Watermarking of Raw Digital Images in Camera Firmware: Embedding and Detection

Peter Meerwald and Andreas Uhl

January 15, 2009

Overview

- Application Scenario and Prior Work
- Camera Image Processing Pipeline
- Firmware Architecture building on CHDK
- Watermark Detection based on Fused Image Components
- Results and Impact of Demosaicking

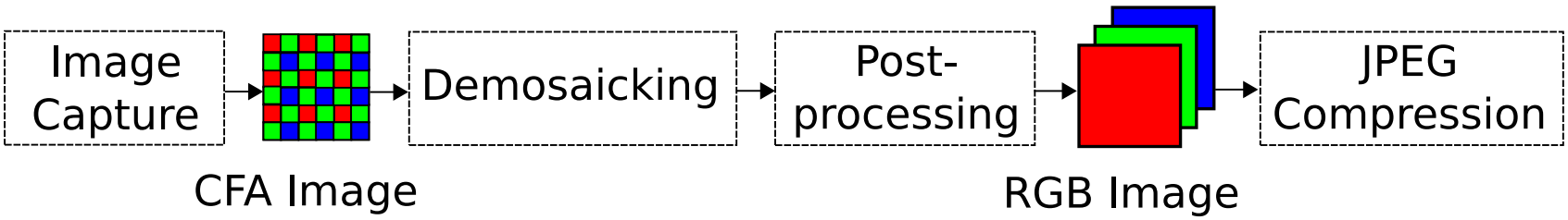
Application Scenario

- Digital images can be easily copied and tampered with
- Active and passive methods have been proposed for copyright protection and integrity verification: watermarking and forensics (e.g. Photo-Response Non-Uniformity (PRNU) for camera identification [Chen et al., 2008] or verification of CFA interpolation patterns [Popescu and Farid, 2005])
- Raw image data probably the most valuable digital image asset
- This work: simple watermarking in camera firmware to protect raw image data

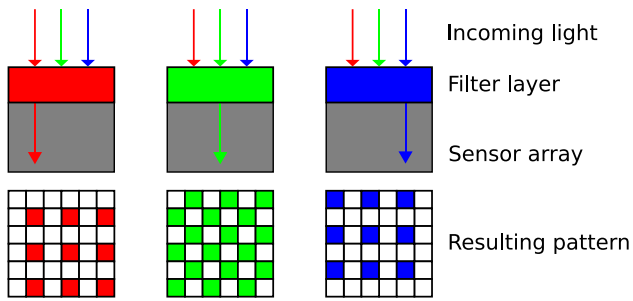
Watermarking: Related Prior Work

- [Blythe and Fridrich, 2004] Capture human iris image through viewfinder, embed in camera image together with camera identification and image hash
- [Lukac and Plataniotis, 2006] Emboss visible watermark in CFA domain
- [Mohanty et al., 2007] VLSI architecture for robust and fragile watermarking
- [Nelson et al., 2005] CMOS image sensor adds pseudo-random watermark to raw data
- Kodak and Epson offer cameras with watermarking capabilities (2003, discontinued?); patents!
- Many JPEG-domain watermarking algorithms may be applicable but do not protect the raw image data

Camera Image Processing Pipeline



- Most cameras use single image sensor with color filter array (CFA)

