

## Computer Exercise B. Characterization of infrared imaging sensors

### Preparations

Read *The Ultimate Infrared Handbook for R&D Professionals*, chapters 1 and 2.

### Content

Part 1: Basic physics and measurement of infrared radiation.

Part 2: Hands-on experiments with a live IR-camera. In this part you will use a simple handheld IR-camera and do some simple experiments that illustrate some basic properties of IR-imaging.

Part 3: Non-uniformity correction of IR-images. Images from an IR-camera typically have a lower quality than a standard camera for the visible range. This is because it is more difficult to implement the necessary measurement process for an IR-sensor, that makes each pixel produce a uniform response to an incoming uniform radiation. In this part of the exercise, you will find out how to deal with this problem.

### Part 1

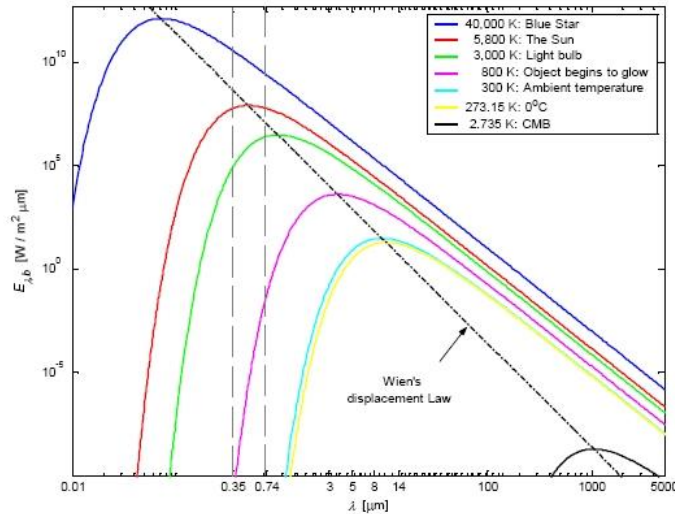
*The Ultimate Infrared Handbook for R&D Professionals* does not give the explicit formula for Planck's radiation law for blackbodies. Here it is

$$M(\lambda, T) = \frac{2\pi hc^2}{\lambda^5 (e^{hc/\lambda kT} - 1)} \quad (1)$$

where  $M(\lambda, T)$  is the spectral emittance or spectral excitance. If  $\lambda$  is given in meters it has a physical unit of Joule/second/meter<sup>2</sup>/meter, or Watt/meter<sup>3</sup>.  $T$  is given in Kelvin (K), and

$c$  is the speed of light,  
 $h$  is the Planck constant,  
 $k$  is the Boltzmann constant.

In the figure below the spectral emittance is plotted as a function of  $\lambda$  for different temperatures  $T$ . Notice that the wavelength is given  $\mu$ -meters and that the plot has a logarithmic scale both on the wavelength and the emittance.



For infrared sensors the emittance is measured only within certain well-specified wavelength bands  $\lambda_1$  and  $\lambda_2$ , e.g., 3-5  $\mu\text{m}$  and 8-12  $\mu\text{m}$  for the sensors used in part 3 of this exercise, by integrating the spectral emittance  $M_\lambda(\lambda)$  in the corresponding range:

$$M_{\lambda_1-\lambda_2}(T) = \int_{\lambda_1}^{\lambda_2} M_\lambda(\lambda, T) d\lambda \quad (2)$$

This means that for a specific camera, for which  $\lambda_1$  and  $\lambda_2$  are specified and fixed, we can compute the integrated emittance that can be detected by the pixel as a function of the temperature  $T$  of the object. It is not possible to derive an analytic expression, but it can be computed numerically by a simple approximation of the above integral as a finite sum. At your help, there is a Matlab script:

`part1script`: contains a skeleton for a script that produces all the necessary computations for an approximation of the emittance in (2). *You need to specify some parameters here.*

**Before you come to the exercise:** What are the numerical values of  $c$ ,  $h$ , and  $k$ , in Planck's radiation law? Give them in SI-units

$c =$   
 $h =$   
 $k =$

You need to give the values for these constants in the script. Do this. You also need to specify `lambda1`, `lambda2`, and `n` = the number of terms in the approximation of (2). One of the sensors you will look at in part 3 has  $\lambda_1 = 3.5 \mu\text{m}$  and  $\lambda_2 = 5 \mu\text{m}$ . Finally, you need to specify for which temperatures the integrated emittance is computed and plotted. Start with the range 1 to 1000 K. Choose a suitable  $n$  and run `part1script` to compute and plot the integrated emittance as a function of temperature.

Q: What does the function look like? Is it linear, or what?

---

At some configuration this sensor can detect a maximum in-band irradiance of 40 W/m<sup>2</sup>, and becomes saturated for intensities above that.

Q: Which temperature does this intensity correspond to?

