Robust autonomous detection of the defective pixels in detectors using a probabilistic technique

Siddhartha Ghosh,1,* Dirk Froebrich,2 and Alex Freitas1
1Computing Laboratory, University of Kent, Canterbury, CT2 7NF, UK
2Centre for Astrophysics and Planetary Science, School of Physical Sciences, University of Kent, Canterbury, CT2 7NH, UK
*Corresponding author: sg55@kent.ac.uk

Received 14 May 2008; revised 22 October 2008; accepted 4 September 2008; posted 22 October 2008 (Doc. ID 96233); published 20 December 2008

Detection of defective pixels in solid-state detectors/sensor arrays has received limited research attention. Few approaches currently exist for detecting the defective pixels using real images captured with cameras equipped with such detectors, and they are ad hoc and limited in their applicability. In this paper, we present a probabilistic novel integrated technique for autonomously detecting the defective pixels in image sensor arrays. It can be applied to images containing rich scene information, captured with any digital camera equipped with a solid-state detector, to detect different kinds of defective pixels in the detector. We apply our technique to the detection of various defective pixels in an experimental camera equipped with a charge coupled device (CCD) array and two out of the four HgCdTe detectors of the UKIRT’s wide field camera (WFCAM) used for infrared (IR) astronomy [Astron. Astrophys. 467, 777–784 (2007)]. © 2008 Optical Society of America

OCIS codes: 040.1520, 040.3060, 040.1240, 100.5010, 100.2000, 100.2550.

1. Introduction
In the past few decades, digital imaging has been revolutionized by the development of increasingly sophisticated solid-state detectors/image sensor arrays, which are able to take high precise and sharp images. This progress has been a ramification of rapid strides made in improvement of semiconductor technology that has resulted in more precise and accurate detectors, along with rapid increase in detector resolution, making it possible to pack greater number of pixels on the same semiconductor chip. The two main strands of progress in semiconductor technology have led to the development of better CCD and complimentary metal oxide semiconductor (CMOS) detectors for visible imaging [1,2] and detectors for imaging in the infrared (IR) spectrum [3–5]. Detectors used for imaging in the visible and the IR spectrum have different architectures due to the different nature of the light spectrum they peer at [6]. However, what is intrinsically common to both these detectors is the idea of a pixel—a single photo sensitive unit, which when exposed to light, acts as a potential well by attracting electrons produced by photons absorbed in the material, that is subsequently amplified and digitized. The end result of this process is a digital image, which is composed of the measurements of all the pixels in a detector. As the pixels of a detector form the core components of the imaging process, their proper functioning has a significant impact on the quality of the digitized image that is formed. Almost all detectors, have a certain proportion of pixels that are defective and are commonly referred to as “bad” pixels [7]. These pixels manifest due to defects in design of the semiconductor chip or manufacturing errors [6,8], and they can also develop during operation of the detectors, particularly when the operating conditions are harsh and the detectors are operated for long periods of time. We describe commonly occurring defective pixel types in detectors in Section 2. It is important to
detect and eliminate the defective pixels in an image sensor array so that their spurious measurements do not affect the quality of the final images obtained using the array, which in turn can have a profound effect on the subsequent interpretation of those images [7]. For astronomers, for instance, it is important to be able to discriminate a pixel that constantly reports a high value from a bright star or a pixel that constantly reports no value from the dark background. In Section 3 we see that, although there are many simple existing ways of identifying defective pixels using calibration images, very little work has been done in the way of autonomously detecting defective pixels in image sensor arrays using images with rich scene information.

In [9] we proposed a novel integrated statistical technique for autonomously diagnosing the defective pixels in the CCD array of an experimental digital camera. We showed how this data driven probabilistic technique could robustly identify the defective pixels in the CCD. The most important feature that distinguishes our novel technique is that it can be applied to raw images containing rich information captured with any image sensor array, and it can identify the defective pixels without requiring any calibration images. Our integrated technique consists of three constituent steps: for a random selection of images, we compute pairwise summed difference estimate (PSDE) between pixels in a block and model their density by a Gaussian mixture model (GMM) [10]; we then classify additional instances of such PSDEs based on applying the Bayes' theorem [11]; finally, we use the classifications of all the PSDEs of every pixel in a simple rank-based scheme to identify the most aberrant pixels amongst those n pixels. We provide a thorough insight into how the constituent steps are linked together and explain the contribution of each step to the overall goal of detecting defective pixels in Section 4. It is important to mention that, although each of the constituent steps represents a powerful statistical technique, and they are widely used within the pattern recognition community, our effort in combining them to conjure a diagnostic technique, to the best of our knowledge, is novel.

