https://doi.org/10.2352/EI.2026.38.16.AVM-114

BrilliantISP: An Enhanced HDR Image Signal Processing Pipeline

Brian Deegan, University of Galway, Galway, Ireland

Abstract

This paper presents brilliantISP, a modular, open-source HDR image signal processing pipeline for automotive camera applications. Unlike existing open-source ISPs, which employ floating-point arithmetic and are not optimized for HDR automotive use cases, brilliantISP adopts a predominantly fixed-point, unsigned integer architecture with explicit, bounded bit depths at each processing stage, mirroring the constraints of production embedded ISPs while remaining accessible for research and experimentation. The pipeline incorporates a configurable decompanding stage that reconstructs a linear-domain signal from piecewise-companded sensor outputs, supporting sensors with effective dynamic ranges up to 144 dB. Multiple global tone mapping operators are provided, including Reinhard, ACES, and Hable, alongside a Durand-style local tone mapping operator that decomposes the image into base and detail layers for contrast-preserving dynamic range compression. Additional pipeline stages include defect pixel correction, black level correction, lens shading correction, auto white balance, a choice of six demosaicing algorithms, local contrast and edge enhancement, and gamma correction. All stages are configurable via YAML parameter files, and comprehensive debug logging provides block-level execution statistics, dynamic range metrics, bit depth utilization, and histogram outputs to support both algorithm development and ISP tuning studies. The pipeline is validated on imagery from a Sony IMX623 split-pixel HDR fisheye sensor, where decompanded input spans approximately 19.26 EV at 20.7-bit effective depth, compressed to a 3.01 EV, 7.9-bit output after tone mapping and gamma correction. BrilliantISP is intended as a practical research platform for studying HDR tone mapping, demosaicing, and ISP tuning in the context of automotive computational photography.

Introduction

Automotive camera systems increasingly operate under challenging illumination conditions, ranging from bright sunlight and deep shadows to complex artificial lighting and LED-based flicker sources, often requiring effective dynamic ranges on the order of 120–140 dB [1]. While HDR image sensors and dedicated ISP hardware have matured commercially, there is a notable gap in accessible, research-oriented ISP implementations that reflect realistic automotive HDR pipelines and enable systematic analysis and tuning [2, 3].

Existing research has shown that ISP tuning can significantly affect both perceived image quality and the performance of downstream computer vision algorithms, with demosaicing and tone compression stages having a particularly significant impact [2, 4]. However, most deployments are closed, hardware-bound production ISPs, whereas many research ISPs adopt floating-point, high-precision designs that differ substantially from real-world embedded implementations [2]. Moreover, HDR-specific issues such as decompanding, multi-exposure fusion, and tone mapping for both human display and machine perception are still under-explored in public datasets and open-source pipelines [2–4].

To address this gap, we present brilliantISP, a modular, open-source HDR ISP pipeline developed with a design philosophy that mirrors production automotive systems while preserving the flexibility and observability demanded by academic and industrial research [5]. The contributions of this work are: (i) a practical, integer-based HDR pipeline architecture, (ii) a concrete implementation and reference configuration for a 144 dB dynamic range automotive sensor, and (iii) a set of debug and diagnostic tools that enable detailed analysis of dynamic range compression, bit depth usage, and block-level behavior [1].

Related Work

Automotive HDR Imaging and ISPs

Modern automotive image sensors employ a range of HDR strategies, including multiple exposure readout, split-pixel structures, and logarithmic or piecewise-linear analog-to-digital conversion to cope with extremely large scene dynamic ranges [1,2]. For example, the Sony IMX623 is a stacked back-illuminated split-pixel sensor with LED flicker mitigation and HDR support, targeted at front-facing automotive applications where both low-light performance and highlight retention are essential [1].

ISPs in this domain must handle wide field-of-view optics, low-light noise, lens shading, and tone mapping in addition to standard functions such as black level correction, white balance, color correction, and gamma encoding [4]. Automotive use cases further complicate ISP tuning because the processed imagery may be used simultaneously for driver display and algorithm consumption, often with differing optimal tone curves and noise characteristics [2]. Despite this complexity, HDR-specific ISP architectures and tuning strategies remain comparatively under-published, and there is a scarcity of publicly available raw HDR datasets for benchmarking [2].