---

The digital output signal  $\mu$  [DN, digital number] increases linearly with the detected radiance  $L$ . For an ideal blackbody source  $L=M/\pi$ . Disregarding the atmospheric influence this is expressed in

$$\mu = K_1 \int_{\lambda_1}^{\lambda_2} L_{\lambda}(\lambda, T) \tau_{\lambda}(\lambda) d\lambda + K_0 \quad (3)$$

where  $K_1$  is the overall system gain and  $\tau_{\lambda}$  is the spectral transmission of the optics. Besides the source, there are additional signal contributions (due to e.g. internal emission) which will add a bias  $K_0$ . The digital output signal  $\mu$  can be transformed and presented in units of radiance or temperature.

In some IR-sensors, the intensity measurement at a pixel can be done over a relative large wavelength range, but often it is limited by an optical filter in front of the sensor, a filter that defines  $\lambda_1$  and  $\lambda_2$ .

Q: How can you change the upper or lower wavelength limits to increase the temperature range of this sensor?

The ratio between the emittance from a real source,  $M_S$ , and the emittance from an ideal blackbody source,  $M_{BB}$ , at the same temperature is called the spectral emissivity,  $\varepsilon_s$

$$\varepsilon_s(\lambda) = \frac{M_S(\lambda)}{M_{BB}(\lambda)} \quad (4)$$

If  $\varepsilon_s$  is a constant  $< 1$  for all wavelengths the source is called a greybody for which  $L=\varepsilon_s M/\pi$ . In the general case  $\varepsilon_s$  is a function of wavelength. Some approximate values of  $\varepsilon_s$  is shown in the table (taken from <http://www.eplus-innovation.com/knowledge.asp>) on next page.

<i>Emissivity</i>		<i>Emissivity</i>	
Material	Emissivity ( $\epsilon$ )	Material	Emissivity ( $\epsilon$ )
Aluminum alloy-oxidized	0.40	Asbestos Board	0.96
Aluminum-highly polished	0.04-0.06	Asphalt, tar, pitch	0.90-0.98
Aluminum-oxidized	0.11-0.31	Brick-red and rough	0.93
Aluminum-Anodized sheet	0.55	Brick-fireclay	0.75
Brass-Oxidized	0.60	Carbon-filament	0.53
Brass-polished	0.03	Carbon-lampblack	0.96
Chromium-polished	0.10-0.38	Cement	0.54
Copper-polished	0.02-0.05	Ceramic	0.90-0.94
Copper-heated at 600 °C	0.57	Concrete	0.92-0.97
Gold-pure, highly polished	0.02	Frost crystals	0.98
Iron-polished	0.21	Glass	0.80-0.95
Iron-oxidized	0.94	Human skin	0.98
rusted iron plate	0.65	Ice	0.96-0.98
Iron-rough steel plate	0.94-0.97	Marble-polished light gray	0.90
Lead-gray and oxidized	0.28	Paints, lacquers, varnishes Black	0.90-0.95
Mercury	0.09-0.12	Paints, lacquers, varnishes aluminum paints	0.55
Nickel-polished	0.12	Paints, lacquers, varnishes flat black lacquer	0.96-0.98
Nickel-oxidized	0.37-0.85	Paints, lacquers, varnishes white lacquer	0.95
Platinum-pure polished plate	0.05-0.10	Paper	0.94
Platinum-wire	0.06-0.16	Plastic	0.84-0.94
Silver-pure and polished	0.02-0.03	Porcelain-glazed	0.92
Stainless steel-polished	0.16	Propellant-Liquid rocket engine	0.90
Stainless steel-oxidized	0.74-0.87	P.V.C.	0.91-0.93
Tin-bright	0.07-0.08	Quartz-opaque	0.75
Tungsten-filament	0.32-0.39	Rubber	0.95-0.97
Zinc-polished commercial pure	0.05	Sand	0.90
Zinc-galvanized sheet	0.23	Snow	0.96-1.00
		Soil	0.92-0.95
		Tape-Masking	0.92-0.95
		Wallpaper	0.85-0.90
		Water	0.95-0.96
		Wood-planed oak	0.82-0.89
<b>METALS</b>		<b>NONMETALS</b>	

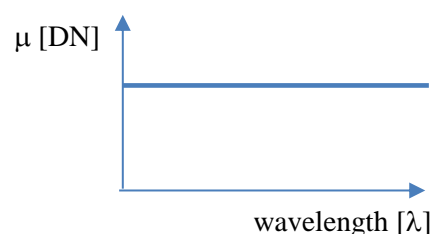
## Part 2

Hands-on experiments illustrating some basic properties of IR-imaging.

There are two main classes of *infrared detectors*: thermal detectors and photon detectors. The camera you will use in these experiments is based on an array with thermal detectors.

