* 更新履歴

2021/06/28　作成

2021/06/30　関数追加

LCEPTマニュアル

* 動作環境
  + OS：Raspberry Pi OS (Debian系Linux)
  + メモリ：８GB以上推奨
* 必要ソフトウェア・ライブラリ
  + Python v3.7.3
  + gphoto2　 v2.5.27
  + dcraw v9.28
  + ptpcam
  + OpenCV
  + Qt
* コードの配布方法
  + GitHubで開発者として共有（オリジナルを更新履歴を残して変更可能）
  + BOX or Dropbox で共有（GitHubで管理しているコードのコピー）
* 配布コード概要
  + common\_tools
    - setup.txt：上記の必要となるソフトウェア等のインストール手順の解説。
    - view\_img.py：画像の表示および画素位置の確認。
    - get\_point\_data.py：HDR合成したRGBのcsvファイルから、point.csvに記述した情報に基づき、その範囲のRGB値の平均値を取得。
    - make\_luminance\_image.py：輝度分布のcsvファイルから、疑似カラー画像作成、および任意の位置の輝度値を表示。
    - make\_luminane\_image\_m.py:輝度分布のcsvファイルに2値化画像のマスクをかけ、その結果の疑似カラー画像、、および任意の位置の輝度値を表示。
    - calc\_e.py：輝度分布のcsvファイルから、照度に変換。
  + カメラ機種ごとの計測用コード
    - capture.sh：evList.csvの情報に従い、カメラの露光を設定し画像取得する。
    - conv.sh：取得したRAW画像をTIFF画像に変換し、XYZのcsvファイルを作成する。
    - conv\_loop.sh：画像保存先のディレクトリ一覧を記述したdirList.txtに含まれている全てのディレクトリに対して、conv.shを実行する。
    - conv\_tiff.sh：取得したRAW画像をTIFF画像に変換する。
    - conv\_xyz.py：TIFF画像からXYZのcsvファイルへの変換を行うメインのコード。
    - conv\_hdr.py：TIFF画像からRGBのcsvファイルへの変換を行うメインのコード。
* システムのディレクトリ構成（ディフォルトの設定方法）
  + ホームディレクトリ（/home/pi）の直下に、ディレクトリ”LCEPT“を作成し、その下位に以下の要領でディレクトリを構成し、コードと撮影データを管理する。
  + 下図で黒のディレクトリおよびファイルは撮影用のコードをPCにコピーする際に作成する。
  + 下図で赤のディレクトリおよびファイルは一連（実験種別など）のシーンの撮影をする際に作成する。ここで「scenario01」や「scenario02」のディレクトリを「画像親ディレクトリ」と呼ぶこととする。
  + 下図で青のディレクトリおよびファイルはシステムの実行により作成される。ここで撮影を行うごとに作成される撮影日時に同期した「yymmdd\_hhmmss」のディレクトリを「画像子ディレクトリ」と呼ぶこととする。
  + /LCEPT/tools/の下位に機種ごとの撮影用コードを保存するが、同一機種であっても、レンズの違いによりレンズの減光特性が異なる場合は、この階層に別ディレクトリを作成し管理する。（レンズ減光補正はconv\_xyz.pyのコード中に記載しているため）