Open-Source ISP Pipelines

Several open-source ISP projects have emerged that provide useful foundations for research and education. OpenISP is a Python-based ISP that implements a set of standard pipeline blocks for Bayer and RGB processing using floating-point arithmetic [6]. Fast-openISP builds

Characteristics of runtime, research and brilliantISP

Aspect	Production	Research	brilliantISP
Speed	Real-time, fixed	Slow, variable	Less slow, variable
Arithmetic	Fixed-point	Floating-point	Fixed-point (mostly)
Data format	Unsigned int	Floating-point	Unsigned int
Pipeline depth	24 → 16 → 8 bit	32/64 bit	32 → 16 → 8 bit
Flexibility	Fixed	Easily extended	Easily extended
Analysis	Very difficult	Easy	Easy

on this work with substantial speed optimizations (on the order of 300×) and additional processing blocks to make it more practical for experimentation [7]. Infinite-ISP extends the concept further with more ISP blocks, tuning and automation tools, and an FPGA port, moving closer to deployment-grade capabilities [8].

While these projects greatly advance transparency and reproducibility, they are not primarily focused on HDR automotive imaging, and their arithmetic and bit depth handling often diverge from the fixed-point constraints seen in production ISPs [2]. In particular, support for decompanding from companded sensor outputs, automotive-grade lens shading correction for fisheye optics, and HDR tone mapping tuned for 120–140 dB scenes is limited [2]. BrilliantISP is designed to fill this niche by explicitly targeting HDR automotive pipelines with a predominantly integer, bounded-bit-depth architecture [5].

Design Philosophy of brilliantISP

BrilliantISP is positioned between traditional production ISPs and flexible research ISPs in terms of implementation language, arithmetic, and analysis capabilities. Table 1 summarizes the key differences among a typical production ISP, a typical research ISP, and brilliantISP.

The central design principle is to employ mostly fixed-point, unsigned integer arithmetic with explicit, bounded bit depths at each stage, mimicking the constraints of embedded ISPs while retaining the convenience of a software implementation [5]. The pipeline generally progresses from up to 32-bit intermediate representations in the Bayer/raw domain, through 16-bit internal RGB/YUV representations, down to an 8-bit output suitable for display and compression.

Unlike production ISPs, which are typically opaque and tightly coupled to hardware, brilliantISP is implemented as a modular software pipeline that can be easily modified, extended, and instrumented [5]. Each block is configurable through a YAML-based parameter file, and detailed logging, histograms, and intermediate outputs can be enabled to facilitate both algorithm development and tuning studies. This makes brilliantISP an ideal platform for experiments on ISP tuning for enhanced computer vision performance, building on prior work demonstrating the importance of demosaicing and gamma compression stages [2, 4].

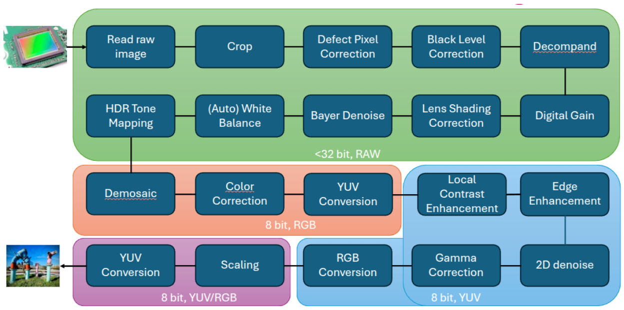

Figure 1. brilliantISP image processing pipeline

Pipeline Architecture

Processing Stages

The brilliantISP pipeline consists of a sequence of processing stages that transform a raw HDR Bayer image into an 8-bit RGB or YUV output, with optional branches for additional processing or algorithm-friendly outputs [5]. A high-level overview of the pipeline is as follows:

- Input and Decompanding: Read the raw image, crop to the region of interest, and decompand the sensor output from a compressed representation to a linear domain.
- Low-Level Bayer Processing: Defect pixel correction, black level correction, digital gain, lens shading correction, and Bayer-domain denoising.
- Color and Tone Processing: Auto white balance (AWB), HDR tone mapping, demosaicing, color correction, and YUV conversion.
- Local Enhancements: Local contrast enhancement, edge enhancement, optional 2D denoising, and gamma correction.
- Output Formatting: RGB conversion, optional scaling, and final YUV or RGB output at 8 bit per channel.