As we did not have any a priori knowledge of the faulty pixels in the CCD array used in our experiments in [9], we had proposed a simple “ground-truth” technique that enabled us to identify the different defective pixels in the CCD. This ground-truth technique is a simple statistical analysis, which is based on assumptions of normality of the measurements of pixels in calibration flatfield images. However, the results presented in [9] were not accurate, as during the acquisition of flatfield calibration images for the ground-truth method; we had not exercised strict control over the experimental factors, such as preventing stray or ambient lighting from being incident on the CCD and the temperature of the camera. Consequently, the images used for the ground-truth technique were not of good quality, and we did not obtain conclusive evidence of the faulty pixels in our CCD. Without having a robust ground truth, it becomes very difficult to thoroughly evaluate the performance of our integrated technique. Hence, in this paper, we obtain better and more accurate flatfield calibration images by augmenting our experimental camera with some special equipment and taking images in a precise laboratory environment, which we describe in the experimental setup description in Section 5. The new setup enables us to precisely control the temperature of the camera and its exposure setting while acquiring the flatfield images to which the ground truth is then applied to discover the faulty pixels in the CCD. We evaluate the performance of our integrated technique at detecting the same defective pixels in real images containing rich scene information and compare our technique's performance to the ground truths'.

The improvement in analysis and experimental procedures resulting in more thorough and substantially different results as compared to our earlier work is the first major contribution of this paper.

The second major contribution of this paper is to extend our technique for diagnosing the defective pixels in the detector arrays of an astronomical camera; we show the performance of our technique at the detection of the defective pixels in the near infrared (NIR) detector arrays of the state of the art WFCAM camera, an IR widefield camera for the UK Infrared Telescope on Mauna Kea (UKIRT) that is currently the most capable IR imaging survey instrument in the world [12]. As far as we are aware, our work is the first of its kind at applying an unsupervised statistical machine learning technique to autonomously identify the defective pixels in astronomical detectors. Astronomers are always interested in high-precision photometry using images that are devoid of cosmetic errors in the imaging system, defective pixels being examples of cosmetic systematic errors [7]. We show the influence of our defective pixel detection technique on some of the photometry done on the images we use for our analysis.

To put things into perspective, this paper consists of two salient parts: The first part is composed of a thorough investigation of our technique at the task of detecting the defective pixels in the CCD of our experimental camera. Once we thoroughly evaluate our technique, we describe the second part, which relates to assessing our technique's performance in diagnosing defective pixels in the detectors of a scientific-grade astronomical camera. The first part effectively serves as a precursor to the second part where we actually demonstrate the utility of our diagnostic technique.

2. Defective Pixels in Image Sensor Arrays

As mentioned earlier, we endeavour to provide a broad but thorough description of the different types of defective pixels that can manifest in solid-state image sensor arrays. To achieve this, we have collated the disparate descriptions of defective pixels,
explained in different contexts and undertake a more thorough treatment of this problem. Gullixson provides a good treatment of this problem in the context of detectors used in astronomy [7]. He mentions that every imaging system suffers from what he terms as “spatial systematic errors”, which are not random in nature, but have some pattern or trend in their incidence and classifies such spatial systematic errors into three different categories: cosmetic spatial systematic errors that result from defects in the design or manufacture of the detector; multiplicative spatial systematic errors that result in the multiplication (usually by a fraction) of the signal in an area in the image and consist of optical transmission variations and variations in the quantum efficiency across the CCD; additive spatial systematic errors that result in the addition or subtraction of signal from an area in the image. It includes bias structure, dark emission, charge skimming, telescope background emission, etc. From Gullixson’s classification it is possible to draw a parallel between Gullixson’s cosmetic spatial systematic errors and our notion of defective pixels, and therefore we do not discuss errors that fall in the other two categories. The idea of defective pixels in CMOS or active pixel sensor arrays was discussed in [13] and defective pixels in CCDs used for x-ray imaging [14]. Overall, defective pixels or cosmetic spatial systematic errors in imaging systems are usually one of the following types:

- **Dead Pixels:**
  Dead pixels are completely insensitive and they do not respond to incident light. Let $I_{ij}$ be a dead pixel in an image $I$. Its value will always be zero, i.e., $I_{ij} = 0$. These pixels appear as black or dark spots in images. Dark pixels manifest due to defects in the underlying hardware: in a CCD or CMOS array, a photodiode may be shorted and, therefore, does not integrate charge; in detectors where pixels are addressed directly such as CMOS and IR detectors, the readout amplifier of a pixel may have malfunctioned. In IR detectors, breakage of the indium bumps between the detector and the readout arrays can lead to pixels becoming dead.

- **Stuck Pixels:**
  Stuck pixels exhibit constantly “high” values, where the actual value is determined by the maximum possible value after digitalization. Hence, in an image $I$, a stuck pixel will exhibit a value $I_{ij} = 2^b - 1$, where $b$ is the number of bits used by the analog to digital (A/D) converter for quantization. For instance, for an 8 bit A/D, the maximum pixel value is $2^8 = 256$, a stuck pixel thereby always showing 255. In contrast to dead pixels, stuck pixels appear as bright/white spots, and they are caused by fabrication errors that result in odd pixels getting saturated. They may also manifest due to failure of the output transistor, which may get stuck and, as in the case of dead pixels, this problem is more prevalent in detector arrays where pixels are $XY$ addressed, as in CMOS and IR detectors.

- **Hot Pixels:**
  Hot pixels are pixels with unusually large dark current due to higher leakage of charge; they are persistent and they appear at fixed spatial locations in a detector [15]. In such pixels, the extra dark current adds a small offset to the pixel’s measured charge, which increases with the exposure setting of the camera, thereby limiting the dynamic range [16]. Since temperature has a direct effect on the production of dark current within a detector, the number of hot pixels increases with increasing temperature and, to mitigate such pixels, cooling the imaging system is necessary. Hot pixels appear as bright spots in uniformly illuminated flatfields and regular images. It has been discovered that hot pixels manifest due to cosmic rays being incident on the detector [17]. When cosmic rays strike a CCD, there is interaction between their heavy nuclei and the silicon (Si) nuclei resulting in bulk displacement damages which causes new Si–SiO$_2$ interface states.

- **Abnormally Sensitive Pixels:**
  For high-grade imaging, it is important for a detector to be uniform. In an ideal detector, upon exposure to a uniformly illuminated flatfield, the response of all pixels should be almost identical, with very little variation across the detector. However, Mackay [18] mentions in the context of a CCD being used for astronomy—“the only uniform CCD is a dead CCD.” What he means is that all detectors are nonuniform and have pixels that are either more sensitive or less sensitive than average. In some situations, such pixels exhibit sensitivity that is abnormally greater (hypersensitive) or abnormally lesser (hyposensitive) than the mean intensity. In detectors, such pixels may also manifest due to the nonuniformity in the readout and the digitization electronics of individual pixels.

- **Column Defects and Traps**
  It is possible for all the pixels in a single column or multiple columns in a detector to be bad. Such columns where all the pixels along the column are either dead, stuck, or abnormally sensitive are called bad columns. Detectors often exhibit bad columns, which are produced due to fabrication faults during the detector manufacturing phase [6]. CCDs, in particular, can have pixels which are commonly called “traps”. Such pixels act as spurious potential pockets by trapping charges and create the problem of deferred charge where charges trapped in the pixel are released at a later time. This in turn leads to a reduction of charge transferred from a certain number of pixels in the same column as the trap [7,19].

It is imperative to make a distinction between the two different but related problems: our task of detecting defective pixels in the image sensor arrays and the other related but subtly different problem of detecting and filtering noisy pixels in single images. Filtering noisy pixels is an important problem within image processing research, where extensive effort has gone into mitigating the effects of noise in digital
images in the form of the development of highly sophisticated noise filters. A comprehensive treatment of noise in digital images and their mitigation by noise filters can be found in [20–22]. We are primarily interested in only those defective pixels that manifest in image sensor arrays due to systematic faults that are either introduced during the detector’s manufacturing process or may arise during the operation of the imaging system. As they are hardware faults, unlike noisy pixels, defective pixels persist from one image to another at the same locations in the images captured with a detector, i.e., they exhibit spatial persistence [7].

