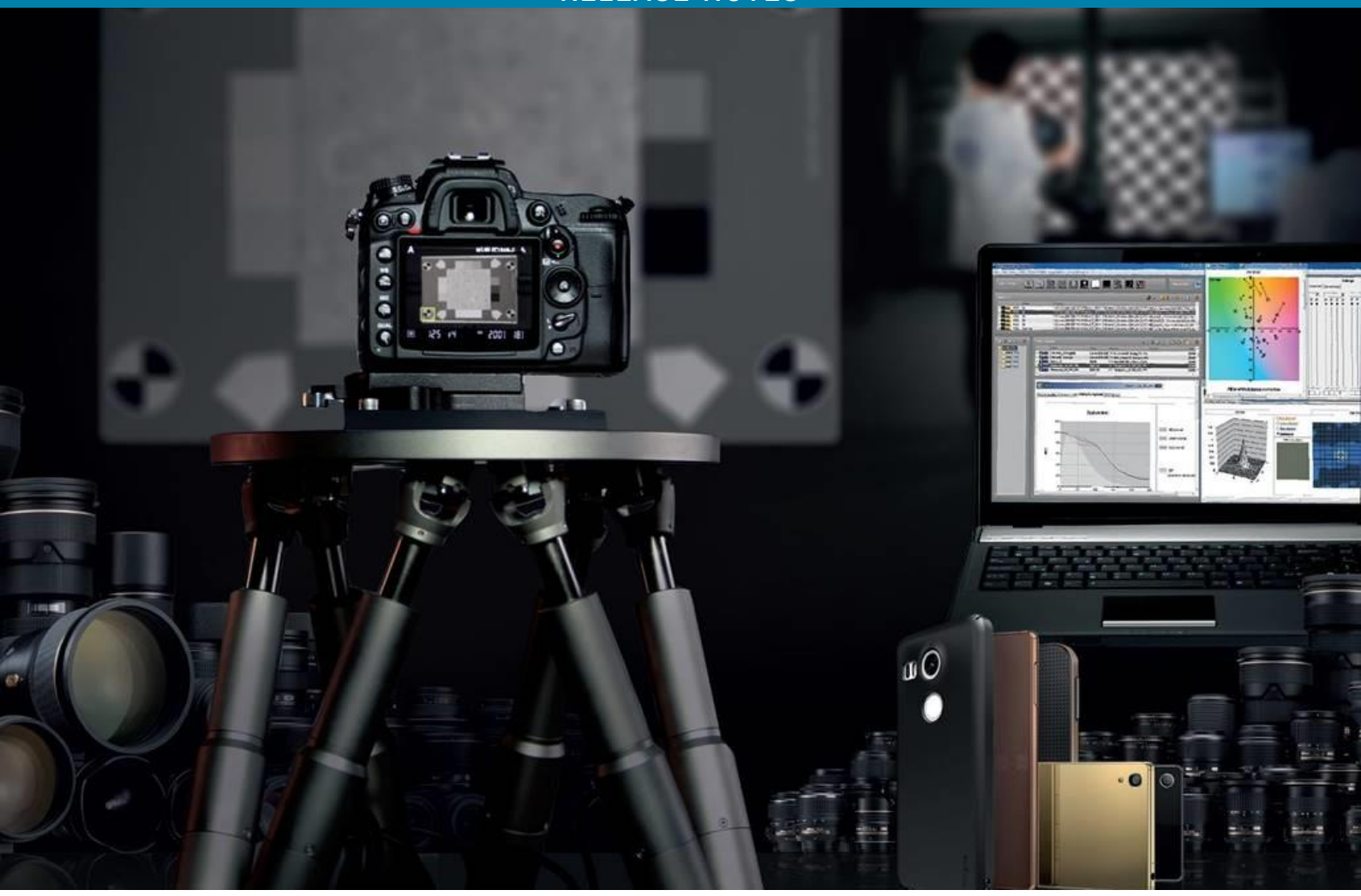
ANALYZER by DXOMARK

V9.0.1

RELEASE-NOTES



Analyzer solution includes everything you need to reliably measure and analyze the imaging performance of any type of image capture device: testing protocols and methodologies, laboratory specifications and installation guidelines, data management and analysis software.

