A Robust Method for Frozen Frame Detection in Safety Relevant Video Streams Based on Digital Watermarking

Kevin Witt *, Jan Bauer ♦
*** Maxim Integrated, San Jose, CA, United States**
♦ Daimler AG, Sindelfingen, Germany

Abstract
Digital watermarking is a promising method for detection of frozen video frames in a video system. We present a real-time video watermark generation and detection method that is immune to processor buffering, video compression, common image processing operations and is robust with a low probability of false detections. By temporally distributing the impact of the watermark and using objective methods, we show that the visual effect of the watermark is below the perception threshold.

1. Introduction
Frozen frame detection is an important capability for automotive video distribution systems. Modern video system architectures route safety relevant video streams (e.g. automotive backup camera) through processors (e.g. automotive central computing unit) that have multiple video frame buffers and often do not follow a safety relevant implementation like ASIL [1]. Due to the missing ASIL implementation, the processor routing the safety relevant video streams cannot be "trusted" and require further measures.

If the input video processing stops or a corruption happens within the processor, while output processing continues to transmit from the output frame buffers, many frozen frame detection algorithms can fail. Furthermore, unintended video processing, video overlays, or video compression (like VESA Display Stream Compression (DSC) [2]) can defeat most frozen frame detection algorithms based on CRCs or modification of unique pixels.

Utilizing the blanking area of the video stream for the frozen frame detection is not an option since the video processor typically removes the blanking area. Hence, the frozen frame detection information has to be tightly woven into the visible area of the safety relevant video stream.

In a safety relevant system, a frozen video stream must be detected quickly to prevent an unsafe situation. For example, an automotive backup camera displayed on a center console must always display a valid video image or change to a safe state (e.g. black screen) quickly. The ISO 26262 requires reaching the safe state in less than 500ms [1].

This paper describes the implementation of a real-time digital watermark generation and detection system to detect frozen video in an automotive video system.

2. System Overview
The watermark-generator inserts a line-based watermark into each pixel. The same watermark is inserted into each line of the frame. A different watermark is inserted in subsequent frames from a set of unique watermarks, so that each frame is distinguishable in time. Because each frame contains thousands of redundant watermarks, detection robustness is provided. The integration into each pixel enables a complete coverage of the safety relevant content. An example watermark is illustrated in Figure 1.

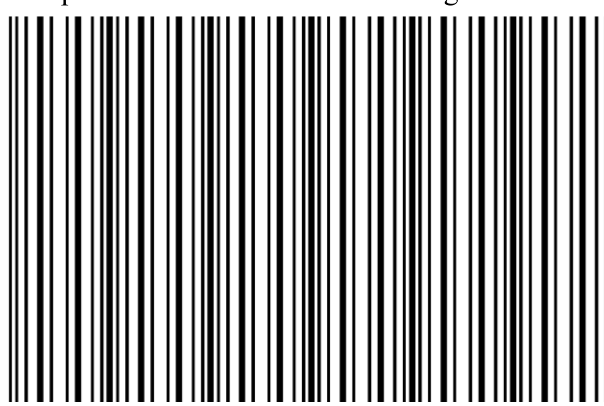

Figure 1: Example Watermark, a Watermark with a length of 320 pixels is shown repeated 5 times across each line of the image above which has 1224 rows by 1632 columns.

The watermark-generator imbeds the watermark in the least significant bits (LSBs) of the chroma to minimize the visual detectability. In the watermark-detector, the least significant bits of the chroma are removed from watermarked video before it is displayed. The watermark generator uses frame rate control (FRC) [3] to temporally spread the chroma information loss caused by imbedding the watermark.

Using the combination of imbedding the watermark in the LSBs of the chroma, the removal of the watermark and the error compensation with frame rate control makes the watermark processing visual undetectable.