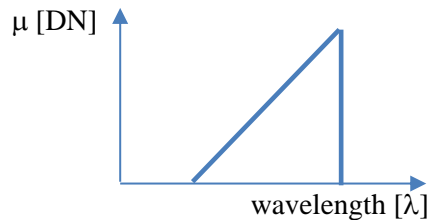
Characteristic features for a thermal detector

- Based on a temperature change of a response element  
⇒ Slow response, a typical frame rate is up to 50 Hz (video rate)
- No cooling required  
⇒ Technically simple and cheap (compared with the photon detector)
- Lower sensitivity than the photon detector
- Ideal thermal detector: If radiation falls on the detector at different wavelengths but with constant power, and plot with output signal vs. wavelength is as follows



Characteristic features for a photon detector:

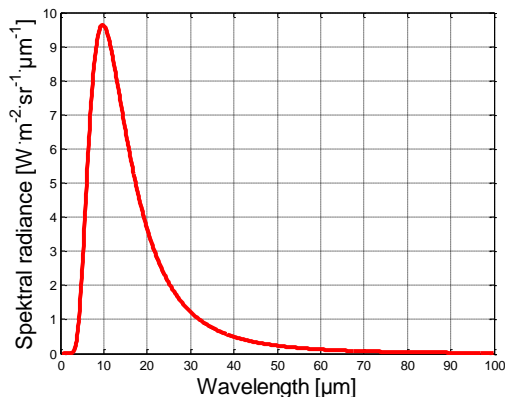
- Based on a photon-electron interaction  
⇒ Fast response, a frame rate  $> 10000$  Hz is possible
- Requires cooling below 80 K to give good performance in the infrared region  
⇒ Technically complex and expensive
- High sensitivity
- Ideal photon detector: If radiation falls on the detector at different wavelengths but with constant power, a plot with output signal vs. wavelength is as follows



**Task 1.** The default camera setting of the emissivity is 0.95, which is in good agreement with the emissivity of human skin ( $\sim 0.98$ ). Use the camera to measure the apparent temperature of your or friends skin. Does the value seem reasonable? Can you think out the apparent temperature if the emissivity setting is changed to an erroneous value, e.g. 0.30? Find out the answer using the camera!

**Task 2.** A problem with imaging in the infrared region is that expensive lens materials like germanium have to be used, due to that the transmission of ordinary lens materials is very poor. Investigate the infrared transmission of the glass plate! Now compare with the transmission of the plastic bag. Can you find out why plastic lenses are not used in infrared imaging?

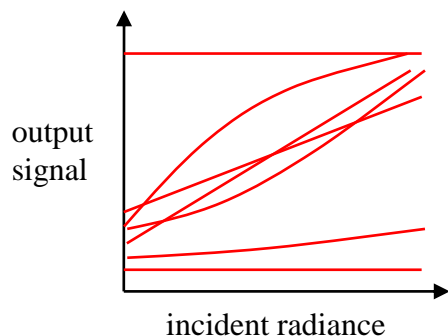
**Task 3.** The figure below shows the spectral radiance at the temperature  $25^\circ\text{C}$ . The maximum emittance is found between  $8\text{--}12\text{ }\mu\text{m}$ , which is the approximate spectral range for the infrared camera you are using in these hands-on experiments. As you have found out the camera is not able to see through e.g. the table. The “see-through” ability is achieved using longer wavelengths,  $70\text{ }\mu\text{m} - 3\text{ mm}$  (i.e. the THz region). However, based on the characteristic features for a thermal detector, it should be possible to use the camera to “see-through” but at least two modifications of the camera probably need to be done. Can you find out which two? Hints: Consider Task 2 and the figure below (you can also compare a motorist daytime and nighttime).



**Figure:** Spectral radiance at 25 °C

### Part 3

Imaging systems based on staring arrays have high sensitivities and high frame rates (compared to the corresponding scanning systems). One disadvantage is that the sensitivities of the detector elements in the same sensor (detector signal/incident radiance) are different, corresponding to a type of static pixel noise (FPN, fixed pattern noise), which will decrease the sensor performance. This applies particularly to detector materials used in the infrared spectral range, where FPN is the main noise contribution.



**Figure:** Schematic sketch of different responses of some pixels in a detector array resulting in fixed pattern noise (for clarity the differences between the responses have been exaggerated in the figure).

### Useful Matlab commands

*imagesc*: scales data and displays data as an image

*colormap(gray)* : sets the current figure's color map to “gray”; this presentation is for most infrared images preferable to the root's default, whose setting instead is “jet”

*imcontrast*: adjust contrast tool

*axis image*: sets the aspect ratio so that width and height of the pixels are equal in size.

$A * X = B \Rightarrow X = A \backslash B = \text{inv}(A) * B$

### Available m-files

There are two m-files available for solving the tasks B-1, B-2, B-3, B-4, C-1, C-2 and C-3:

*part3\_B*: to solve the tasks B-1, B-2, B-3 and B-4; calls the functions *calc\_C*, *NUC* and *id\_dp*.

