

```
1: %-----%
2: %----- Matlab Image Intensity Levels converter -----%
3: %-----%
4: %
5: % !!!! TIF version !!!!
6: %
7: % V0.20 2004/12/16
8: % Changelog:
9: % from V0.11 (2004/12/16)to V0.20: removed extrapolation for dynamic range < 256
10: % from V0.1 (2004/12/15) to V0.11: added +0.5 for correct rounding
11: %
12: %
13: %
14: % Description:
15: % This Matlab script can be used to convert a 16-bit tif image into a 8bit*
16: % tif or jpg (100% quality) by defining the intensity range equivalent to
17: % 0.."256" (if the dynamic range before is greater than 256 intensity levels, the
18: %         levels are lowered linearly otherwise the dynamic range is kept the
19: %         same but clipped within 256 intensity levels)
20: % (*: the bit level may be adjusted too if desired - possible with tif images)
21: %
22: %
23: % Input arguments:
24: %     1st: "path" - path string to the folder where the images to be converted
25: %                are contained (e.g. 'c:\pivimg\')
26: %     2nd: "Imin" - original level equivalent to 0
27: %     3rd: "Imax" - original level equivalent to 255
28: %
29: % Example 1:
30: % to convert the original levels ranged 100 to 500 into 0 to 255
31: % (linearly interpolated / shifted) with all files in folder 'd:\myimages\'
32: % type: LevelConv('d:\myimages\', 100, 500)
33: % the intensity level 100 will now correspond to 0 and 500 to 255, with all levels
34: % inbetween interpolated linearly
35: %
36: % Example 2:
37: % if the original levels now ranged 100 to 200
38: % type: LevelConv('d:\myimages\', 100, 200)
39: % the intensity level 100 will now correspond to 0 and 200 to 100, with all
40: % original levels <100 now being zero and the ones >355 now equal to 255;
41: % the reason for this is not to artificially increase the actual dynamic range and
42: % thus introduce additional potential errors
43: %
44: %
45: % basic function description (based on the assumption of a 16 bit tif originating
46: % from a 10 bit camera and shifted by multiplication with 2^6)
47: % note images aquired with a different software than epix might behave differently
48: % 1. the 16-bit tif is read
49: % 2. the 16-bit tif is shifted to 10 bit by multiplication with 2^-6
50: % 3. the lower level is set to zero by subtracting the lower level (e.g. 100)
51: % 4. the range of 0-255 is achieved by multiplication (linear shifting)
52: %
53: %
54: function LevelConv(path, Imin, Imax)      % function definition
55: %
56: % Adjustable parameters
57: imgReadFormat = 'tif'; % format of image to read, choose from: 'tif', 'jpg', ...
58: imgWriteFormat = 'tif'; % format to write the image to; carefull: set quality etc
59: % as well!
60: PicResBit = 16; % picture resolution
61: CamResBit = 10; % camera resolution in bit
62: %
63: % adjust Imax (new in V0.20)
64: if ((Imax - Imin) < 255)
65:     Imax = Imin + 255;
66: end;
```

```
67:
68: % get files from path
69: files = dir ([path, '*.', imgReadFormat]);
70:
71: % process all files in the path
72: for i=1:size(files,1)
73:     X = imread([path,files(i).name]); % read a particular image file
74:     ImgFName = [path, '8bit', ...
75:         files(i).name(1:findstr(files(i).name,imgReadFormat)-1), imgWriteFormat]
76: % image file name to be written display not suppressed to allow viewing progress
77:     X = uint8( ( double(X)*2^(CamResBit-PicResBit) - Imin ) ...
78:         /(Imax-Imin)*255 + 0.5); % image manipulation as described
79:     % note any levels above 255 will be cropped to 255 as unlike other software
80:     % Matlab won't rebegin at zero due to bit shifting (makes life easier)
81:     imwrite(X,ImgFName,imgWriteFormat); % writing the new image
82: end
83:
```