Analyzer is the only solution that can fully measure not just the quality of captured images, but also artifacts from electronic shutters, the effectiveness of 6-axis image stabilization systems, the geometry of dual-module cameras for 3D or stereoscopic vision, and the dynamic response of imaging devices to changing scenes and light levels. Coupled with Analyzer's ability to measure images the way consumers see them, it is possible to fully evaluate cameras and lenses in a way that predicts how they will perform in real world situations.

Table of Contents

1 - Software Installation

1.1 Analyzer & Workflow Manager Installation

We recommend using the final user account to install the software.

Starting from Analyzer 8.3, the entire suite is available as an unified installer. In 9.0.1, the installer will install the Analyzer application suite, the Workflow Manager python libraries, the AZ Photo and AZ Video analysis applications and the LabManager lab equipment control service and user interface as well as all the necessary documentation and API references.

An installation manual is available alongside the installer, which describes in detail how to install these different components.

This manual also shows how to handle Workflow Manager installation without Internet connection, as well as some troubleshooting instructions, including how to verify if software components were successfully installed.

1.2 First launch and license

Once Analyzer has been installed, you can follow these steps:

1. Plug in the Analyzer USB dongle if you have one.
2. Launch Analyzer.
3. If a license is required, a license dialog will appear that will let you request a license code from Support team.
4. Copy-paste the code you receive into the [Set license…] dialog.

Contact support support.analyzer@dxomark.com if code registration fails or if you did not receive your license.

2 - Analyzer 9.0.1

2.1 New features

2.1.1 MLS support in CameraV5 protocols

Since Analyzer V9.0 release MLS devices are supported in some protocols. Following protocols are concerned:

- Photo
    - AFHDR
    - DMC
    - DMC Zoom
- Video
    - DMC
    - DMC Zoom
    - DMC Zoom Smoothness

2.1.1.1 Automatic detection

The scripts that run these protocols should automatically select your current lighting system based on heuristic, but sometimes this mechanism fails to behave correctly.
In this case, it is possible to force lighting system to use in a protocol you can add an optional setting to run the protocol.
This setting is called “use_mls”, and takes a Boolean (“True”, or “False”) value, and is passed to run() method through “options” parameter (see concerned protocols documentation for further details).

2.1.1.2 Scripts update

In order for automatic lighting system to work (in all protocols), Python scripts need to follow more closely Python correct development requirements. This translates in your scripts by having the code in a code block defined by “*If __name__ == “__main__”:*” statement.
You can see examples in the documentation of all concerned protocols.
Do not hesitate to contact the support (support.analyzer@dxomark.com) in case you have issues with this change.

2.2 Bug fixes

2.2.1 Protocol Automation

#6069: Fix aggregated outputs for Zoom protocols in Protocol Automation
#6018: waitTime argument was not taken into account for AFHDR automation

2.2.2 AZPhoto

#6033: User was unable to load any images without HDR16 license bit

3 - Analyzer 9.0.0

3.1 Breaking changes

3.1.1 FlareMeasure 

The IEEE-P2020 standard committee has decided to replace the “flare attenuation” metric with “flare intensity”. The flare measurement now includes flare intensity output keys. See the API documentation, and Flare user manual for more information.

Some output keys have also been **removed** from `FlareMeasure`:

- `IlluminanceMap`
- `MaximalIlluminance`
- `MeanIlluminance`
- `FlareAttenuationMap`
- `LargestMeasurableFlareAttenuationdB`
- `SmallestMeasurableFlareAttenuationdB`

`FlareMeasure.SaveAll()` and `FlareSensitivity.SaveAll()` functions have been renamed to `SaveResults()` in order to be more consistent with other measures, and the `"customName"` parameter has been removed.