* 各コードの使用方法
  + 以下の解説で、コマンドライン引数としてディレクトリやファイルを指定する場合に、[ディレクトリ名　や　ファイル名]と表記する。ディレクトリ名やファイル名はフルパス（/home/pi/LCEPT/data/xxxxxx）で入力する。実際の入力においては[ ]は入力しない。
  + capture.sh
    - 撮影画像を保存する画像親ディレクトリを作成し、その下位に撮影に用いる露光情報を記したevList.csvを配置する。
    - カレントディレクトリを、撮影に用いるcapture.shを保存してあるディレクトリに移動し、以下を実行。
    - $ sudo ./capture.sh [画像親ディレクトリ]
    - 画像親ディレクトリ下位に、画像子ディレクトリが作成され、その中に撮影した画像が保存される。また、画像子ディレクトリ名は、dirList.txtに順次追記される。
  + conv.sh
    - カレントディレクトリを、変換に用いるconv.shを保存してあるディレクトリに移動し、以下を実行。
    - $ sudo ./conv.sh [画像子ディレクトリ]
    - 画像子ディレクトリ内のpicInfo.csvおよびsysInfo.csvに基づき、RAW画像からTIFF画像への変換、およびRGB値からXYZ値への変換を行い、XYZ値をcsvファイルで書き出す。
  + conv\_loop.sh
    - カレントディレクトリを、変換に用いるconv\_loop.shを保存してあるディレクトリに移動し、以下を実行。
    - $ sudo ./conv\_loop.sh [画像親ディレクトリ]
    - 画像親ディレクトリにあるdirList.txtに記載されている画像子ディレクトリすべてに対してconv.shが実行される。
  + conv\_tiff.sh
    - カレントディレクトリを、変換に用いるconv\_tiff.shを保存してあるディレクトリに移動し、以下を実行。
    - $ sudo ./conv\_tiff.sh [画像子ディレクトリ]
    - 画像子ディレクトリ内のRAW画像がTIFF画像に変換される。
  + conv\_xyz.py
    - 通常はconv.shおよびconv\_loop.sh内で呼び出されて用いられる。
    - TIFF画像を合成してHDR画像を作成し、XYZ値への変換を行うコード。
    - XYZ値への変換を行う際の各種変数（HDR合成に用いるRGB値の信号範囲、XYZへの変換係数、レンズの補正）は、sysInfo.csvに記載し、必要に応じて記述を変更する。
    - カレントディレクトリを、変換に用いるconv\_xyz.pyを保存してあるディレクトリに移動し、以下を実行。
    - $ python3 conv\_xyz.py --input [画像子ディレクトリ]
    - 画像子ディレクトリ内のpicInfo.csv およびsysInfo.csvを読み込みXYZ値のcsvファイルを書き出す。
  + conv\_hdr.py
    - カレントディレクトリを、変換に用いるconv\_hdr.pyを保存してあるディレクトリに移動し、以下を実行。
    - $ python3 conv\_hdr.py --input [画像子ディレクトリ]
    - 画像子ディレクトリ内のpicInfo.csvを読み込みHDR合成後のRGB値のcsvファイルを書き出す。
  + view\_img.py
    - カレントディレクトリを、view\_img.pyを保存してあるディレクトリに移動し、以下を実行。
    - $ python3 view\_img.py --input [画像子ディレクトリ/特定のTIFFファイル]
  + get\_point\_data.py
    - 画像子ディレクトリにRGB値を取得したい画素位置および範囲を記述した、point.csvファイルを保存する。
    - カレントディレクトリを、get\_point\_data.pyを保存してあるディレクトリに移動し、以下を実行。
    - $ python3 get\_point\_data.py --input [画像子ディレクトリ]
    - 画像子ディレクトリにあるRGB値のcsvファイルから、point.csvに記載されている箇所のRGB値を平均化し、rgb\_vals.csvのファイルとして書き出す。
  + make\_luminance\_image.py
    - カレントディレクトリを、make\_luminance\_image.pyを保存してあるディレクトリに移動し、以下を実行。
    - $ python3 make\_luminance\_image.py --input [画像子ディレクトリ/dataY.csv]