A block diagram of such a system is illustrated in Figure 2. As illustrated in this example, the watermark is inserted at the camera, and its safety relevant video is routed through a processor with a system on chip (SOC) to a display. The watermark detector is located at the display electronics receiver and it continuously monitors the incoming video stream for the presents of a watermark. If a watermark is present, the detector monitors if the watermark changes in subsequent frames.

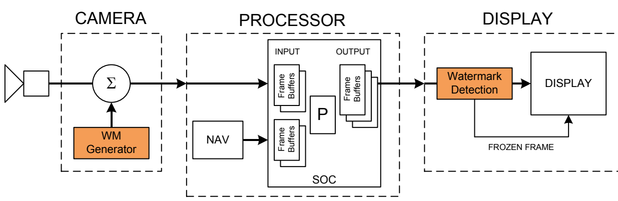

Figure 2: Example Video Distribution System. Having the Watermark Generator at the source and the Watermark detector at the sink enables an end-to-end validation of the safety relevant video.

The presence of a watermark identifies safety relevant video and the frame-to-frame change of the watermark ascertains that the video stream is not frozen. If a frozen frame condition is detected, the display can be blanked or the supervisor processor (e.g. within the display) can be notified to initiate the switch to the safe state.

3. Watermark Design and Detection

The watermarks used can survive image manipulation, image compression and have a high degree of redundancy and detectability. They are generated in real-time from a set of basis functions stored in register memory with a length of 32 to minimize hardware cost and power requirements. The basis functions are a concatenation of a pseudo-random bit-stream with a time-reversed copy of the same pseudo-random bit-stream.

Unique features of these basis functions enable them to survive image manipulations (in the following discussion the image manipulation will be called stressors). The watermark is vertically (by line based repeating) and horizontally (by design of the watermark) symmetric. This makes them immune to horizontal or vertical flipping of the image. They are highly redundant (e.g. a Full HD Frame contains 6480 Watermarks) across the row and frame, enabling it to be robust to video overlays that may cover a large percentage of the image.

The number of LSBs replaced serves as the watermark's gain K_o; the more LSBs used the higher the detectability and robustness. However, this increases the chance of visual detection.

A challenging stressor for the watermark to survive is image resizing. Therefore, we apply oversampling to the basis function to reduce the spatial frequency content. In addition, we use several filters within the detector to detect the resulting length of the basis function due to the image resizing. The watermark's redundancy/robustness and oversampled qualities leads to an immunity against noise and image compression.

To detect the watermark, the watermark I_K is extracted from image data I and convoluted with the basis functions W_k with $k=1...n$, where n is the number of basis functions.

$$M_k = I_k * W_k$$

Many video processors use triple buffering to synchronize the video streams between two domains (e.g. between camera and video processing unit); therefore, we have chosen the number of unique watermarks $n \geq 4$ to ensure the detection of a frozen image. The four basis functions used are:

W_1=0x5B 9B D9 DA

W_2=0xA4 64 26 25

W_3=0xBB 8D B1 DD

W_4=0x44 72 4E 22

The matched filter output M shows a high correlation if the watermark is present in the image data. In ideal conditions in the video processing path, the result of the convolution will be $M=L$, where L is the length of the basis function W at the position where the extracted image data matches the basis function exactly. To enable a loss of watermark information in the video processing path (e.g. due to noise or compression), a threshold T is defined with $T \leq L$. Watermark detection occurs when the convolution result is higher than the threshold:

