiver to separate the frames if the last packet of a frame was lost. A detection of lost packets is obviously not possible without additional information. Analysis results are shown in Fig.5.

I: Transmission via UDP with sequence headers and packet loss. The receiver was configured to ignore losses within a unit and pass all of the correctly received packets to the decoder and to let the decoder decide how to deal with the lost packets. Analysis results are shown in Fig.6(a).

D: Transmission via UDP with sequence headers and packet loss. The receiver was configured to discard any units in which one or more packets were lost. Only the correctly received units were passed to the decoder. Analysis results are shown in Fig.6(b).

Channel measurements in the car have shown that more than 90% of all lost packets are single losses. Instead, burst losses represent the smallest part of the in-vehicle packet losses. Accordingly, the simulations were carried out for random single packet losses.

Loss probabilities: The analyses for each scenario (U, I and D) in the testbed from Section 2.1 were conducted for five packet loss probabilities: 0.1%, 1%, 2.5%, 5%, and 10%. For the simulation of packet losses and jitter in the network, NetEm [19] was used. NetEm is a tool to emulate various types of networks. Amongst others, rules can be set for the packet delay and delay variations (jitter), packet loss, and packet corruption. NetEm delays or discards packets on the network layer in a way transparent to the application layer. Compared to other common network emulation tools, NetEm provides the advantage of being a part of the 2.6 Linux kernel. It does not have to run on a separate computer and it does not considerably impact the CPU load. It allows to set rules for both the incoming and outgoing traffic on the specified network interfaces. Also for NetEm, a Linux kernel HZ value of 1000 is necessary in order to allow a granularity of 1 ms on the packet delay values.

Test sequence parameters and codecs: The tests for different scenarios and loss probabilities were conducted for M-JPEG, MPEG-2, and for the two implementations of MPEG-4 on the basis of the VGA and the Highway sequence. In the following, only the graphs for the VGA sequence are included, since the simulations for the Highway sequence yielded similar results. In order to make the results comparable in terms of distortion, the codecs were configured to yield an SSIM of 0.95 when varying the PSNR and a PSNR of 40 dB when varying the SSIM according to Table 1. Error concealment strategies in the decoder were activated. For the error-prone simulations, the H.264/AVC decoder has not been studied due to the insufficient performance of the x264 software codec.

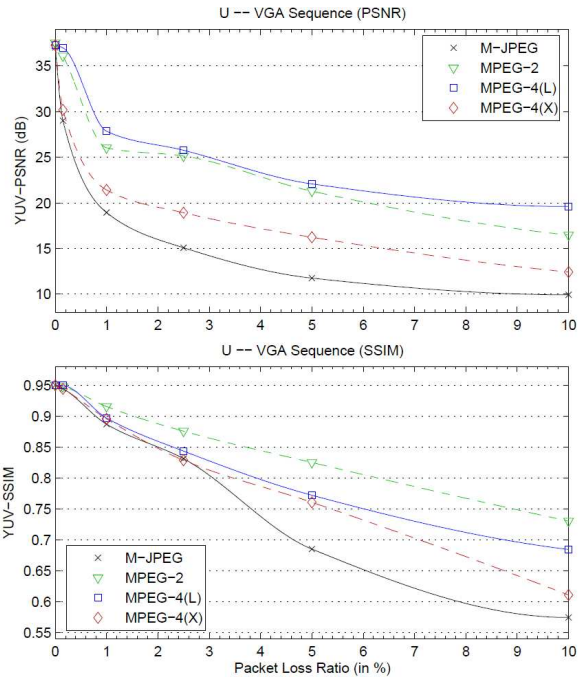
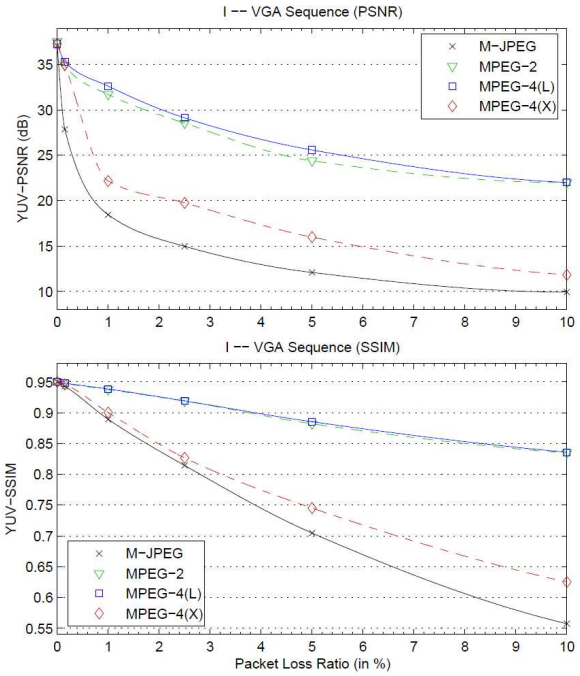