*part3\_C*: to solve the tasks C-1, C-2 and C-3; calls the function *id\_dp*

Functions:

*calc\_C*: calculates correction coefficients

*NUC*: performs a non-uniformity correction of a raw image

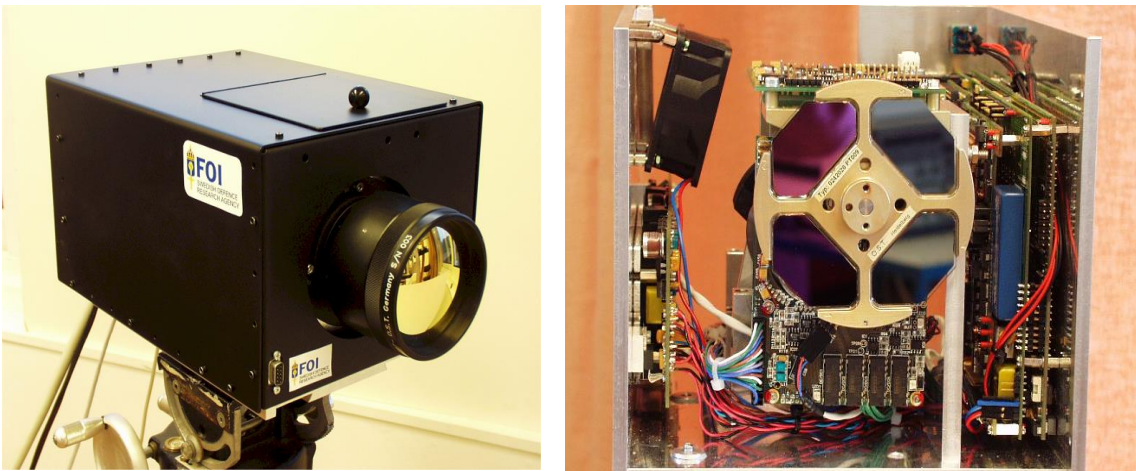
*id\_dp*: identifies and replaces dead pixels in an image using median filtering

### Sensor 1: Multispectral sensor (AIM)

All data are loaded into Matlab workspace: *load Refdata1*, etc.

A multispectral sensor collects data in several spectral bands at the same time. This sensor is based on a cooled MCT-detector with the spectral range 1.5-5.2  $\mu\text{m}$ . The number of pixels in the focal plane array is 288x384. The sensor is equipped with a rotating filter wheel containing four band pass filters (see figure below).

The full frame rate is 4x25 Hz. Generally the ability to detect an object increases with the number of spectral bands and the width of the spectral range covered by the spectral bands. However, having several spectral bands and a broad spectral range also makes the camera more technically complex – and more expensive.



**Figure:** Multispectral sensor based on an MCT detector. Left: with the optics mounted; right: the uncovered camera with the rotating filter wheel containing four band pass filters: Filter 1 (1.55 – 1.75  $\mu\text{m}$ ), Filter 2 (2.05 – 2.45  $\mu\text{m}$ ), Filter 3 (3.45 – 4.15  $\mu\text{m}$ ), Filter 4 (4.55 – 5.2  $\mu\text{m}$ )

The data used in the exercise has been collected in the spectral band 4.55-5.2  $\mu\text{m}$  (filter 4). Refdata1, Refdata2 and Refdata3 are registrations of surface radiators (homogeneous surfaces) at three different temperatures. In particular FPN is a characteristic feature for the detector material MCT. In addition the detector array was severely degraded at the

time for the measurement which is reflected by a large number of bad pixels. This can be observed as salt and pepper in the image.

Refdata1: registered surface radiator at 3.5 °C during 0.8 s (288x384x20, double)

Refdata2: registered surface radiator at 22 °C during 0.8 s (288x384x20, double)

Refdata3: registered surface radiator at 34 °C during 0.8 s (288x384x20, double)

Scenedata: registration of “Slestadsrondellen” during 0.8 s (288x384x20, double)

## A. Plotting the sensor data

**A-1.** Plot an image of an area radiator (e.g. the first image in the first reference data set, Refdata1) and look at the pixel values over the array. An ideal sensor would only show the temporal noise, which can be observed as random pixel fluctuations over the array. Does the pixel noise seem to be randomly distributed over the array?

The integrated radiances for the temperatures 3.5, 22.5, 33.5 °C are 0.809, 1.58 and 2.25 [W/(m<sup>2</sup>,sr)], respectively.

**A-2.** Plot the pixel means vs. the integrated radiances for the reference data sets. The pixel mean is estimated by averaging the pixels over the array in the first image in the data set (the difference between using only the first image and all 20 images in the calculation is here negligible). The mean pixel value should be linear to the radiance level. One can expect that the mean pixel value  $\approx 0$  at a radiance level = 0  $\Leftrightarrow$  the curve is a line through the origin. Is the curve a line through the origin ? If not, can you find any explanation ?

(Hint: For an ordinary digital camera the optical power on the detector is close to zero when the lens is covered, but what about an infrared camera ?)