    - 疑似カラー画像と輝度ヒストグラムが表示される。
  + make\_luminance\_image\_m.py
    - 輝度画像に対してマスクをかける領域の2値化画像を作成し、「mask.jpeg」というファイル名で、マスクをかける対象の輝度CSVファイルがあるディレクトリに保存する。
    - マスク画像は、TIFF画像の同じ画素数の配列とし、平均輝度の算出範囲に指定しない箇所を黒（0, 0, 0）、算出範囲に指定する箇所を白（255, 255, 255）とする。
    - カレントディレクトリを、make\_luminance\_image\_m.pyを保存してあるディレクトリに移動し、以下を実行。
    - $ python3 make\_luminance\_image\_m.py --input [画像子ディレクトリ]
    - 疑似カラー画像と輝度ヒストグラムが表示される。
  + calc\_e.py
    - カレントディレクトリを、calc\_e.pyを保存してあるディレクトリに移動し、以下を実行。
    - $ python3 calc\_e.py --input [画像子ディレクトリ/dataY.csv]
    - 照度値が出力される。
* 計測システムの校正（sysInfo.csvの記述方法）
  + RGB＞XYZ変換係数の導出
    - 校正用画像の保存先の画像親ディレクトリを作成する。
    - 校正時に用いるevList.csvを画像親ディレクトリに保存する。
    - XYZ値が既知（CS100-A等での実測含む）である色票（マクベスカラーチェッカー）等を用意し、その色票を照明強度が変更でき、なるべく均一な照明環境下に配置する。
    - 色票への照明強度を複数段階設定し、各照明強度下において色票が画像中央となるよう、システムで撮影（capture.sh）するとともに、各照明強度下での色票のXYZ値を取得（計測）する。
    - 撮影画像を現像（conv\_tiff.sh）する。
    - 現像した画像を表示（view\_img.py）し、撮影した画像中の各色票に対して、色票中心のx座標, 色票中心のy座標, 中心からの幅dを確認。これらを記載したpoint.csvファイルを画像子ディレクトリに保存する。
    - HDR画像のRGB値csvファイルを作成（conv\_hdr.py）する。
    - RGB値のcsvファイルからpoint.csvファイルの情報を用いて、色票箇所のRGB値の平均値を取得（get\_point\_data.py）する。
    - XYZ値とRGB値から重回帰分析を行い、変換係数を導出する。
    - 得られた変換係数をsysInfo.csvの定数定義に反映させる。
  + 画像描画範囲の確認
    - 撮影・現像したTIFF画像から、画像の幅および高さを確認し、sysInfo.csvの定数定義に反映させる。
    - 魚眼レンズを使用する場合、画角180度に相当する、中心からの半径を確認し、sysInfo.csvの定数定義に反映させる。
    - 画像中央から画像端へかけて、レンズによる減光補正を行う場合は、conv\_xyz.pyコード中のmatLensに値を代入している箇所を直接編集する。
  + 変換に用いるRGB値範囲の設定
    - それぞれの露光量で撮影された画像において、HDR画像への変換に用いるRGB値の上限および下限を設定し、sysInfo.csvの定数定義に反映させる。
    - TIFF画像はガンマ１で現像しているため、原理的にはRGB値が0または255でない限り、HDR画像への変換において有効であるが、ノイズの影響を考慮して、下限と上限を設定する。
* 計測の手順
  + 撮影した画像等を保管するディレクトリ（画像親ディレクトリ）を /home/pi/LCEPT/data 以下に作成（例えばscenario01とする）し、作成したディレクトリに、計測システムの設定ファイル（sysInfo.csv）および露光量設定ファイル（evList.csv）を保存する。
  + カレントディレクトリを計測に用いるシステムのbashファイル（capture.shなど）を保存しているディレクトリに移動し、capture.shを実行する。
  + 画像子ディレクトリ（capture.shを実行した時刻のディレクトリ名）が自動で生成され、撮影画像が保存される。また画像子ディレクトリには、画像ファイル名や露光量の情報を記述したpicInfo.csvとsysInfo.csvのコピーも同時に保存される。
  + 撮影されたraw画像の現像およびXYZ値への変換は、conv.shまたはconv\_loop.shで行う。
  + 輝度値や照度値の確認は、出力された輝度分布のCSVファイルから、make\_luminance\_image.pyやcalc\_e.pyを用いて算出する。