3. Existing Techniques for Defective Pixel Detection

Here we discuss the existing techniques and methods for detecting defective pixels in image sensor arrays. Such existing techniques fall under two categories based on the kind of images used for defective pixel detection: (1) techniques that make use of calibration images such as dark frames and flatfields to identify defective pixels; (2) techniques that use real images captured by a detector.

A. Techniques Based on Calibration Images

In astronomical imaging, calibration images are predominantly used during the “data pipeline” stages, where the raw images captured by a detector are subject to several processing stages before the final science data are obtained. Depending on the detector type, calibration images such as flatfields and dark frames are used during the numerous processing stages of the data reduction pipeline, which reduce the various errors (as per Gullixson’s terminology). A common practice during such pipeline stages is to use a defective pixel map that contains the locations of all the defective pixels in the detector to remove the effects of cosmic errors or defective pixels. There are several ways of creating such defective pixel masks. A common practice is to use calibration images, such as dark frames and flatfields, to identify the defective pixels and create the mask. Dark frames can help identify hot pixels, which report intensity values much higher than other pixels and appear brighter than the rest. The use of dark frames in identifying defective pixels can be found explained in [23]. When such calibration images are used to generate the defective pixel maps, simple thresholds are used to discriminate aberrant pixels from normally functioning pixels. Sabbey et al. and Irwin et al., for instance, explain how a master flatfield can be used to detect defective pixels in the detector [24, 25] and, as such, their routines are incorporated in their infrared data reduction software package (IRDR) [24]. The flatfield images are used to produce a gain map per detector by dividing the flatfields by the modes of the detector. Bad pixels are automatically identified in the gain maps by looking for outliers: pixels that are >5σ from the median in 15 × 15 pixel blocks or pixels with extremely low or high sensitivity (>30% from the median gain). We believe that such a method can identify the most conspicuously defective pixels but may fail to identify the more subtle defects. To detect defective pixels in the detectors of the NICMOS astronomical camera, a median image is created from a set of dark frames, a ring median filter was passed over the median image, and the resulting image subtracted from the original median. A final image in units of sigma (from the mean) is created for the purpose of defective pixel clipping, whereby the defective pixel thresholds are set to 5σ. For detecting the defective pixels in the optical CCD of the Chandra astronomical observatory, Cresitello-Dittmar et al. propose a simple method for detecting “warm pixel”—pixels with increased dark current (akin to hot pixels discussed earlier). They compute a simple threshold for the underlying dark current in the CCD, and a pixel whose dark current level is greater than the threshold is considered to be “warm” [26]. Jin et al. proposed a simple soft-test/repair model to identify defective pixels in CCDs that relies on the use of special images taken under special lighting to identify different defective pixels [27]. Their simple test mechanism is based on the premise that every such image taken under a particular lighting condition is expected to follow a certain distribution. Hence, defective pixels can be discriminated from healthy pixels as being the outliers in the distribution of the pixel values. By their own admission their test requires a special experimental setup—for instance, a very high quality light source is required to detect such pixels, which limits their techniques’ widespread usage. In addition, their use of simple thresholds is not very robust, as only the most aberrant pixels can be detected. Lopez-Alonso and Alda applied Principal Component Analysis, a widely used statistical analysis technique, to identify defective pixels in both CCD and IR in [28]. The authors apply principal component analysis to a set of uniformly illuminated images of a stable scene, akin to a flatfield, to statistically characterize the pixels. Although PCA is a powerful data dimensionality reduction technique and, as the authors show, it yields good results at the defective pixel detection task, its use in identifying defective pixels is fundamentally limited by the assumption that the data, which in their case is a calibration image, need to be similarly distributed. Images of real scenes do not exhibit this property, and hence PCA cannot be applied to detecting defective pixels in real images, which exhibit great variations in how they are distributed. The overall problem with using calibration images to identify defective pixels is that these images need to be very high fidelity if robust identification of defective pixels is based on using these images. In addition, a significant amount of time and effort needs to be invested in acquiring such images. This is partly justified for astronomical observations, where such images have other utility in the overall pipeline. Nonetheless, acquiring good quality flatfields and dark frames can be a problem.
B. Techniques Based on Real Images