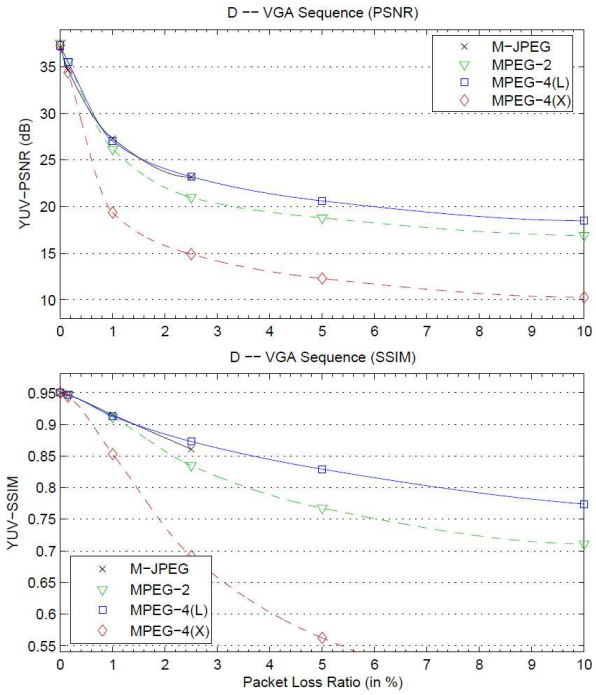
Figure 5: Error Robustness: VGA sequence transmitted via UDP without special packetization.

M-JPEG: In the case of M-JPEG, no splitting into individual units was possible, such that a unit consisted of a whole frame. In the case of discarding an incomplete unit, at a loss probability of

[3]In order to limit the temporal propagation of errors due to packet loss and to support multiple clients with a low resynchronization time, the I frame interval has been set to 15 frames



(a) Incomplete units ignored.



(b) Incomplete units discarded.

Figure 6: Error Robustness: VGA sequence transmitted with sequence header fields and unit based packetization.

5% only an average of 200 and for 10% loss only 50 out of the 715 frames were received correctly. Since a "moving" video could not be displayed anymore, the quality indices are not considered in the graphs.

Generally, the M-JPEG codec performed worst in this simulation, even though it is based on still images and therefore, there is no temporal propagation of errors. A reason is that the compression efficiency of M-JPEG is lower than the MPEG compression schemes and therefore, with the higher number of packets, more frames and multiple parts of a frame are affected by the errors. Also, the error resilience of M-JPEG may be improved in different implementations with the use of restart markers in the bit stream and with an adapted packetization rule. However, for scenario "D" in which a unit containing lost packets was discarded (the whole frame in the case of M-JPEG), Figure 6(b) shows an advantage of using a codec based on still images, since at low packet loss ratios, the quality degradation is among the lowest. This is because any frames exhibiting errors were discarded and the last correctly received frame was left on the screen. Therefore, the errors did not directly become visible. Though, this simple error concealment strategy is only applicable at low packet loss ratios, since it results in heavy motion judder.

MPEG Compression: Comparing the two MPEG-4 implementations, the libavcodec implementation outperforms Xvid significantly. The reason is the missing support of the error resilience functions in the Xvid MPEG-4 implementation. Additionally, because of the lack of slice boundaries, decoding errors due to lost blocks propagate throughout the frame. Also, Xvid does not allow for a reasonable slice based encoding and packetization which becomes especially clear when comparing the graphs of UDP from Fig.5 and the sequence header mode transmission from Fig.6.

The libavcodec implementation, on the other hand, supports the data partitioning tool and even more important, it allows a flexible sliced encoding in such a way that the number of the macroblocks in a slice is adapted to the packet size and coefficients prediction is limited to the slice boundaries. Therefore, the errors due to the loss of a packet only affect the respective slice and do not propagate in the whole frame. For MPEG-2, a similar packetization pattern was used with the only difference that the number of macroblocks per slice is fixed. For the simulations, the number of slices transmitted per packet was adjusted such that the resulting packet size was as close as possible to the maximum packet size. In the case of larger slices than the packet size, a slice was split into several packets. As stated before, the slice based packetization rules were only implemented for the sequence header mode. Due to similar packetization rules, the results of the packet loss simulations for MPEG-2 and MPEG-4 (L) are comparable, yet, MPEG-4 always yields less quality degradations. In the UDP mode, in which a frame was packetized into equally sized packets without regard to syntax boundaries, the quality degradation was worse for any of the codecs compared to the sequence header mode. Only in the UDP mode, the SSIM index suggests that the quality degradation is less visible for MPEG-2 than for MPEG-4 (see Fig.5).

