**Explanation of uses of the program (c++ script) Lumi**

Program Lumi - after fjnn - from program fotoj3 based on judi based on old programs Judit, and Jcolor2 (see PhD Judit Lopez-Besora)

weights parts of bmp picture and finds average partial luminances and deviations, Toni Isalgue and Judit Lopez-Besora, 2025

This is a "research" application. The results depend on appropriate calibration (with a luminancimeter). Also, might depend on kind of digital camera. Actual values were appropriate for a CANON EOS 550 camera saving pictures as .jpg files (using s-RGB colour space).

Previous: modification /08/2016 produce grayscale with luminances. modification /01/2018 to introduce 3 pictures as HDR (high-definition range)

suggerence 5-9 steps in grey picture for luminances modification: Feb 2018 scales of grey modified

March 2018: give info for colours, all and parts of picture

October 2024 simplified October 2024, 3 rectangles, and write down averages for the rectangles (R G B, L) and standard deviations. Final plot of Luminances improved

(The c/c++ script does not need windows, has been tested under linux ubuntu 24.04 LTS)

Luminance of digital pictures. Needs pictures in non-compressed, s-rgb coded, bmp format .bmp as input:

Picture files: dib1.bmp Where ASA/ISO, Exposition, Aperture are given

dib2.bmp Less exposition, recommended -2 stops

dib3.bmp More exposition, recommended +2 stops

Remember: The 3 bmp pictures need to be: Well aligned, non-compressed, s-rgb colours, and of the same width, height. If you have only 1 picture, copy and rename the picture, introduce 3 identical pictures and assign steps between pictures = 0. Now, pictures up to 9000\*6000 pixels can be introduced (modify dimensions under your responsibility if needed).

Numerical file (in ascii) input.dat Contents:

Line1: C1, integer, 0 normal processing reads ISO/ASA, Aperture Exposition

Line2: C2, integer, stops between pictures ('bracketing')

Line3: C3, integer, number of luminance bands (recommended 5-7)

Line5: C5c float, value of gamma compression (if 0, takes 2.2 as default)

Line6: if C1=0, ISO/ASA of picture dib1.bmp (example: 200)

Line7: if C1=0, Aperture of picture dib1.bmp (ex: if f/11, give: 11)

Line8: if C1=0, Exposition of dib1.bmp (ex: if 1/320 s, give: 320)

Line9: C9 integer ask for RGB values of C9 points (xl[], yl[]). 0 if not

Lines10-11\*: xl[], yl[] C9 times 2 numbers (2\*C9 files) (\*if C9=0, omit) if C9>0 Attention bmp starts left down corner

3 times:

Lin12-13: xini, xefe: x start, x end, x define rectangle to find averages

Lin14-15: yini, yefe, y start, x end, y define rectangle to find averages

Attention: coordinates from down, left picture

Line24: Value Normalize cd/m2 Luminance graph (to the right last division)

Line25: 0, normal computation (reserved)

Results of the script:

Res.bmp: bitmap file

picture in grey scale, with bars with the luminance intervals in the lower part (values in the numerical file)

The output res.bmp will have the same headings as the central picture dib1.bmp

Lumi.dat: ascii file output with numerical results:

Average and standard deviations in the pictures, luminances,

Average and standard deviations in the defined rectangles, RGB values and standard deviations,

Luminance values corresponding to each bar on the lower part of the output picture

First bar values from 0 and first value in luminance (cd/m2),

second bar values from first and second, and successively

Values of luminance points indicated (average +/- 1 pixel, total 9) for the 3 pictures and for the central picture only