The `"MinimalFlareAttenuationIndB"` output is now computed as minimal flare attenuation on the whole flare map, instead of percentile 1% in the previous version. Values may differ. Similarly, `"MaxFlareIntensityIndB"` is computed as maximum flare intensity on the whole flare map.

3.1.2 MtfCollimatorMeasure 

The input `"UserSpecifiedROIs"` of `MtfCollimatorMeasure` is renamed to `"PatchesPosition"` in order to be more consistent with other measures. Additionally, its structure has changed. For a complete description about its structure, see the inline help of `MtfCollimatorMeasure.Inputs`.

The output `"ROI"` is renamed to `"PatchesPosition"`, and now uses the same format as the input.

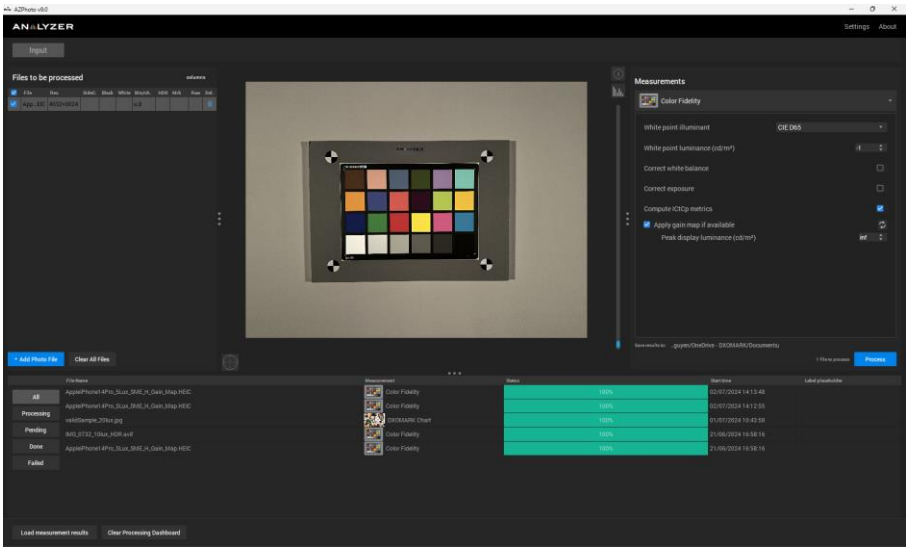
3.1.3 P2020NoiseMeasure 

The optional input key `"ForceSaturationToMaxSignal"` has been replaced by `"SaturationLevel"` in P2020 Noise measurement. This new input allows users to define a custom saturation level for the test images, it is particularly useful for the computation of saturation related metrics in the case that cameras have non constant saturation.

3.2 New features

3.2.1 New AZ Photo application

AZ Photo is the new graphical user interface dedicated to photo measurements.



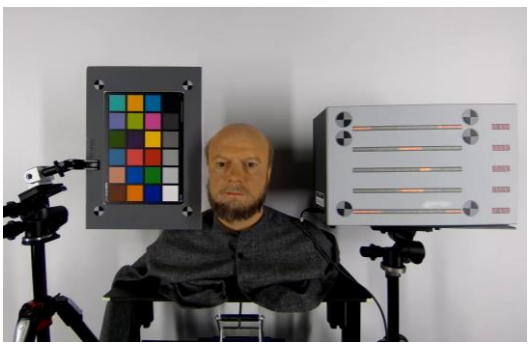
This new interface provides easy to use reports for the following photo measurements in Analyzer 9.0, with more to come in future releases:

- Color Fidelity, with HDR support (**`dxomark.core.measure.ColorMeasure`**)
- DXOMARK chart (**`dxomark.core.measure.DmcMeasure`**)

Note that the legacy graphical interface (“Analyzer”) is still available for the transition period; it is however considered deprecated and will be removed once all measurements are available in AZ Photo and AZ Video. It will also provide no access to new inputs, outputs or features that have been added recently (such as HDR support).