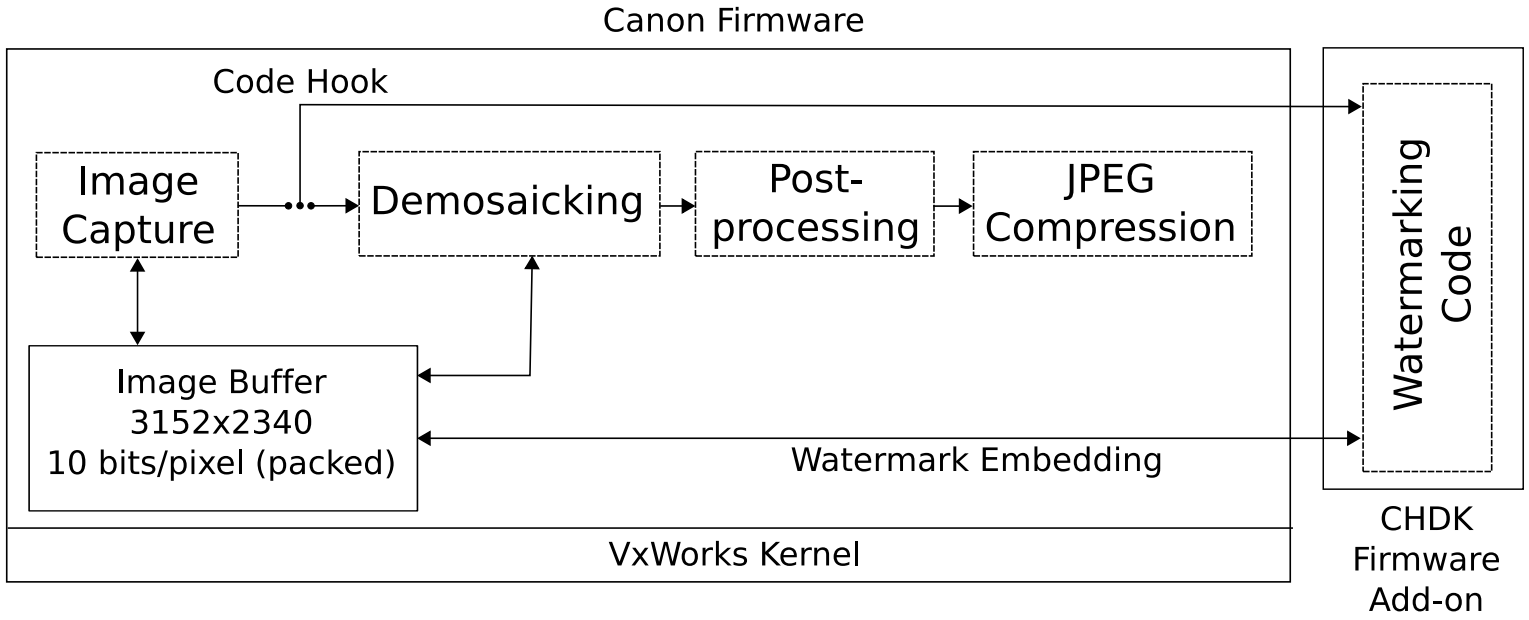


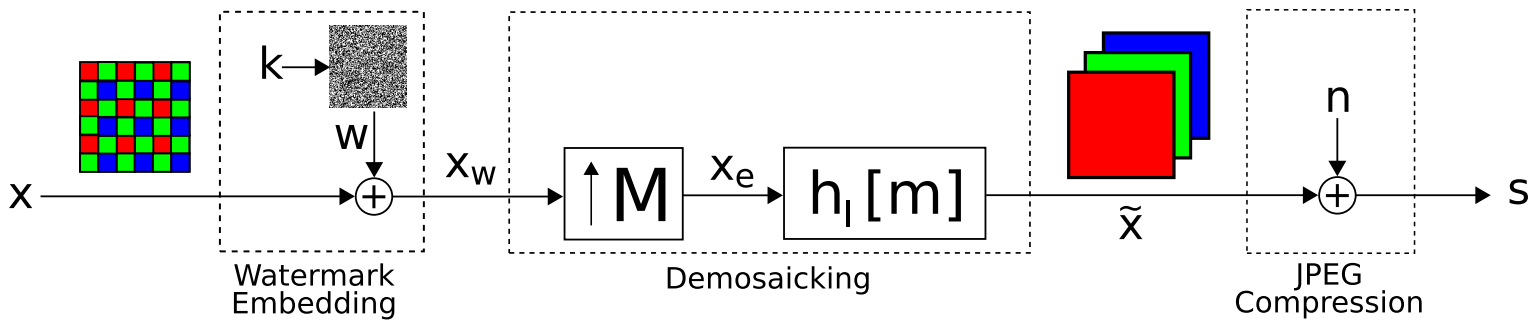
- Demosaicking is basically interpolation to get full-resolution RGB image
- Many different demosaicking approaches (bicubic, gradient-based, adaptive homogeneity-directed, . . .)
- Actual camera implementation is unknown

Access to the Camera Pipeline: CHDK

- CHDK: open-source firmware add-on for Canon DIGIC II and DIGIC III cameras, `http://chdk.wikia.com`
- Target ARM9 CPU core with custom hardware, VxWorks operating system
- Adds bracketing of exposure, RAW file support, BASIC scripts, remote camera control, additional data display (histogram, battery life), longer exposure time, faster shutter speed, games, ...
- Provides Linux-hosted cross-compilation system, using arm-elf-gcc 3.4.6

Watermarking Firmware Architecture





Firmware in Action

- On-screen menu allows to set watermark embedding strength and watermark key

Implementation Details

3.5 MB firmware image

- 150 KB CHDK add-on firmware
- watermarking code consumes approx. 3 KB
- 1 MB usable free memory, approx. $\sim$ 45 MB/sec memory bandwidth