$M \geq T$: Watermark detected

$M < T$: Watermark not detected

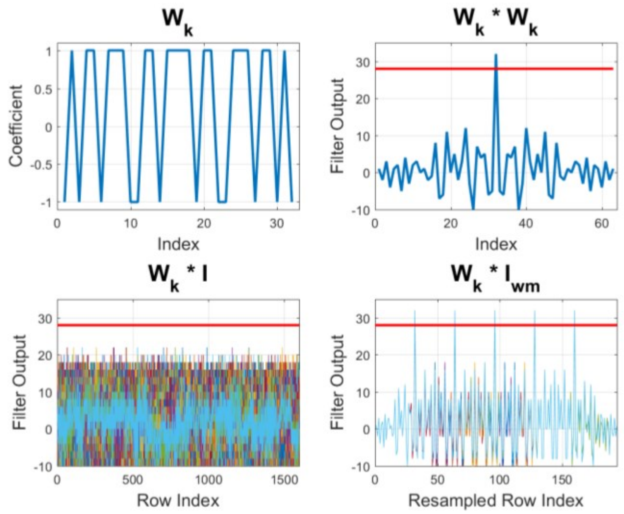

Figure 3: Example Watermark Basis Function W_k and the Matched Filter's M_k output to its' Basis Function W_k, Non-Watermarked Image data I and Watermarked Image data I_{wm}.

Figure 3 illustrates an example basis function W_k (top left), the matched filter output M to the basis function W_k (top right), the matched filter output M to video image data I without a watermark (bottom left) and the matched filter output M to watermarked image data I_{wm} (bottom right.)

As shown in the example the matched filter output M reaches the threshold when the basis function (watermark) is present, but stays below the threshold T otherwise. With the threshold T the false-positive and false-negative detection rate is lowered. Depending on the application requirements, the threshold T can be adjusted accordingly to make the detection more strict (higher chances of false negative) or more loose (higher chances of false positive).

4. Watermark Generator

Figure 4 illustrates the watermark generator architecture. The Input block converts the Video Stream from RGB into the YCbCr color space. Then the desired LSBs K_o of the chroma Cb[1] are replaced with the watermark W_k. Different watermarks are inserted in subsequent frames to make the watermark distinguishable over time.

The difference between the original chroma and the masked chroma information is the basis for the frame rate control (FRC) pattern look-up-table (LUT).

[1] The chroma information Cb is used here as an example, Cr could be used in the same manner.

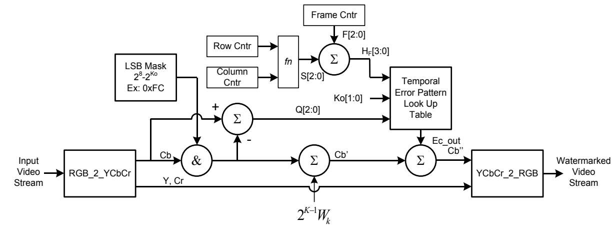

Figure 4: Watermark Generation, usage of basic functions enables a hardware efficient and low latency implementation.

Frame rate control spreads the error caused by the removal of the watermark over subsequent frames. The error is added to the MSBs of the chroma. The value added is a function of the watermark gain K_o, error, frame index and relative pixel position in the frame.

The error is spread over 2^{K_o} subsequent frames (e.g. 4 frames for $K_o=2$). The spread of the error by FRC results in an average chroma after watermark removal equal to the original value for still images and nearly the original value for moving pictures. Optimal spatial error diffusion is used to minimize flickering that can potentially occur in solid colors [3]. Finally, the video stream is converted back from YCbCr to RGB color space.

5. Watermark Detector

The watermark detector consists of three processes:

- extraction,
- row-based match filtering, and
- frame-based filtering.

Watermark Extraction

In the extraction process, each row of video is converted to YCbCr format and the LSBs of the chroma are filtered and resampled at a sample rate above and below the oversampling rate of the watermark generator. The resampled data is processed with a bank of matched filters optimized for each watermark basis function.

Match Filter Design

The matched filters M_k produce an optimal response when the watermark basis function is present in the video. The match filter is a Finite Impulse Response (FIR) filter. This filter is implemented on the extracted bit stream I_{wm} A custom digital logic implementation, illustrated in Figure 5, minimizes power and cost. The advantage of this approach over alternatives is its efficiency. Because the input data and the FIR coefficients are encoded as +/-1, the required memory is implemented with a simple shift register and the multipliers are implemented with 4 simple logic gates. The adder tree is implemented in parallel and bit width is only grown as needed.