The pipeline selectively operates at different internal bit depths: raw Bayer data are represented in up to 32 bits, intermediate YUV/RGB stages in 16 bits, and the final output in 8 bits. The HDR tone mapping is applied before most local enhancement stages to compress the dynamic range to a manageable level while preserving detail in both shadows and highlights [2].

Decompanding and HDR Handling

Many automotive HDR sensors, including split-pixel designs, deliver pixel values in a companded format to reduce on-sensor bit widths and data bandwidth [1, 2]. BrilliantISP includes a configurable decompanding stage that reconstructs a higher-resolution linear domain signal from the companded input using piecewise-defined pin/pout mappings specified in the configuration file. An illustration of a companding/decompanding curve pair is shown in Figure 2. For a representative scene captured with the

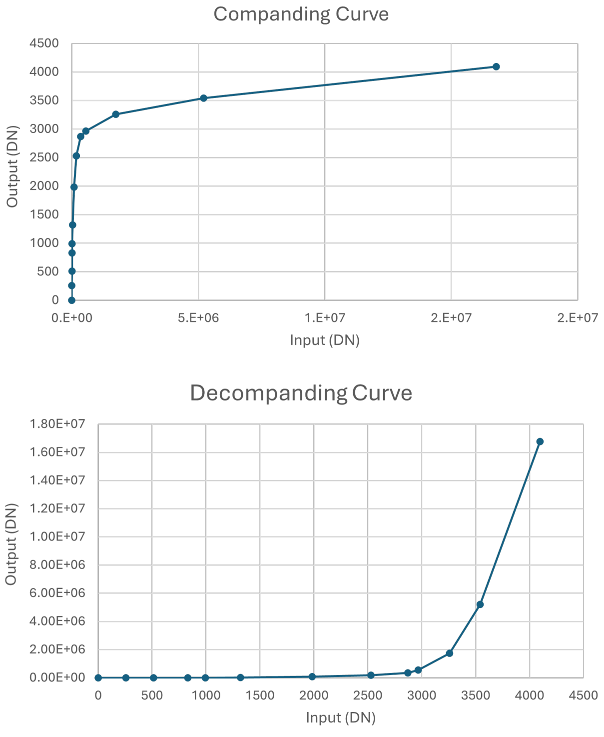

Figure 2. Plots of companding and decompanding curves for HDR image data bit depth management

IMX623 sensor, the input after decompanding exhibits a dynamic range of approximately 19.26 EV, with a minimum value of 1, a maximum of around 1.7 million digital numbers (DN), and effective utilization of about 20.7 bits.

This decompanded signal is then fed into the rest of the pipeline, where subsequent blocks must manage both the wide dynamic range and the quantization structure inherited from decompanding [2]. By instrumenting this stage with histograms and percentile statistics, brilliantISP allows researchers to quantify how much of the sensor's encoded dynamic range is effectively utilized in real scenes and how much is preserved downstream [2].

Lens Shading Correction and Demosaicing

Automotive fisheye lenses with fields of view on the order of 190 degrees introduce significant vignetting and color shading, which must be corrected to ensure uniform exposure and color across the image [4]. BrilliantISP implements a lens shading correction block that compensates for these effects using gain maps calibrated to the specific lens-sensor combination, as illustrated in Figure 3, showing images with and without correction.

Following lens shading and Bayer denoising, the pipeline offers multiple demosaicing algorithms, including bilinear interpolation, Malvar [9], variable number of gradients (VNG) [10], Hamilton–Adams [11], pattern-perturbation gradient (PPG) [12], and linear minimum mean square error (LMMSE) methods [13]. Figure 4 illustrates the differences in resolution and aliasing artifacts across the algorithms included. The inclusion of several demosaicing options enables comparative studies of their impact on both perceived image quality and downstream vision tasks.

Figure 3. Illustration of lens shading correction for wide FoV cameras

HDR Tone Mapping and Local Enhancements

Global Tone Mapping Operators

To the best of the author's knowledge, brilliantISP is the first open-source ISP with HDR tone mapping capability. HDR tone mapping compresses the wide dynamic range of the input scene into a lower dynamic range suitable for 8-bit display and compression, while aiming to preserve perceptually important contrast [2]. Examples of global tone mapping operators implemented in brilliantISP include Reinhard [14], ACES [15], and Hable [16].