Raw image buffer is 10 bit/pixel (packed): time-consuming to access individual pixels

- Require optimized, pipelined implementation to process 9 MB raw sensor data
- ARM instruction set provides cheap bit shift operations

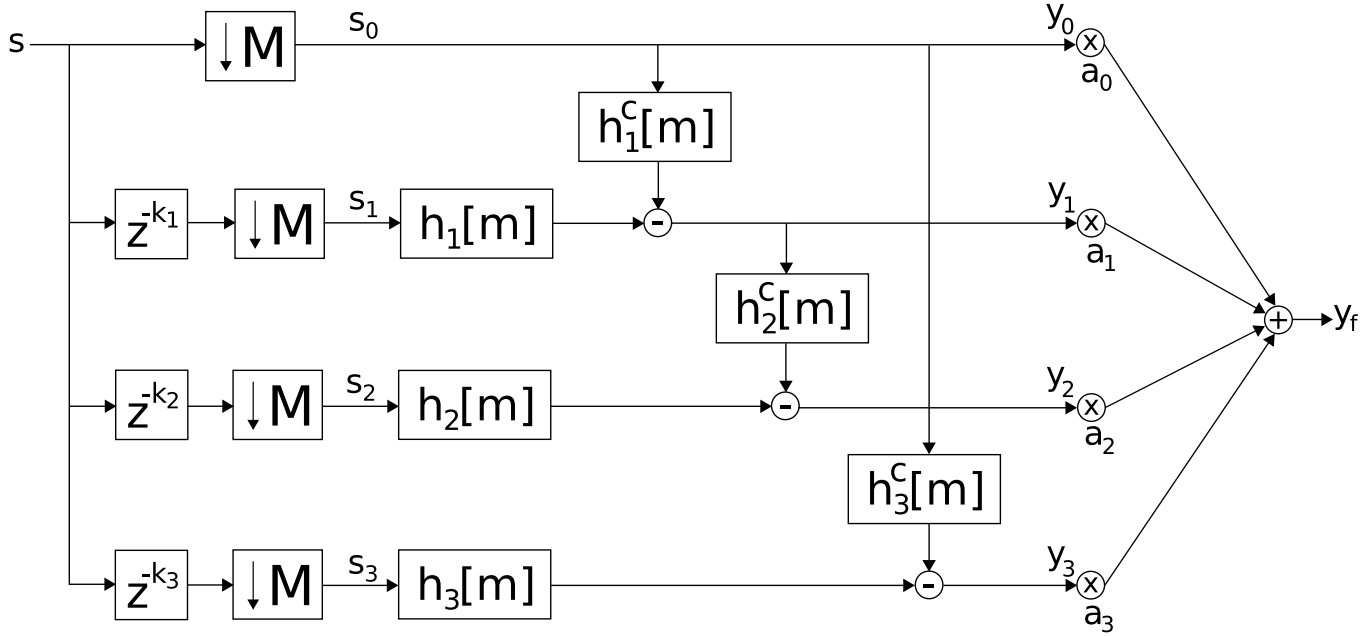
Watermark Detection

Adopt watermark detection strategy for interpolated, noisy images [Giannoula et al., 2006]

- Watermark is embedded in low-resolution raw data (blue channel): $x_w[m] = x[m] + \alpha w[m]$
- Watermark detection in upsampled, demosaicked image
- Exploit watermark information spread due to interpolation

Fusion of Polyphase Components

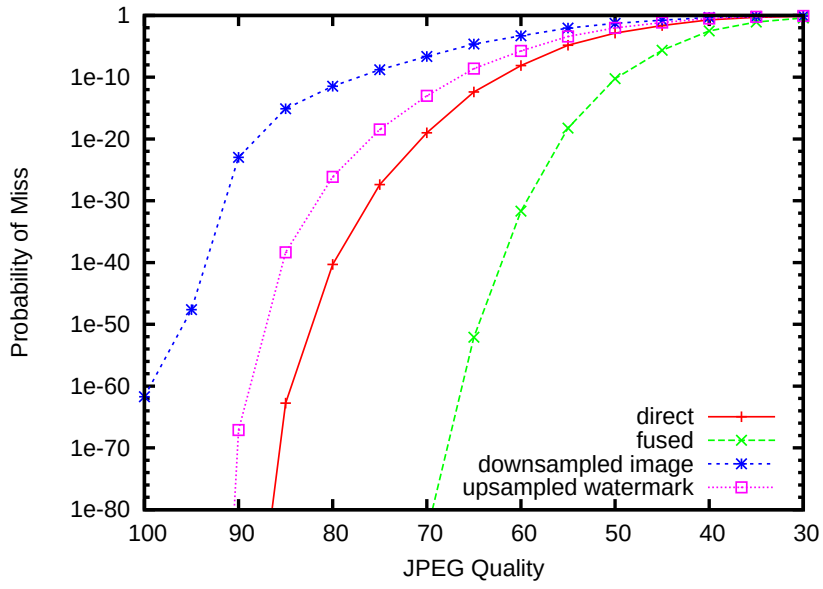
- Split demosaicked image into polyphase components
- Compute noise estimates of x_w using interpolation filters h_i and interference cancellation filters h_i^c
- Fuse components according to weight factors α_i depending on the estimated noise variance of the components:
  $y_f[m] = \sum_{i=1}^{3} \alpha_i \cdot y_i[m]$