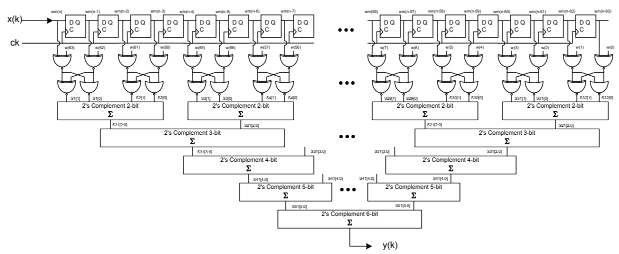

Figure 5: Matched Filter Implementation. The matched filter output convolution of the basis function with the extracted image data.

Image resizing is a challenging stressor to mitigate. As illustrated in Figure 6, the oversampled extracted bit stream is resampled at a variety of decimation rates ***Ni*** and processed in a bank of matched filters. Each decimator and match filter produces an optimal output for a range of image resizing. The parallel bank of filters enables support for 50% image reduction up to 200% enlargement.

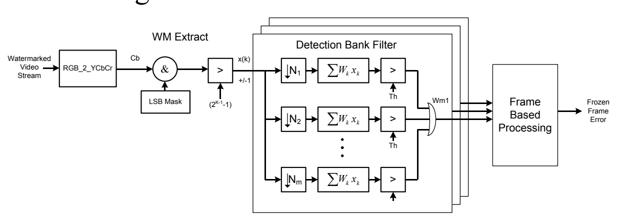

Figure 6: Detection Filtering. The watermark is extracted from LSB of the chroma, resampled to remove the oversampling, then processed with a bank of parallel matched filters. The outputs are compared to a threshold to suppress false alarms.

The output of each match filter in the bank filter is compared to the threshold ***T*** to minimize false alarms and then or-ed together to determine if a watermark occurred in a video frame. Each match filter in the bank filter has a different decimation rate. This feature leads to robustness to image resizing as illustrated in Figure 7. A bank filter is implemented for each unique watermark generated.

Frame-based Processing.

Within each frame the number of detected watermarks is accumulated. On the vertical sync, the accumulated watermark detections are compared to a frame-based threshold T_F to differentiate a watermark detection from a potential false positive.

Figure 7: Bank Filter Response for different Image Scale Factors. For each image resize factor a match filter with a different decimation rate responds to the imbedded watermark, resulting in robustness to image resizing.

Additionally the presence of all unique generated watermarks is observed. If the number of observed watermarks n_o is less than the number of generated watermarks ***n*** (this means a frame is lost e.g. due to a failing triple buffer), an error is generated.

To further increase robustness, a programmable time-based filter is applied to the error signal. This filter requires the error condition to exist for a programmable amount of time d_F (with d_F < 500ms to satisfy the ASIL requirement) before an external error is asserted. This filter makes the system robust to a frame-based false negative (e.g. due to short errors in the video processing chain.)

A key aspect of this system is the utilization of a set of watermarks larger than the number of frame buffers utilized in any video processing stage. Video frame buffers are of particular concern because an input process could fail and leave unique frames in the buffers. If this were a watermarked video stream, a different watermark would be detected for each frame.

By generating more watermarks than the number of frame-buffers at any stage in the video processing system and requiring detection of all generated watermarks, the system can detect a frozen state in a video processor that has a multitude of frame buffers.

6. Results

The removal of the watermark would degrade the output image quality without compensation. The combination of embedding the watermark in the chroma, frame rate control error compensation, and optimized error diffusion makes the watermark processing visual undetectable as shown in Figure 8.

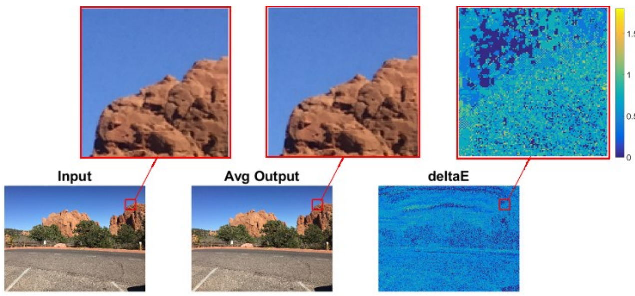

Figure 8: Example Image Before and After Watermark and DSC Processing. The original image (left) without watermarking, the average output of the watermarked image after DSC w/ watermark removal (middle), and the calculated ΔE for each pixel (right) illustrate the impact is not visually detectible.