3.2.2 Single Person Video Conference Measure (SPVC)

This new measure is available in Workflow Manager, and is designed to perform video analysis of the following setup using one of the available realistic mannequins:

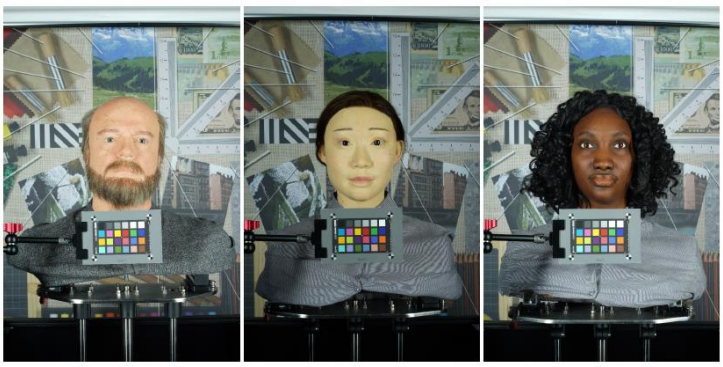


See **`dxomark.core.videomeasure.SinglePersonVideoConfMeasure`** for more information about this measure. In particular, the following outputs are available:

- Face exposure
- Face detail preservation
- Video Timing
- Video Color fidelity
- Video Exposure convergence
- Video Color stability

3.2.3 New model for Realistic Mannequin measure

The Realistic Mannequin (RM) Detail Preservation Metric is a learning-based method that assesses the level of perceived texture on a face in a controlled portrait setup environment. This measure is available in Workflow Manager in **`dxomark.core.measure.RealisticMannequinMeasure`**.



A new trained model has been added, with the following improvements from the previous model:

- Support for the dark skin tone realistic mannequin “Diana”.
- Wider coverage in image quality levels, with a much greater range of very low quality and very high quality devices in the annotated training set.
- New output for aligned quality metrics, allowing comparisons between all realistic mannequin models (“Eugene”, “Sienna” and “Diana”).

Users can manually select the model by setting the input key **`"RealisticMannequinVersion"`** to one of the options below.

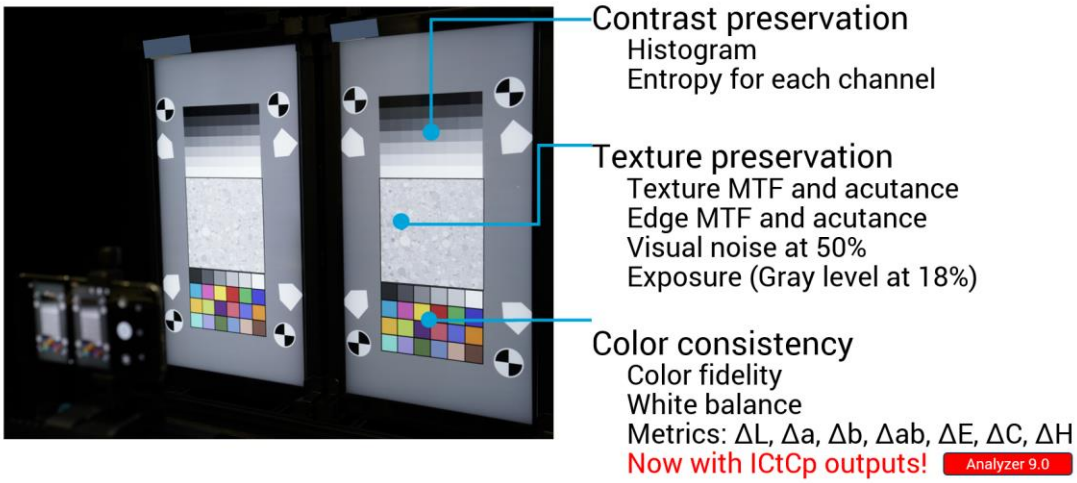
- **`RealisticMannequinModelVersion.RM_V1`**:
  Previous model supporting Eugene and Sienna, default value.
- **`RealisticMannequinModelVersion.RM_V2`**:
  New model supporting Eugene, Sienna and Diana.