**A-3.** Identify the row and column numbers for 2-3 bad pixels (“salt and pepper”) and 2-3 “normal” pixels. Plot the single pixel values against the pixel means for the three levels and compare the results for bad pixels (“salt and pepper”) and “normal” pixels.

**A-4.** Plot also some single pixel values over time (there are 20 time points in each data set). The temporal noise for a bad pixel which is truly dead (a pixel showing one constant value) is of course zero. Try to find one or more bad pixels (“salt and pepper”) where the value is not constant over time. Compare the temporal noise for these bad pixels with “normal” pixels. Is there any difference?

### Summary of A

In an ideal sensor all pixels have the same response and the same temporal fluctuation (variance). As has been shown a real sensor is not ideal. The non-ideality also concerns sensors in the visual region, but here the non-uniformity is small compared to sensors in the infrared. Different methods are available to perform non-uniformity corrections (NUC). Below a method based on polynomial fitting is presented [1].

## B. Reference based NUC

As has been shown under A, the response curve of a single pixel tends to be non-linear. An algorithm for polynomial fitting has been proposed by Schultz and Caldwell [2]. For pixel  $j$  the difference  $\Delta Y_j$  between the pixel value  $Y_j$  and the mean value  $\langle Y \rangle$  of the  $N$  detectors in the array is fitted by the  $n$ th order polynomial of  $\langle Y \rangle$  according to

$$\Delta Y_j = Y_j - \langle Y \rangle \approx \sum_{i=0}^n C_{ij} \langle Y \rangle^i = \Delta Y_j^c \quad j=1,2 \dots N \quad (5)$$

where  $\Delta Y_j^c$  is the estimated correction term. The corrected pixel value is denoted  $Y_j^c$  and is obtained as

$$Y_j^c = Y_j - \Delta Y_j^c \quad (6)$$

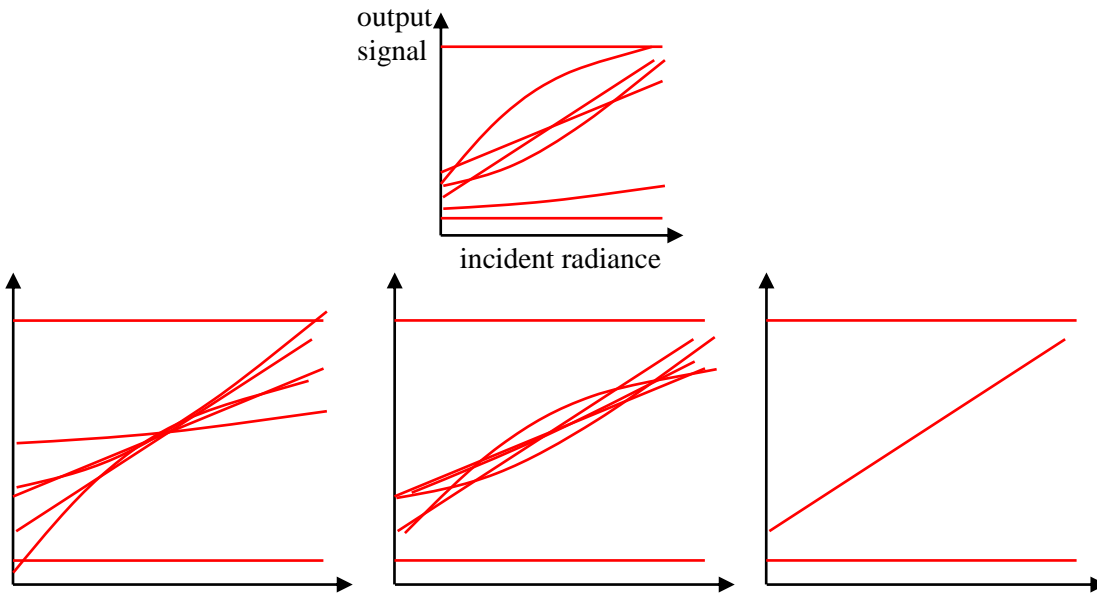
If a second degree polynomial is used, the approximation (5) turns into

$$\Delta Y_j^c = C_{0j} + C_{1j} \langle Y \rangle + C_{2j} \langle Y \rangle^2 \quad (7)$$

To estimate the coefficients  $C_{ij}$ , homogeneous radiating surfaces are registered at  $n+1$  radiation levels ( $=3$  for a 2<sup>nd</sup> degree polynomial). By eliminating  $\langle Y \rangle$  from (7), see Appendix A, the corrected value  $Y_j^c$  is obtained