For the calculation of the ΔE, we use the LAB color space, an exemplary sRGB display at a white point of D65. As mentioned in previous work, [4] a color difference smaller than ΔE≤2.3 is visually undetectable.

The ΔE of the time-averaged image after watermark removal with and without DSC results in an average ΔE of 0.84 and 0.57 respectively for 10 images and a watermark gain of 1, 2, 3 bits as shown in Figure 9. The images analyzed are typical automotive safety relevant scenes like rear view camera views and surround view displays. For these images, the impact of watermark and DSC processing is visually undetectable.

Detection and False Alarm performance

A MatLab model and the RTL implementation of the system were tested with a variety of video clips and stressors including:

- Image resizing (50, 75, 125, 175, and 200%)
- Image rotation (+/- 5degree),
- VESA Display Stream Compression (DSC),
- Obstructions of 25% of the image area,
- Contrast adjust,
- Color compensation,
- Image horizontal and vertical flipping.

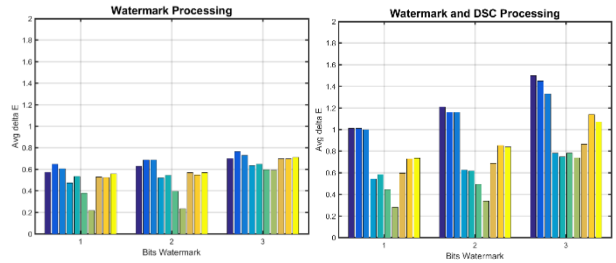

Figure 9: Watermarked Image Quality Impact w/ and w/o DSC. All tested images a have an average ΔE_{LAB} below the perception threshold.

Each video clip is eight frames long and the pre-stressed resolution is 1224 by 1632. Figure 10, 11 and 12 illustrates the frame-based watermark detections and false alarms for the RTL implementation with image resizing, DSC and image rotation. Because the image size changes, the detection ratio is normalized to the detection ratio needed to meet the frame-based threshold T_f. The false positives illustrated are detections of watermarks that were not generated in a frame. Color differentiates the watermark gain: blue, green and black represent 1, 2 and 3 bit gain levels.

The frame-based detection threshold T_f was chosen to be 32 to minimize false positives and maximize detections on image-processed video.

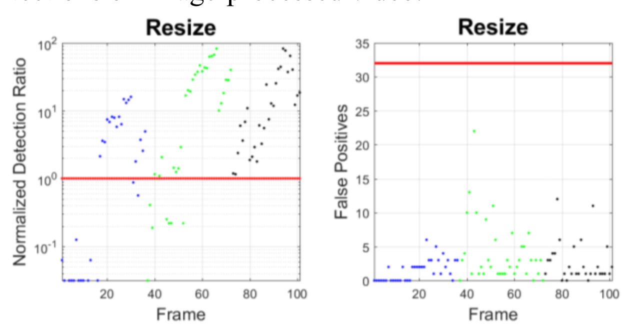

Figure 10: Watermark Detector Performance to Image Resizing (scale factor of 50, 75, 125, 175 and 200%) with a Gain K_o of 1, 2 and 3 bits.