Packetization and Application Layer Protocols: The simulations of packet losses clearly show that an appropriate packetization scheme together with an additional protocol that supplements UDP, e.g., RTP or the simple header mode introduced in this work, help to reduce the negative impact of transmission errors. A supplementing higher level protocol is particularly necessary in the case of jitter and packet re-ordering during transmission. Using the sequence header information, the mode "I", in which it was left to the decoder to decide upon a good strategy to deal with partly received units, yields the lowest quality degradation and thus the highest level of

error robustness for MPEG compression. For M-JPEG though, the best results could be obtained by discarding the incomplete frames, yet this is only feasible at low loss ratios.

4. HARDWARE IMPLEMENTATION CONCEPTS FOR IP CAMERAS INCLUDING VIDEO CODECS

Designing an IP camera with video codecs for automotive applications can basically follow two design methods. The first one would be a highly customized solution to fulfill the requirements of the automotive sector while the second one is based on the utilization of hardware and software toolboxes of the consumer electronic industry that should be set accordingly, in order to meet the automotive requirements.
The customized solution would meet the automotive requirements, but can be more cost intensive and inflexible while the utilization of consumer electronic solutions would be more promising in terms of flexibility and low cost thanks to the economy of scale caused by the consumer electronic market. However, meeting the automotive requirements with the consumer electronic tools is challenging. Both realization concepts with some examples are briefly described in the following.

4.1 The Customized Solution - FPGA/ASIC Implementation

In the case of customized solution, the compression algorithm, the communication protocols such as RTP/UDP/IP, the Ethernet MAC, and if required the image processing algorithms should be integrated in the FPGA/ASIC design. Most of the mentioned components which have to be integrated in the FPGA/ASIC are already available as IP cores. Important electronic components which are left out of the FPGA/ASIC are the power supply, voltage control, the Ethernet PHY, and the camera sensor. An example for the CMOS camera sensor which is able to deliver a video sequence with VGA resolution and 30 frames/s is the LM9628 sensor from National Semiconductor [21]. Generally, the FPGA/ASIC is connected via an 8 to 12 bit data bus and is controlled via an I^2C bus.

4.2 Solutions from the Consumer Electronic Industry

The application of the consumer electronic hardware and software toolboxes leads to two different solutions.

4.2.1 Multimedia Processor Solution

A multimedia processor based solution provides a higher flexibility than the FPGA/ASIC based solution. The current multimedia processors are equipped with all necessary features for building an IP-based camera system. They include MPEG-4 and H.264 Baseline and main profile codecs for video compression with a VGA resolution and 30 frames/s transmission rate, an integrated ARM processor for applying the TCP/IP stack and implementing application layer protocols such as RTP and RTSP and an Ethernet MAC for the network connection.
The integrated system on chip design of the multimedia processors offers the possibility to reach very small dimensions of the PCB (printed circuit board) layout which is also a very important requirement for an in-car camera system. Similar to the ASIC solution, electronic components that are left out of the processor are the camera sensor, power supply, voltage control and the Ethernet PHY. Several well-known multimedia processor manufacturers such as Qpixel Technology and Freescale Semiconductor have already presented promising solutions.

4.2.2 CMOS On-Chip Compression Solution

A further interesting solution is the utilization of a CMOS camera sensor with an integrated video codec. So far, it has been only possible to integrate the M-JPEG compression algorithm because of its low complexity compared to the MPEG family compression algorithms. An example is the VS6724 single-chip camera module from ST microelectronics [20] shown in Fig.7. This kind of sensor

Figure 7: VS6724 single-chip camera module from ST microelectronics. The left picture shows the whole circuit board while the right picture shows the image sensor.

provides a M-JPEG video compression up to a SVGA (800x600 pixel) resolution. It also integrates digital image processing functions like lens shading correction, sharpening etc. The sensor can be controlled by an I^2C interface and provides for the video data an 8-bit parallel video interface. For a complete IP camera implementation, a processor with sufficient processing power such as the ARM9 processor should be added to the sensor in order to packetize and send the compressed images captured by the sensor via an Ethernet interface. Assuming that the Ethernet MAC is also provided by the applied processor, the remaining electronic components to create an IP camera will be similar to the multimedia processor solution mentioned before.