The selected version of the mannequin will appear in the output dictionary as **`"RM_V1"`** or **`"RM_V2"`**.

The Analyzer manual and Workflow Manager API documentation have been updated to provide detailed information about the new model.

3.2.4 Composite HDR measure

The “HDR Composite” measure performs comparison between two images (reference and tested sample) of two composite charts (left and right) placed side by side, in terms of color differences, sharpness, contrast and noise.



Previously only available in the legacy “Analyzer” interface, this measure is now available in the Workflow Manager API at **`dxomark.core.measure.CompositeHDRMeasure`**.

This new API also provides access to color difference metrics compatible for HDR-encoded images in the ICtCp color space.

The Analyzer manual and Workflow Manager API documentation have been updated to provide detailed information about the use of this measure.

3.2.5 Flare

3.2.5.1 Flare Intensity

Flare Intensity metrics as defined in IEEE P2020 draft have been added to the flare measure outputs. See the flare user manual and API documentation for more information.

3.2.5.2 NIR Flare measure

So far, the flare measure quantifies the performance of a device in terms of flare using a visible light source. The measure has been extended to measure stray lights using near infrared (NIR) light sources. It is done using the class **`dxomark.core.measure.FlareMeasure`**.

3.2.5.3 Flare Aggregation

Flare aggregation (aggregation of the flare measurement for all angles of a camera) is now available using the class **`dxomark.core.measure.FlareAggregation`**. See the API documentation for more information.

3.2.5.4 New inputs and outputs in FlareMeasure and FlareAggregation

The **`FlareMeasure`** and **`FlareAggregation`** measurements have new inputs and outputs.

A new output has been added under the key **`"CdfFlareIntensityIndB"`**, giving the cumulative distribution function of flare intensity in dB.

A new input has been added under the key **`"FlareIntensityThresholdIndB"`**. When provided, a percentage of the image that is above this value will be output under the key **`"PercentageOfImageWithFlare"`**.

3.2.6 HDR formats and color management support  

Analyzer 9.0 has a lot of improvements related to HDR support!

3.2.6.1 Support for Apple iOS 18 HEIC gain map image format

Support for HEIC gain map images generated by Apple devices using iOS 18 (following standard draft ISO/CD 21496-1) is added.

This joins the existing support for Apple (pre-iOS 18) and UltraHDR (Android) still image formats that incorporate gain maps. Gain maps allow these devices to store both HDR and SDR renderings in the same file while retaining backwards compatibility with SDR displays.

3.2.6.2 Considering the peak display luminance when computing HDR gain maps

It is now possible to provide the target peak display luminance as input when HDR gain maps are applied. For example, this allows measurements as if the image was seen on a 1000 nits display, even if the fully-applied gain map image can go up up to 1800 nits.

In the case of UltraHDR images, the computation is performed according to the corresponding specification since such a possibility was described there. As for Apple images, the provided luminance is used for computing the headroom and the latter overrides the one specified in the MakerNote values described in the Apple documentation, and looks visually identical to the rendering of the Apple Photos software.

3.2.6.3 Processing 3-channel UltraHDR gain maps, with gamma not equal to 1.0

The single-channel limitation in UltraHDR gain maps has been removed. Also, if gamma is not equal to 1.0, such images are accepted, and the corresponding correction is applied. Such images can be generated by some third-party software like Adobe Photoshop.

