Calculating CCMs for Fast-OpenISP

Will be using cameraRGB values from [dcraw.c]

And the standard XYZ<-linear sRGB transformation matrix.

```
XYZsRGB = np.array([[0.412390799265959, 0.357584339383878, 0.180480788401834],
[0.212639005871510, 0.715168678767756, 0.072192315360734],
[0.019330818715592, 0.119194779794626, 0.950532152249661]])
```

From DCRAW, get your camera model cameraRGB values:

```
"Nikon D3200", 0, 0xfb9,
  { 7013,-1408,-635,-5268,12902,2640,-1470,2801,7379 } },
```

```
nikon = [7013,-1408,-635,-5268,12902,2640,-1470,2801,7379]
```

Reshape from flat array to 3x3 matrix and divide by 10,000.

```
nikon = (np.array(nikon).reshape(3,3))/10000
```

Then calculate camera RGB <- linear sRGB matrix by dot product of previous two.

```
camerasRGB = np.dot(nikon, XYZsRGB)
```

To preserve neutral pixel (white/gray), you row-wise normalise camerasRGB.

```
sumRows = camerasRGB.sum(axis=1)
normalised = camerasRGB/sumRows[: , np.newaxis]
```

Final step is to invert and multiply by 1024.

```
finalCCM = (np.linalg.inv(normalized_array))*1024
```

Nikon D3200

```
>>> nikon
array([[ 0.7013, -0.1408, -0.0635],
       [-0.5268,  1.2902,  0.264 ],
       [-0.147 ,  0.2801,  0.7379]])
>>> XYZsRGB
array([[0.4124, 0.3576, 0.1805],
       [0.2126, 0.7152, 0.0722],
       [0.0193, 0.1192, 0.9505]])
>>> camsRGB=np.dot(nikon,XYZsRGB)
>>> camsRGB
array([[0.25805649, 0.14251552, 0.05606214],
       [0.0621394 , 0.76583616, 0.24899704],
       [0.01316793, 0.235718  , 0.69506367]])
>>> sumRows= camsRGB.sum(axis=1)
>>> norm = camsRGB/sumRows[:,np.newaxis]
>>> norm
array([[0.56512744, 0.3121    , 0.12277255],
       [0.05769822, 0.71110088, 0.23120091],
       [0.01394982, 0.2497146 , 0.73633557]])
>>> nikonCCM = (np.linalg.inv(norm))*1024
>>> nikonCCM
array([[1896.2644323 , -810.61572837,  -61.64870392],
       [-159.80139391, 1686.78949762, -502.98810371],
       [  18.26909996, -556.68643368, 1562.41733372]])
>>>
```

Canon 600D

```
>>> canon
array([[ 0.6461, -0.0907, -0.0882],
       [-0.43  ,  1.2184,  0.2378],
       [-0.0819,  0.1944,  0.5931]])
>>> XYZsRGB
array([[0.4124, 0.3576, 0.1805],
       [0.2126, 0.7152, 0.0722],
       [0.0193, 0.1192, 0.9505]])
>>> camsRGB=np.dot(canon,XYZsRGB)
>>> camsRGB
array([[0.24546656, 0.15566328, 0.02623841],
       [0.08628938, 0.74597744, 0.23638238],
       [0.01900071, 0.18044496, 0.56299428]])
>>> sumRows= camsRGB.sum(axis=1)
>>> norm = camsRGB/sumRows[:,np.newaxis]
>>> canonCCM = (np.linalg.inv(norm))*1024
>>> canonCCM
array([[ 1924.68347469, -1057.41213554,   156.72866084],
       [ -224.89075439,  1756.30958322,  -507.41882882],
       [    7.12272003,  -527.22672627,  1544.10400624]])
>>>
```

Lumix S1 – From RAWPY

	0	1	2
0	0.97440	-0.39050	-0.07790
1	-0.48990	1.28070	0.23240
2	-0.07980	0.16300	0.58270
3	0.00000	0.00000	0.00000

```
>>> lumix
array([[ 0.9744, -0.3905, -0.0779],
       [-0.4899,  1.2807,  0.2324],
       [-0.0798,  0.163 ,  0.5827]])
>>> XYZsRGB
array([[0.4124, 0.3576, 0.1805],
       [0.2126, 0.7152, 0.0722],
       [0.0193, 0.1192, 0.9505]])
>>> camsRGB=np.dot(lumix,XYZsRGB)
>>> camsRGB
array([[0.31731879, 0.05987416, 0.07364115],
       [0.07472738, 0.76847048, 0.22493579],
       [0.01299039, 0.15749896, 0.55122105]])
>>> sumRows= camsRGB.sum(axis=1)
>>> norm = camsRGB/sumRows[:,np.newaxis]
>>> norm
array([[0.70384824, 0.13280752, 0.16334423],
       [0.0699607 , 0.71945162, 0.21058768],
       [0.01799945, 0.21823014, 0.76377041]])
>>> lumixCCM = (np.linalg.inv(norm))*1024
>>> lumixCCM
array([[1480.82428914, -193.47091509, -263.35337406],
       [-145.99303865, 1572.28145018, -402.28841153],
       [   6.81626131, -444.68445207, 1461.86819076]])
```