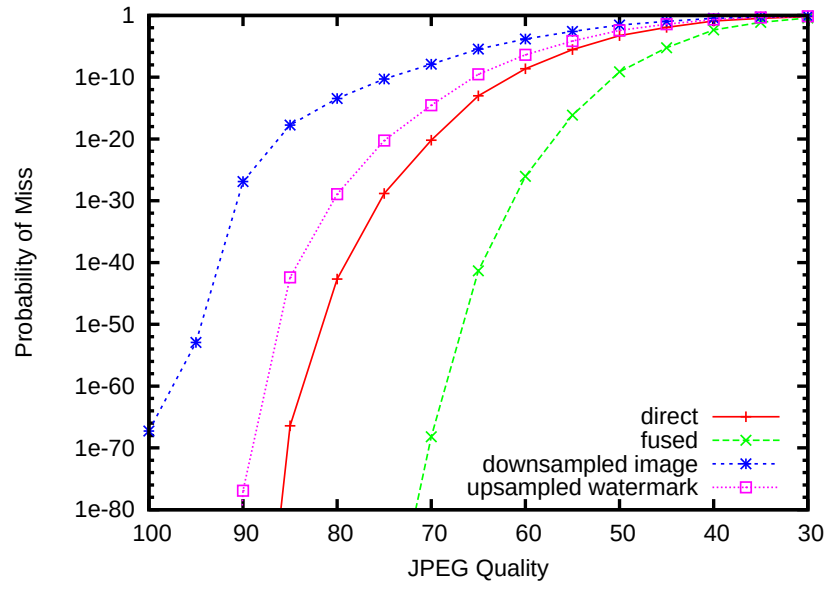


Experimental Results

- Implemented on Canon IXUS 70 (7.1 MP) and Canon Powershot A720 (8 MP)
- Embedding in blue color channel only; < 1 sec for embedding (software only)
- Test four detection methods (Probability of Miss for False-Alarm Rate of 10^{-6})
- Test three different demosaicking methods
- Test different camera settings (resolution, compression)

Watermark Detection after AHD Demosaicking and JPEG





- Proposed detector (fusion of polyphase components) outperforms other detection strategies

Impact of Demosaicking on Watermark Detection

Image	AHD		VNG		PPG	
	Direct	Fused	Direct	Fused	Direct	Fused
#1	$9.9 \cdot 10^{-20}$	$1.8 \cdot 10^{-84}$	$6.2 \cdot 10^{-43}$	$3.4 \cdot 10^{-294}$	$6.1 \cdot 10^{-12}$	$2.3 \cdot 10^{-29}$
#2	$9.9 \cdot 10^{-08}$	$1.1 \cdot 10^{-35}$	$1.2 \cdot 10^{-21}$	$3.3 \cdot 10^{-160}$	$7.5 \cdot 10^{-6}$	$1.2 \cdot 10^{-13}$
#3	$4.0 \cdot 10^{-10}$	$3.2 \cdot 10^{-38}$	$2.2 \cdot 10^{-23}$	$2.6 \cdot 10^{-145}$	$9.2 \cdot 10^{-7}$	$7.9 \cdot 10^{-16}$
#4	$5.3 \cdot 10^{-15}$	$5.7 \cdot 10^{-80}$	$6.5 \cdot 10^{-42}$	0.0	$6.1 \cdot 10^{-9}$	$4.8 \cdot 10^{-19}$
#5	$6.9 \cdot 10^{-5}$	$1.2 \cdot 10^{-15}$	$5.8 \cdot 10^{-19}$	$1.9 \cdot 10^{-116}$	$4.8 \cdot 10^{-4}$	$5.1 \cdot 10^{-7}$
#6	$7.3 \cdot 10^{-6}$	$3.6 \cdot 10^{-18}$	$2.5 \cdot 10^{-16}$	$2.3 \cdot 10^{-102}$	$2.1 \cdot 10^{-4}$	$3.2 \cdot 10^{-9}$
#7	$6.6 \cdot 10^{-21}$	$6.4 \cdot 10^{-69}$	$3.0 \cdot 10^{-53}$	$3.9 \cdot 10^{-289}$	$1.0 \cdot 10^{-14}$	$3.3 \cdot 10^{-29}$
#8	$1.5 \cdot 10^{-3}$	$8.5 \cdot 10^{-15}$	$8.9 \cdot 10^{-10}$	$5.7 \cdot 10^{-69}$	$1.3 \cdot 10^{-2}$	$1.3 \cdot 10^{-6}$
#9	$2.5 \cdot 10^{-4}$	$5.3 \cdot 10^{-11}$	$8.5 \cdot 10^{-15}$	$9.4 \cdot 10^{-77}$	$4.5 \cdot 10^{-3}$	$1.8 \cdot 10^{-4}$