3.2.6.1 Improved color space conversion tools

The class **`dxomark.corewrappers.ColorSpaceConverter`** is now the recommended way to perform color conversion between different color spaces (such as: Nonlinear RGB, Linear RGB, CIE-XYZ, CIE-L*a*b* and IC_TC_P). This new colorspace converter supports images loaded using **`dxomark.corewrappers.Image`** ; image value ranges and memory allocation are now managed automatically.

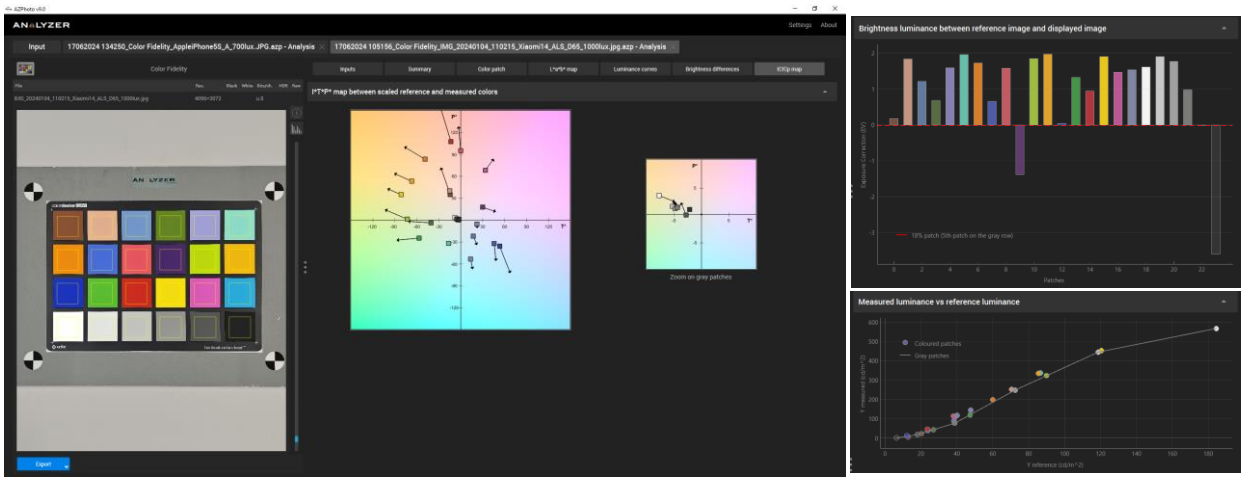
The conversion tools previously available in **`dxomark.core.measure.ColorSpaceConverter`** are fully replaced by this new class and are considered deprecated.

3.2.7 HDR metrics and measurement support

Analyzer 9.0 continues the work on supporting HDR-encoded formats in all measurements and analysis tools.

3.2.7.1 Color Fidelity reports for HDR-encoded images in AZ Photo

AZ Photo supports Color Fidelity reports for HDR images encoded using either one of the supported ISO 22028-5 compatible HDR formats (using ITU-T H.273 CICP metadata with the HLG or PQ transfer functions, such as AVIF/HEIF/PNG), or one of the supported gain map formats.



The generated reports provide visualizations of IC_TC_P and CIE-XYZ metrics, allowing the user to analyze and compare different HDR formats, or even between HDR and SDR formats.

3.2.7.2 ICtCp output available in more measures

Several measures that output CIE-L*a*b* related metrics have been modified to also output IC_TC_P related metrics when using input setting **`'ComputeHdrMetrics'`** as True.

The concerned measures are:

- AutofocusHDRMeasure
- ColorMeasure
- CompositeHDRMeasure
- DmcMeasure
- FaceExposureMeasure
- PortraitHDRMeasure
- PortraitTimingColorMeasure
- PortraitTimingColorSelfieMeasure
- SingleColorMeasure
- SingleImageHdrMeasure
- TextureMeasure
- VideoColorFidelityMeasure
- VideoColorStabilityMeasure
- VideoDmcMeasure
- VideoExposureConvergenceMeasure
- VideoTextureMeasure

3.2.7.3 DeltaCSITP and NESITP outputs for IC_TC_P computation

Several additional outputs, including ΔC_{SITP} and NE_{SITP}, are now available when computing IC_TC_P metrics (input setting **`'ComputeHdrMetrics'`** as True in supported measurements).