Local Operators and Contrast Management

Beyond global tone mapping, brilliantISP includes a Durand-style local tone mapping operator, which decomposes the image into base and detail layers and applies a contrast gain to the base layer [17]. Demonstrations with contrast gains of 0.5 and 2.0 highlight how this operator can either compress or emphasize local contrast while still respecting the overall dynamic range constraints imposed by the global tone mapping.

Local contrast enhancement and edge enhancement blocks are applied in the tone-mapped domain to refine perceived sharpness and detail. These blocks can be toggled and configured through the YAML file, allowing researchers to measure their impact on metrics such as edge acuity, noise amplification, and algorithm performance [4]. Optional 2D denoising can be introduced at this stage to balance detail recovery against noise suppression in low-light regions.

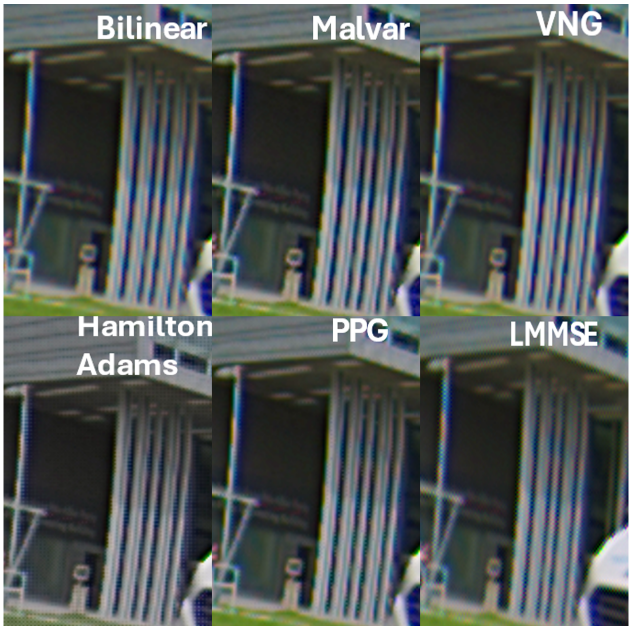

Figure 4. Comparison of demosaic algorithms

Gamma Correction and Output Formatting

The final steps in the pipeline include gamma correction and color space conversions to yield an 8-bit RGB or YUV output suitable for display or encoding. BrilliantISP supports YUV 4:4:4 formatting and standard RGB conversion, with explicit logging of configuration choices such as whether RGB output is enabled and which YUV format is selected. For the IMX623 test case, the output dynamic range after tone mapping and gamma correction is approximately 3.01 EV, with values spanning from 0 to about 238 DN and effective utilization of roughly 7.9 bits.

Histogram comparisons between input (after decompanding) and tone-mapped output in both linear and logarithmic scales provide insight into how much contrast is preserved in different luminance regions. This is particularly relevant in automotive scenes, where both very bright sky regions and dark road surfaces must remain interpretable by both humans and computer vision algorithms [2, 4].

Debug, Logging, and Configuration Features

brilliantISP includes debug and diagnostic capabilities, which are often unavailable in closed production ISPs. Configuration is handled through YAML files that define platform parameters, sensor information, companding curves, and per-block enable flags and parameters. For example, the configuration can specify whether progress bars are shown, whether 3A rendering is enabled, the output directory and file naming, and whether histograms should be generated in linear or logarithmic scales.

Runtime logging provides detailed, block-level information including execution times, enable/disable status for each stage, and summary statistics of input and output distributions. For a representative IMX623 frame, log entries include processing steps such as decompanding, lens shading correction, Bayer denoising, HDR tone mapping, sharpening, and RGB/YUV conversion, along with messages indicating that noise reduction or scaling were disabled in the particular run. Dynamic range statistics, percentiles, and bit depth utilization are reported for both input and output, and histogram comparison images are automatically saved to the output directory.

These diagnostic tools are designed to support both classical ISP tuning workflows and emerging approaches, where ISP parameters are optimized directly with respect to computer vision metrics [2, 4]. By exposing internal states and enabling selective activation of pipeline blocks, brilliantISP can serve as a testbed for such research.

Case Study: IMX623 HDR Fisheye Sensor

Sensor and Optics

To demonstrate the capabilities of brilliantISP in an automotive context, we consider a test image acquired with a Sony IMX623 sensor paired with a 190-degree automotive fisheye lens [1]. The IMX623 is a 1/2.42-inch, 2.95 MP stacked back-illuminated CMOS sensor featuring split-pixel HDR and LED flicker mitigation, designed specifically for automotive applications. The sensor's effective dynamic range can reach approximately 144 dB, enabling the capture of scenes with very high contrast between the sky, road, and illuminated objects.