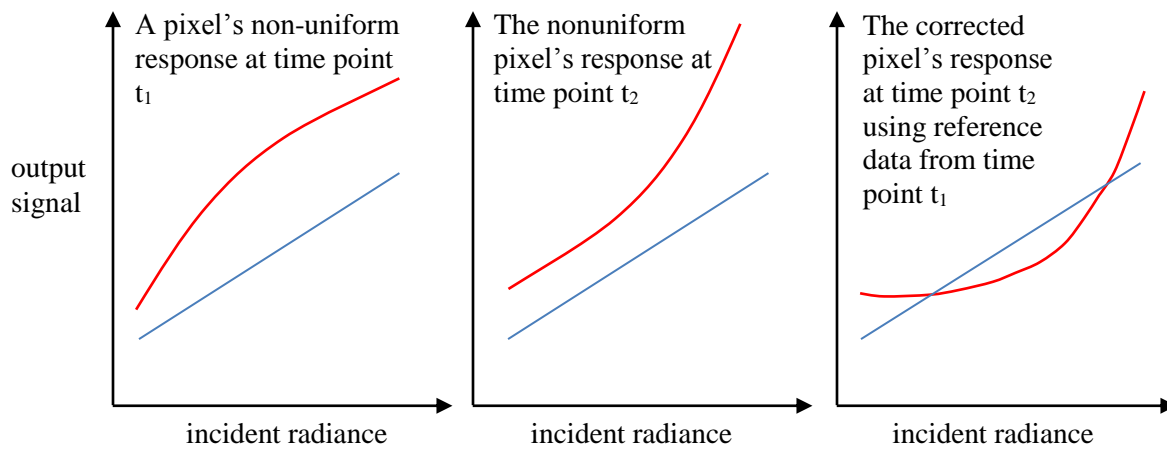
$$Y_j^c = -\frac{1 + C_{1j}}{2C_{2j}} \pm \sqrt{\frac{(1 + C_{1j})^2}{4C_{2j}^2} + \frac{Y_j - C_{0j}}{C_{2j}}} \quad (8)$$

The figure below shows schematic sketches of non-uniformity corrections with  $n$ th order polynomials.



**Figure:** non-uniformity corrections. Upper: uncorrected data; lower left: a 0<sup>th</sup> order correction; lower middle: a 1<sup>st</sup> order correction; lower right: a 2<sup>nd</sup> order correction

Ideally, the highest polynomial degree should give the best correction. One reason why this may not apply to all pixels is that the fixed pattern noise is not totally fixed but changes over time. A schematic example is shown in the figure below:



**Figure:** A non-uniform correction using a 2<sup>nd</sup> degree polynomial; the blue line marks an average pixel's response. The offset and the slope are corrected (right) but due to pixel drift the 2<sup>nd</sup> order correction has made the correction worse. In this case a better result had been obtained by a 1<sup>st</sup> order polynomial.

In B-1, B-2 and B-3 below, a rawdata image is compared with the results after a NUC using a 0<sup>th</sup>-degree (offset), a 1<sup>st</sup>-degree (linear) and a 2<sup>nd</sup> degree (square) polynomial fitting. By visual inspections, assign values on a (relative) 10 degree-scale in the table following B-3, where the raw image is given the value = 1, and a fictive “perfect” image (=an image where *all* pixels seem to have exact the same response) the value = 10.

**B-1.** Plot a scene image (raw data image).

**B-2.** Plot a NU corrected scene image. Compare the NUC results using offset, 1<sup>st</sup> degree and 2<sup>nd</sup> degree polynomial fitting. Use the Matlab command “imcontrast” to adjust the contrast in the images. Are there any differences?

Store the calibration coefficients and image data by proper names for later use in C, e.g. for a 0<sup>th</sup>-degree polynomial fitting: C0 (= the calibration coefficients) and OI\_c0 (the corrected image).

**B-3.** Repeat B-2 with identification and replacement of the dead pixels using median filtering. The difference between raw data pixels and median filtered pixels is used for identification of the dead pixels. A pixel is defined as a dead pixel if the difference rawdata pixel - median filtered pixel > K, where K is an arbitrary value that is varying between datasets and/or sensors. For the distributed dataset a K value = 1000 gives a good result. The value of the bad pixel is replaced with the corresponding pixel value in a median filtered image.

You do not need to store the data generated under B-3.

	<b>Point (1-10)</b>
<b>Image data</b>	<b>Visual inspection</b>
Raw	<b>1</b>
NUC 0 <sup>th</sup>	
NUC 1 <sup>st</sup>	
NUC 2 <sup>nd</sup>	
NUC 0 <sup>th</sup> + dp repl	
NUC 1 <sup>st</sup> + dp repl	
NUC 2 <sup>nd</sup> + dp repl	

**B-4.** For the distributed dataset a difference between rawdata pixels and median filtered pixels  $> 1000$  gives a good result. Repeat B-3 with the following bad pixel definition: rawdata pixels – median filtered pixels  $> 100000$ . For clarity, use raw data as input data. The amount of "salt and pepper" in the image now has become larger. On the other hand a too low value on the difference will cause a number of "normal" pixels to be identified and defined as bad pixels – and replaced. To identify and replace bad pixels is a compromise between minimization of the amount of salt and pepper in the image and retention of as much of the original information as possible.