These metrics take into account the linear “exposure” correction of reference IC_TC_P value respect to the measured luminance value. More details are available in the Color Fidelity part of the User Manual.

3.2.7.4 New convergence thresholds for Video Exposure Convergence measure

In the Video Exposure Convergence Measure, IC_TC_P exposure convergence metrics are computed with adapted thresholds and not with the L*a*b* thresholds.

3.2.7.5 sRGB linearization in Photo and Video Vignetting measures

A new optional input **`"UseImageColorMetadata"`** has been added to **`VignettingMeasure`** and **`VideoVignettingMeasure`** in Workflow Manager.

This inputs directs the measure to perform sRGB linearization before performing the measurement, allowing comparison of results generated from files with different color encodings (e.g. one SDR image and one HDR image). By default, it is set to False.

3.2.8 Additional instrument control in LabManager

3.2.8.1 Hexapod interface

It is now possible to control a PI Hexapod (H-811, H-840 and H-860 models) using the Hexapod interface Python API in LabManager.

Features directly overlap everything that the Shaker software can already do, which includes:

- Adding, removing, listing wave motions (move motions are no longer supported as they are deprecated in the PI control API)
- Running / stopping / progress of loaded motions
- Moving to a given position
- Moving to neutral / shipping / rest predetermined positions
- Setting and getting the pivot point position

3.2.8.2 LED Universal Timer interface

It is now possible to control a LED Universal Timer mark I/II using the Ledbox interface Python API in LabManager.

Features directly overlap everything that the Timer Pilot software can already do, which includes:

- Setting a configuration in timing or flickering modes
- Retrieving a set configuration
- Setting led line period, intensity, duty cycle, phase values independently
- Setting 7-segment display intensity (not independent) and values
- Starting and stopping the monitoring of autofocus captures when used together with a Digital Trigger in autofocus triggering mode, a Digital Probe and an AMO

3.2.8.3 Digital Trigger interface

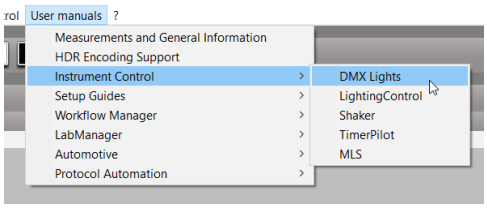
It is now possible to control a Digital Trigger using the Digital Trigger interface Python API in LabManager.

Features directly overlap everything that the Timer Pilot software can already do, which includes:

- Triggering a single shot on a device under test
- Starting and stopping the autofocus triggering mode

3.2.9 LitePanels Gemini user manual

A new user manual has been written for the LitePanels Gemini fixtures. It has been merged with the already existing Kino Flo user manual under the name DMX Lights, aggregating all resources about DMX light fixtures.



This user manual is available through the Analyzer GUI under the “User manuals” > “Instrument Control” > “DMX Lights” menu.

3.2.10 Jeti Specbos 2501 spectrophotometer support

On top of the Jeti spectraval 1501 and 1511 spectrophotometers, LabManager now officially supports the Jeti Specbos 2501 through the ***spectrophotometers_interface***. The same commands are all available for this new device, that features the same technical specifications as the Jeti Spectraval 1511 spectrophotometer.
WARNING: the serial number used to control a spectrophotometer through the ***spectrophotometers_interface*** has changed to match the serial number printed on the side of the device.

3.3 Modified features

3.3.1 Linearity calibration of the MLS speed up

The linearity calibration can now be performed in 3 hours 20 minutes using the JETI spectrophotometer (20 minutes using the CSS45 spectrophotometer) instead of 17 hours.