In the test scenario, the scene dynamic range is estimated to be around 120 dB, with content ranging from bright sky to shadowed road surfaces and vehicle interiors. The sensor uses an RGGB color filter array and delivers a resolution of 1920×1536 pixels, with HDR companded output mapped to a 24-bit input pipeline in brilliantISP. Lens shading effects are pronounced due to the fisheye optics, necessitating robust lens shading correction to avoid underexposed peripheral regions [4].

Dynamic Range Compression and Bit Depth Utilization

For the IMX623 test frame, brilliantISP's logging indicates that the input (after decompanding) exhibits a dynamic range of approximately 19.26 EV, with minimum, maximum, and percentile values of (1, 1 736 000, 1/626 059) and an effective bit depth utilization of 20.7 bits. After processing through tone mapping and subsequent stages, the output dynamic range is reduced to about 3.01 EV, with output values ranging from 0 to 238 and percentiles (29, 238), corresponding to a bit depth utilization of approximately 7.9 bits. Figure 5 shows the impact of modifying the Durand tone mapping operator contrast gain on output image brightness and contrast.

Histogram plots produced by the pipeline highlight how the decompanded input distribution spans several orders of magnitude, while the tone-mapped and gamma-corrected output is concentrated in a narrower, display-friendly range (see Figure 6). This illustrates the trade-off between preserving fine-grained intensity resolution across the entire dynamic range and concentrating contrast in perceptually or algorithmically relevant regions [2, 4]. By adjusting tone mapping parameters and local enhancement

Figure 5. Durand tone mapping algorithm with 0.5 and 2.0 contrast gain

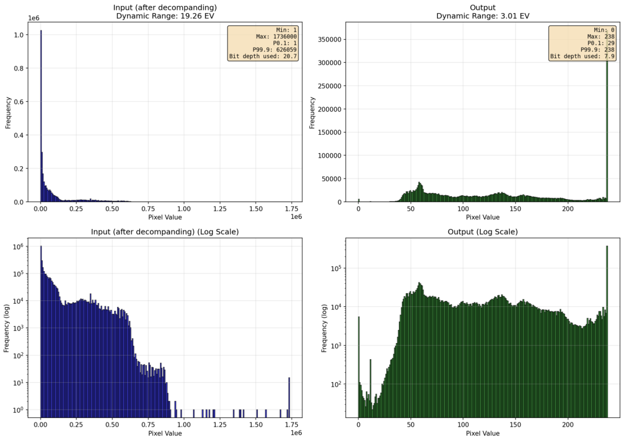

Figure 6. brilliantISP debug histograms illustrate the effect of tone mapping on image brightness, contrast, and bit depth utilization

gains, researchers can explicitly explore these trade-offs within brilliantISP.

Discussion and Future Work

BrilliantISP demonstrates that it is feasible to construct an open-source ISP that retains key characteristics of production automotive pipelines, including fixed-point arithmetic, explicit bit depth management, HDR decompanding and tone mapping, and lens shading correction, while remaining flexible and analyzable for research. Compared with previous open-source ISPs, brilliantISP uniquely emphasizes HDR automotive use cases, decompanding, and detailed dynamic range diagnostics [2, 6–8].

Future work on brilliantISP includes extending the pipeline to support multi-image HDR combination, integrating learning-based ISP blocks, refining the parameter structure for more systematic tuning, and adding further debug tools. A C++ implementation for batch image conversion and a GPU-accelerated version for integration into machine learning pipelines are also planned, for both real-time use and large-scale dataset training. Additionally, coupling brilliantISP with publicly available HDR driving datasets and standardized metrics would help advance the currently under-researched area of HDR ISP processing for automotive vision.

Conclusion

In this work, we introduce brilliantISP, an open-source HDR ISP pipeline designed for automotive camera applications. brilliantISP combines a realistic fixed-point architecture, multiple global and local HDR processing blocks, and extensive debug and diagnostic features, and provides a practical platform for exploring ISP tuning, HDR tone mapping, and their impact on both human-viewed imagery and automated perception systems.

References

[1] Sony Corporation. Sony imx623 automotive image sensor technology overview, 2023. Technical Whitepaper.