5. CONCLUSION AND OUTLOOK

The performance of software based video codecs has been analyzed to introduce video compression in driver assistance camera systems of future cars with a convergent IP network system for viewing purposes. It turned out that the MPEG-4 (Part 2) compression algorithm leads to the best trade-off between the quality of service and complexity values. It fulfills the strict delay, complexity and quality requirements of in-vehicle cameras while at the same time, it provides an adequate compression ratio of 0.066 (mean bit rate of 7.4 Mbit/s) for a PSNR value of 40 dB and a SSIM value of 0.967. A compression ratio of 0.032 delivers a bit rate of 3.6 Mbit/s on the transmission link for PSNR and SSIM values of 37.3 dB and 0.95, respectively. In order to further improve the compression performance and be able to apply more complex codecs such as H.264/AVC that is dominating the consumer electronic market, several hardware implementation concepts for IP cameras with integrated video codecs have been presented. In our future work, the mentioned hardware realization concepts will be precisely evaluated in order to define the maximum achievable processing power to select the most appropriate codec for the application in the car. Also, the influence and effects of video compression on automotive image processing algorithms will be explored in order to define

the minimum acceptable video quality for driver assistance applications.

6. REFERENCES

[1] Robert Bosch GmbH, CAN (Controller Area Network) Specification, Version 2.0, http://www.semiconductors.bosch.de/pdf/can2spec.pdf, 1991.

[2] MOST Cooperation, MOST (Media Oriented Systems Transport) Specification, Version 2.4, http://www.mostcooperation.com/downloads/Specifications, 2005.

[3] LIN Consortium, LIN (Local Interconnect Network) Specification package, Revision 2.0, http://www.lin-subbus.org, 2003.

[4] Mehrnoush Rahmani, Joachim Hillebrand, Wolfgang Hintermaier, Eckehard Steinbach and Richard Bogenberger, A Novel Network Architecture for In-Vehicle Audio and Video Communication, *IEEE Broadband Convergence Networks Workshop*, Munich, Germany, May 2007.

[5] Mehrnoush Rahmani, Bernd Mueller-Rathgeber, Wolfgang Hintermaier and Eckehard Steinbach, Error Detection Capabilities of Automotive Network Technologies and Ethernet - A Comparative Study -, *IEEE Intelligent Vehicles Symposium*, Istanbul, Turkey, June 2007.

[6] Zhou Wang and Alan C. Bovik and Ligang Lu, Why is image quality assessment so difficult?, *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '02)*, volume 4, pages 3313-3316, http://ieeexplore.ieee.org/xpl/abs free.jsp?arNumber=1004620, 2002.

[7] Zhou Wang and Ligang Lu and Alan C. Bovik, Video quality assessment based on structural distortion measurement, *Signal Processing: Image Communication, special issue on objective video quality metrics*, volume 19 no. 2, pages 121-132, http://www.cns.nyu.edu/ zwang/files/papers/vssim.pdf, 2004.

[8] Bernd Girod, What's wrong with mean-squared error?, *Digital Images and Human Vision*, MIT Press, pages 207-220, 1993.

[9] FFmpeg Multimedia System, http://ffmpeg.mplayerhq.hu, 2006.

[10] VideoLAN Project, x264, http://www.videolan.org/developers/x264.html, 2006.

[11] The Imaging Source, Products, http://www.theimagingsource.com/en/products, 2006.

[12] Arne Caspari, unicap library, http://www.unicap-imaging.org, 2006.

[13] Xvid, http://www.xvid.org, 2006.

[14] Arizona State University, Aalborg University, and acticom GmbH. YUV video Sequences, http://trace.eas.asu.edu/yuv/index.html, 2006.

[15] Xiph.org Foundation, Theora I specification, http://ww.theora.org, March 2006.

[16] BBC Research, Dirac-Online documentation, http://dirac.sourceforge.net/documentation.html, 2006.

[17] LIVE555 Streaming Medialibraries, Online documentation, http://www.live555.com/, 2007.

[18] Testsequences, ftp://ftp.ldv.e-technik.tu-muenchen.de/pub/test_sequences/601, 2007.

[19] Stephen Hemminger, Network emulation with NetEm, http://linux-net.osdl.org/index. php/Netem, April 2005.

[20] ST Microelectronics, User manual UM0469, November 2007.

[21] National Semiconductor, News Release, http://www.national.com/news/item/0,1735,759,00.html, May 2002.

[22] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, G. J. Sullivan, Rate-Constrained Coder Control and Comparison of Video Coding Standards, *IEEE Transactions on Circuits and Systems for Video Technology*, volume 13 no. 7, pages 688-703, July 2003.