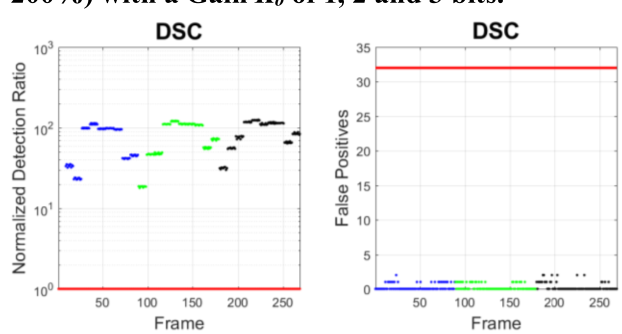

Figure 11: Watermark Detector Performance to Display Stream Compression with a Gain K_o of 1, 2 and 3 bits.

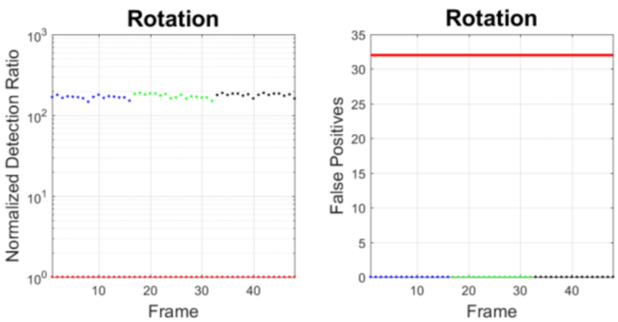

Figure 12: Watermark Detector Performance to Image Rotation (+/-5 degrees) with a Gain K_o of 1, 2 and 3 bits.

The data demonstrates most test cases have an order of magnitude margin to the required frame based detection threshold T_f.

The tests were designed to determine the limits of the system. Image reduction (50% or 75% using MatLab's interpolating image resizing function) caused lower margins with a 3-bit watermark and false negatives with a 1-bit or 2-bit watermark gain. Closer inspection revealed an interaction with the frame-rate control algorithm and the interpolation filter used for image resizing. The net effect is an attenuation of the effective watermark gain.

False positives are the detection of a watermark not transmitted in a frame. On the right hand side of Figure 10, 11 and 12, the total number of false positives is typically an order of magnitude lower than the frame-based threshold required to generate a false alarm

Non-watermarked video data (249 frames) processed by the detector yielded no frame-based false positives. The margin to the frame-based threshold was typically an order of magnitude.

The MatLab and RTL simulations lead to the choice of 3-bits for the watermark gain because it passed all stressed image test and has high visual quality.

7. Conclusion

In this paper, we presented a watermarking generator and detection algorithm for safety relevant video content in automotive environments with the following key features:

- high robustness against image processing
- complete coverage of the safety relevant content
- robust detection with low false negative and low false positive detections
- high visual quality of the video content
- efficient hardware implementation for ASIC or FPGA designs

Even with challenging stressors it was shown that the detection rate was 100% in all test cases and no false alarms were generated.

Objective methods show that the visual impact of the watermark embedding and removal is below the perception threshold due to optimized error distribution with frame rate control. We anticipate high customer satisfaction as well a safe detection of frozen images.

The watermark algorithm was tested in a FPGA implementation where the hardware efficiency was proven. Tests with frozen frames and image stressors were successfully performed.

We would like to encourage further work in designing different watermark basis functions within different video image representations domains like wavelets or other color spaces. Other algorithm designs can challenge the results from the paper for hardware efficiency, visual quality and detection rate as a benchmark.

8. References

[1] "ISO: 26262 - Road vehicles-Functional safety", 2011

[2] "VESA Display Stream Compression (DSC) Standard", Version 1.1, http://www.vesa.org/, 2014.

[3] J. Bauer, M. Kreuzer, T. Jung, D. Schäfer, "Increasing the Perceived Grey Value Resolution by Combining Frame Rate Control and Error Diffusion to Reduce Visible Artefacts in Local Dimming Applications", electronic displays Conference 2015, Nuremberg, Germany, February 25-26, 2015

[4] M. Mahy, L. Van Eyckden, and A. Oosterlinck, "Evaluation of uniform color spaces developed after the adoption of CIELAB and CIELUV," Color Res. Appl., vol. 19, no. 2, pp. 105–121, Apr. 1994.