[2] Lucie Yahiaoui, Jonathan Horgan, Brian Deegan, Senthil Yogamani, Ciarán Hughes, and Patrick Denny. Overview and empirical analysis of ISP parameter tuning for visual perception in autonomous driving. Journal of Imaging, 5(10), 2019.

[3] Dara Molloy, Brian Deegan, Darragh Mullins, Enda Ward, Jonathan Horgan, Ciaran Eising, Patrick Denny, Edward Jones, and Martin Glavin. Impact of ISP tuning on object detection. Journal of Imaging, 9(12), 2023.

[4] Diarmaid Geever, Tim Brophy, Dara Molloy, Martin Glavin, Edward Jones, and Brian Deegan. ISP tuning for improved image quality in machine vision. Electronic Imaging, 36(9):257–1–257–1, 2024.

[5] Brian Deegan. BrilliantISP: A work-in-progress HDR image signal processor. `https://github.com/briandeegan82/brilliantISP`, 2026. Accessed: 2026-03-21.

[6] Crux. openISP: Open image signal processor. `https://github.com/cruxopen/openISP`. Accessed: 2026-03-21.

[7] Jueqin Qiu. fast-openISP: A faster re-implementation of openISP. `https://github.com/QiuJueqin/fast-openISP`, 2021. Accessed: 2026-03-21.

[8] 10x Engineers. Infinite-ISP: A full-stack ISP development platform. `https://github.com/10x-Engineers/Infinite-ISP`, 2023. Version 1.1. Accessed: 2026-03-21.

[9] Henrique S. Malvar, Li-wei He, and Ross Cutler. High-quality linear interpolation for demosaicing of Bayer-patterned color images. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), volume 3, pages III–485–III–488, 2004.

[10] Edward Chang, Shiufun Cheung, and Davis Y. Pan. Color filter array recovery using a threshold-based variable number of gradients. In Proceedings of SPIE, Sensors, Cameras, and Applications for Digital Photography, volume 3650, pages 36–43, 1999.

[11] John F. Hamilton and James E. Adams. Adaptive color plane interpolation in single sensor color electronic camera. U.S. Patent 5,629,734, May 1997.

[12] Dave Coffin. Decoding raw digital photos in Linux (dcraw). Open-source software, release 8.71. Available: `https://www.dechifro.org/dcraw/`, 2005. Patterned Pixel Grouping (PPG) demosaicing algorithm introduced in this release.

[13] Lei Zhang and Xiaolin Wu. Color demosaicking via directional linear minimum mean square-error estimation. IEEE Transactions on Image Processing, 14(12):2167–2178, December 2005.

[14] Erik Reinhard. Parameter estimation for photographic tone reproduction. Journal of Graphics Tools, 7(1):45–51, 2002.

[15] The Academy of Motion Picture Arts and Sciences, Science and Technology Council, and Academy Color Encoding System (ACES) Project Subcommittee. Technical Bulletin TB-2014-004: Informative Notes on SMPTE ST 2065-1 – Academy Color Encoding Specification (ACES). Technical Bulletin TB-2014-004, Academy of Motion Picture Arts and Sciences, 2014.

[16] John Hable. Uncharted 2: HDR lighting. Presentation at the Game Developers Conference (GDC), March 2010. Naughty Dog. Session held 13 March 2010, Room 305, South Hall, San Francisco, CA. `https://www.gdcvault.com/play/1012351/Uncharted-2-HDR`.

[17] Frédo Durand and Julie Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. In Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), pages 257–266, New York, NY, USA, 2002. ACM.

Author Biography

Brian Deegan received his bachelor’s degree in computer engineering and the M.Sc. degree in biomedical engineering from the University of Limerick, Limerick, Ireland, in 2004 and 2005, respectively, and the Ph.D. degree in biomedical engineering from the University of Galway, Galway, Ireland, in 2011. From 2011 to 2022, he was with Valeo Vision Systems as a Vision Research Engineer, focusing on Image Quality. In 2022, he joined the Department of Electrical & Electronic Engineering, the University of Galway as a Lecturer and Researcher. His research interests include the relationship between blood pressure and cerebral blood flow in humans, high dynamic range imaging, LED flicker, Topview harmonization algorithms, and the relationship between image quality and computer vision.


electronic
IMAGING


Image Engineering


STUDENT/YP SHOWCASE
PURDUE