**B-5.** Median filtering is the method used to identify the dead pixels in B-3. Can you find out other methods to identify dead pixels?

Hint: see e.g. example A-4: Compare the temporal noise between dead pixels and other pixels. Are there any differences ? In B-2 values of the coefficients were calculated. Are there any differences between dead pixels and other pixels?

## C. Image quality measures

A NUC will never be perfect and residual fixed pattern noise will therefore be added to the temporal noise:

$$\sigma_{tot}^2 = \sigma_{temp}^2 + \sigma_{spat}^2 \quad (9)$$

In B-1 – B-3 the quality of the NUC methods was evaluated by a visual inspection. However, if the amount of data is large, a substantial time may be needed to select the most optimal NUC methods by visual inspections. In addition, by a (subjective) visual inspection, it is not always obvious which NUC method is the most optimal and therefore the results will depend on the individual observer. Root-mean-square error (RMSE), UIQI or Q (universal image quality index) and roughness ( $\rho$ ) are four proposed measures of the goodness of the NUC to describe the quality of the correction with an objective number instead of a (subjective) visual inspection.

Collect the results you obtain in C-1 – C-3 in the table after C-3.

### C-1. RMSE

RMSE [4] measures the improvement by a NUC. RMSE is defined as the square root of the average (over the entire array and block of frames) of the square of the difference between the corrected (the “true”) pixel signal and the non-corrected pixel signal

$$RMSE = \sqrt{\frac{1}{N} \sum_{j=1}^N (\langle y_j \rangle - \langle x_j \rangle)^2} \quad ; [RMSE \geq 0] \quad (10)$$

where  $x_j$  denotes a pixel in the uncorrected image. With no improvement  $RMSE = 0$ .

**C-1a.** Calculate  $RMSE$  in an image from Scenedata, before and after NUC including replacement of bad pixels. Cut the upper right corner in the image before the calculation (~ the columns 351-384) which contains the cluster with bad pixels.

### C-2. UIQI

The universal image quality index UIQI (also denoted  $Q$ ) [5] measures the improvement by an NUC and compares the image before ( $x$ ) and after ( $y$ ) a NUC. UIQI is expressed in the following equation

$$UIQI = \frac{4 \cdot \sigma_{xy} \cdot \bar{x} \cdot \bar{y}}{(\sigma_x^2 + \sigma_y^2)(\bar{x}^2 + \bar{y}^2)} \quad ; [-1 \leq UIQI \leq 1] \quad (11)$$

where  $\bar{x}$ ,  $\bar{y}$ ,  $\sigma_x$  and  $\sigma_y$  are the mean and standard deviation of both images. With no improvement  $Q = 1$ . (18) can be rewritten

$$UIQI = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \cdot \frac{2\bar{x} \cdot \bar{y}}{\bar{x}^2 + \bar{y}^2} \cdot \frac{2\sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2} \quad (12)$$

The first component is the correlation coefficient between  $x$  and  $y$ . The second component measures the relative difference between  $x$  and  $y$ . The third component measures the contrast difference between the images.

**C-2a.** Calculate  $Q$  in an image from Scenedata, before and after NUC including replacement of bad pixels. Cut the upper right corner in the image before the calculation (~ the columns 351-384) which contains the cluster with bad pixels.

### C-3. Roughness $\rho$

The roughness  $\rho$  [6] for an image is defined

$$\rho(f) = \frac{\|h_1 * y\|_1 + \|h_2 * y\|_1}{\|y\|_1} \quad ; [\rho \geq 0] \quad (13)$$

where  $h_1$  is a horizontal mask  $[1, -1]$ ,  $h_2 = h_1^T$  a vertical mask,  $\|y\|_1$  is the  $L_1$  norm of the image = the sum of the magnitudes of all pixels. For a uniform image  $\rho = 0$ , and increases with the pixel-to-pixel variation in the image.

**C-3a.** Calculate  $\rho$  in an image from Scenedata, before and after NUC including replacement of bad pixels. Cut the upper right corner in the image before the calculation (~ the columns 351-384) which contains the cluster with bad pixels.

Image data	<i>RMSE</i>	<i>Q</i>	<i><math>\rho</math></i>
Raw	<b>0</b>	<b>1</b>	
NUC 0 <sup>th</sup>			
NUC 1 <sup>st</sup>			
NUC 2 <sup>nd</sup>			
NUC 0 <sup>th</sup> + dp repl			
NUC 1 <sup>st</sup> + dp repl			
NUC 2 <sup>nd</sup> + dp repl			

## D. Discussion section B and C

In practice a NUC based on a 1<sup>st</sup> degree polynomial is most common, followed by an offset correction. A 2<sup>nd</sup> degree polynomial fitting is not used as often, why ?