- Proposed detector shows best performance for all demosaicking methods tested

Impact of Camera Settings on Watermark Detection

Resolution	Quality	Direct	Fused	Downsampled Image	Upsampled Watermark
3072×2304	SuperFine	$2.4 \cdot 10^{-161}$	0.0	$2.4 \cdot 10^{-15}$	$2.5 \cdot 10^{-100}$
3072×2304	Fine	$3.0 \cdot 10^{-125}$	0.0	$2.2 \cdot 10^{-15}$	$2.8 \cdot 10^{-83}$
3072×2304	Normal	$5.1 \cdot 10^{-88}$	0.0	$1.2 \cdot 10^{-14}$	$1.9 \cdot 10^{-63}$
2592×1944	SuperFine	$4.0 \cdot 10^{-68}$	0.0	$3.4 \cdot 10^{-14}$	$1.1 \cdot 10^{-50}$
2048×1536	SuperFine	$3.3 \cdot 10^{-60}$	$4.4 \cdot 10^{-223}$	$1.7 \cdot 10^{-16}$	$4.5 \cdot 10^{-46}$
1600×1200	SuperFine	$2.4 \cdot 10^{-38}$	$2.9 \cdot 10^{-117}$	$1.2 \cdot 10^{-8}$	$6.8 \cdot 10^{-29}$

- Watermark can be reliably detected for all camera settings (resolution, JPEG quality) tested

Conclusion

- Raw image data is probably the most valuable image asset; little prior work on watermarking of raw image data and and impact of a camera's image processing pipeline
- Presented real-time, software-only implementation of additive, spread-spectrum watermarking in digital camera firmware
- Exploited weighted fusion of image components to improve watermark detection performance of demosaicked images
- Evaluated the impact of demosaicking on watermark detection
- Further work to address perceptual shaping of watermark and embedding in all color channels

Questions?

- Source code of watermarking firmware add-on and watermark detector upon request: `http://www.wavelab.at/sources`
- Open-source implementation of three demosaicking methods: dcraw `http://www.cybercom.net/~dcoffin/dcraw/`

Bibliography

Blythe, P. and Fridrich, J. (2004).

Secure digital camera.
In *Digital Forensic Research Workshop*, Baltimore, MD, USA.

Chen, M., Fridrich, J., Goljan, M., and Lukas, J. (2008).

Determining image origin and integrity using sensor noise.
IEEE Transactions on Information Security and Forensics, 3(1):74–90.

Giannoula, A., Boulgouris, N. V., Hatzinakos, D., and Plataniotis, K. N. (2006).

Watermark detection for noisy interpolated images.
IEEE Transactions on Circuits and Systems, 53(5):359–363.

Lukac, R. and Plataniotis, K. K. (2006).

Secure single-sensor digital camera.
Electronic Letters, 42(11).

Mohanty, S. P., Kougianos, E., and Ranganathan, N. (2007).

VLSI architecture and chip for combined invisible robust and fragile watermarking.
IET Computers & Digital Techniques, 1(5):600–611.

Nelson, G. R., Julien, G. A., and Yadid-Pecht, O. (2005).

CMOS image sensor with watermarking capabilities.
In *Proceedings of the IEEE International Symposium on Circuits and Systems, ISCAS '05*, volume 5, pages 5326–5329. IEEE.

Popescu, A. C. and Farid, H. (2005).

Exposing digital forgeries in color filter array interpolated images.
IEEE Transactions on Signal Processing, 53(10):3948–3959.