3.3.2 P2020DynamicRangeMeasure

A new optional input **“SaturationLevel”** has been added to P2020 Dynamic Range measurement. This input allows users to define a custom saturation level for the test images, it is particularly useful for the computation of dynamic range metrics in the case that cameras have non constant saturation.

3.4 Known limitations

3.4.1 White point value in ICtCp and ComputeHdrMetrics

The **"WhitePoint"** input has a different interpretation for ICtCp metrics than it has for CIELAB-based metrics. While the results are technically correct, they might be of difficult interpretation because of several color adaptations that occur.

When using the **"ComputeHdrMetrics"** option in any of the measures, it is recommended that the input **"WhitePoint"** is left to its default value (D65). Furthermore, when interested in chromatic differences in ICtCp it is recommended that the references be provided directly using this color space.

3.4.2 Measurement ROIs sometimes incorrectly positioned in AZ Video preview

In AZ Video, when viewing results of measurements performed using previous version of AZ Video (8.0 to 8.4), measurement ROI positions will be incorrectly displayed in the video preview if the videos have a rotation tag. The measurement results are still correct, only the display is affected. To have proper ROI positions in the display preview, measurements have to be launched again using AZ Video 9.0.

3.4.3 Viewing Conditions in AZ Photo and AZ Video

In AZ Photo and AZ Video, custom Viewing Conditions are lost if the user goes back to settings panel after validating them. They can still be used in the measurement if the user does not go back into settings panel after adding them.

3.5 Bug fixes

3.5.1 General

#4047: Improve the maker detection in some corner cases
#5444: fix the CSF normalization factor in acutance when another CSF is used
#5453: fix a random crash in ViewingConditions when image object is loaded in input dict
#5535: improve the installation of a new version when a WorkflowManager virtual env is still running
#5542: fix the low quality of few pictures in the 3D geometry chapter of main Analyzer manual
#5547: fix the computation of the MTF curve in FocusRangeMeasure
#5583: fix the display of numerous logs about ImageAutoOrientation errors
#5730: fix a crash in MtfComputeLsfTransition
#5753: fix a bad memory management in VideoVignetting leading to high memory usage
#5766: improve the removal of outliers in WideAngle LCA measurement
#5771: fix the management of CFA files saved with NumpyImage in LCA measurement
#5810: fix a memory leak in some processings
#5921: fix a misleading documentation for TCGamma22 input in MTFMeasure
#5954: fix the inline help of AFHDR measurements

3.5.2 AZVideo

#5289: fix a case where the tooltip was staying on screen
#5290: fix the synchronization of the timeline and the video preview and chart

#5354: fix the wrong orientation of the overlay over videos with orientation tag
#5400: fix time pointers stuck on the left
#5402: fix a warning when loading measurement report
#5417: fix drop list stuck on wrong value for Visual Video Noise measurement
#5433: fix the preview frame still visible after having removed the video from the list
#5434: fix a random crash when launching a processing
#5462: improve the display of elapsed time
#5568: improve the smoothness of video preview
#5574: improve the management of trying to open a removed video
#5943: fix a crash when launching AZVideo while an instance is already running

3.5.3 LabManager

#5706: fix the issue to have to rerun the spectrum calibration when adding a preset for MLS
#5707: fix the launch of linearity calibration when spectrometer measurements are displayed in the UI
#5857: clarify the naming restrictions for MLS custom presets
#5919: fix the calibration steps about NIR

3.5.4 Collimator

#5715: clarify the Compass procedure to go back to storage position when using the collimator
#5725: fix wrong ROI detection with collimator in presence of hot pixels
#5726: fix the management of UserSpecifiedROIs input when using Collimator measurement

24-26, quai Alphonse le Gallo
92100 Boulogne-Billancourt - France

www.dxomark.com