What is the basic difference between the three image quality metrics studied in this exercise, considering eq. (10), (11) and (13) ?

Which quality measure is in best agreement with your visual inspection in B?

The quality metrics RMSE and Q compare two images and return values of the change, instead of absolute values. Do these metrics tell if a change is positive or negative ? If (raw) scene data and reference data are registered at different time points they may not match each other and in this case a reference based NUC will distort the scene data. Can you find out a solution to this problem?

## E. Comparison with a standard camera

In the previous sections you have been looking at image data collected with a quite old infrared camera. The quite noisy data made it easier to see differences before and after the correction steps. In this section, you will compare with an infrared camera that is more representative of a standard infrared camera.

Data are found under 'Multiband' and are loaded into Matlab workspace as before: *load Refdata1*, etc.

**E-1.** Plot a scene image (raw data image) and make a visual comparison with a scene image from the first camera

**E-2.** Residual non-uniformity (RNU) is a measure used to quantify fixed pattern noise in image data from uniform surface radiators. It is defined

$$RNU = \frac{\sigma_{spat}}{DR} \cdot 100\%$$

where DR is the dynamic range, which is  $2^{14}$  for both cameras.

Use Refdata2 (both cameras). In a first step the data is averaged over time so that most of the temporal noise is eliminated. In a second step the standard deviation gives the spatial noise  $\sigma_{spat}$ . In camera 1 the columns 351-384 contains a cluster with bad pixels in the upper right corner which are excluded in the calculation of  $\sigma_{spat}$ . In camera 2 the first row contains a time code in the upper right corner which is excluded in the calculation of  $\sigma_{spat}$ . Using Matlab,  $\sigma_{spat}$  is calculated in the following way

*RefdataTempAve = mean(Refdata2,3);*

*Camera 1:  $\sigma_{spat} = std2(RefdataTempAver(:,1:350))$*

*Camera 2:  $\sigma_{spat} = std2(RefdataTempAver(2:512,:))$*

A RNU value below 0.1 % is a very good result. What can you say about the FPN quality of the first and the second camera ?

**E-3.** If time remains and you are interested, you can of course repeat the steps A through D also with the second camera.

*References*

- [1] Gerald C.Holst, *Electro-optical imaging system performance*, Fifth edition, JCD Publishing Co (2008)
- [2] M.Schultz, L.Caldwell. *Nonuniformity correction and correctability of infrared focal plane arrays*. Infrared physics and technology 36:p.763-777 (1995).
- [3] S.N. Torres, J.E. Pezoa, M.M. Hayat, *Scene-based nonuniformity correction for focal plane arrays by the method of the inverse covariance form*, Applied Optics, Vol.42, No.29 (2003)
- [4] Z.Wang, A.C.Bovik, *A Universal Image Quality Index*, IEEE Signal Processing Letters, Vol.9, No.3 (2002)
- [5] M.M. Hayat, S.N. Torres, E. Armstrong, S.C. Cain, B. Yasuda, *Statistical algorithm for nonuniformity correction in focal-plane arrays*, Applied Optics, Vol.38, No.8 (1999)

## Appendix A: Schultz approximation

**Schultz (polynomial fitting):**

$$\Delta Y_j = Y_j - \langle Y \rangle \approx \sum_{i=0}^n C_{ij} \langle Y \rangle^i = \Delta Y_j^c;$$

Ideally only temporal noise will remain after the correction and  $Y_j^c \approx \langle Y \rangle$  for all pixels  $j$

Offset:

$$Y_j^c = Y_j - C_0 \tag{A1}$$

1<sup>st</sup> degree:

$$Y_j - Y_j^c = C_{0j} + C_{1j} \cdot Y_j^c$$

$$Y_j - C_{0j} = (1 + C_{1j}) \cdot Y_j^c$$

$$Y_j^c = \frac{Y_j - C_{0j}}{1 + C_{1j}} \tag{A2}$$

2<sup>nd</sup> degree:

$$Y_j - Y_j^c = C_{0j} + C_{1j} \cdot Y_j^c + C_{2j} \cdot (Y_j^c)^2$$

$$Y_j - C_{0j} = (1 + C_{1j}) \cdot Y_j^c + C_{2j} \cdot (Y_j^c)^2$$

$$(Y_j^c)^2 + \frac{(1 + C_{1j})}{C_{2j}} (Y_j^c) - \frac{(Y_j - C_{0j})}{C_{2j}} = 0$$

$$Y_j^c = -\frac{1 + C_{1j}}{2C_{2j}} \pm \sqrt{\left(\frac{1 + C_{1j}}{2C_{2j}}\right)^2 + \frac{Y_j - C_{0j}}{C_{2j}}} = -\frac{1 + C_{1j}}{2C_{2j}} \pm \sqrt{\frac{(1 + C_{1j})^2}{4C_{2j}^2} + \frac{Y_j - C_{0j}}{C_{2j}}} \tag{A3}$$