There has been little work done in trying to detect defective pixels in image sensor arrays using images with rich scene information. Jin et al. proposed a technique for online detection of defective pixels in CCDs used for X-ray imaging in [14]. Their technique is based on identifying defective pixels by applying a $3 \times 3$ mean filter centered on every pixel in raw images stored in the frame memory of an X-ray imaging system. They applied a simple statistical test for locating defective pixels that computes the difference between every pixel’s value and the average value of its eight neighboring pixels. A pixel is deemed to be defective if this difference is higher than a preset threshold. Slightly different variations of this simple statistical test enable them to identify three kinds of defective pixels: dead, stuck, and hypersensitive pixels. We believe that such simple techniques are effective for a quick and online defective pixel test-repair system, where the paramount concern is speed of X-ray processing, which the authors address. However, the actual diagnostic technique is too simplistic and will fail to capture a wide range of defects except for the most obvious ones. In addition, by the authors’ own admission, their technique cannot detect defective pixels clusters. It also remains to be seen as to how effective such techniques are when applied to detectors having different architectures, where the nature of some pixels defects can vary.

Chapman et al. proposed a simple Bayesian algorithm for online identification of faults in image sensor arrays in [8]. Their notion of defects in image sensor arrays is based on the following premise: in an image $I$, let $y_{ij}$ be the output of a pixel at location $i,j$, such that $y_{ij} = mx_{ij} + b$, where $m$ is the gain or sensitivity of the pixels and $b$ is the bias. Then they introduce the concept of a defect level, where depending on the values of $m$ and $b$, there can be different types of defects that are categorized into various levels from Level 1 to Level 5. For instance, Level 1 includes all kinds of stuck defects; Level 5 includes all defects for gains ranging from $m = 0$ to $m = 1.5$, amongst others. The entire summary of defect models can be found in [29]. They artificially corrupt pixels in the array subject to such defect levels and propose a simple Bayesian algorithm for classifying every pixel into a particular defect type. In [29,30] this simple Bayesian algorithm is refined, whereby during the process of classification of every pixel an estimate of its true defect type is made by interpolating its value from its neighbors. However, a key aspect to their research is that all defects are simulated and no effort has been made to detect real defects in detectors. In addition, a fundamental flaw in their model is the assumption that the interpolated value of a pixel using values of its neighbors is a reliable estimate. This works well when there are no clusters of defective pixels and, by the authors’ own admission, their technique fails in this regard. We also believe that the authors make matters more complex by introducing their artificially constructed defect levels. Classifying every pixel into a defect category hugely increases the computational cost, particularly in large arrays, as every single pixel has to be checked for membership in each of the levels. We believe that by trying to classify individual pixels we add no value with increased complexity, if all that we are interested in is identifying defective pixels, which by definition are far less in number as compared to normally functioning pixels in modern day detectors. There is also no evidence to suggest that they have applied their algorithm to images captured with IR detectors. Yap-Peng and Acharya proposed an algorithm based on the sequential probability ratio test (SPRT) for defective pixel detection using regular images [31]. They apply their method toward detecting three different types of faulty pixels: dead, stuck, and abnormally sensitive pixels. From their published results, it is evident that their approach is prone to false-positives even for such simple cases of faulty pixels. Their method is not very flexible as it is based on the assumption that a defective pixel does not have any other defective neighbors, an assumption that grossly limits the performance of their technique. In addition, to discriminate different defects, they presuppose the existence of well-specified probability density functions for each of the defect types, which makes their method reliant on a priori knowledge and not very easy to use when such information is unavailable. This in turn makes this techniques susceptible to errors in diagnosis, if applied to other detectors.

Rodgers and Riess devised an artificial neural network (ANN) for the automatic detection and classification of CCD defects in their CCDs used for the MACHO project, which was a part of an astronomical search for dark matter [32]. Their ANN system was composed of two separate feedforward types of neural networks: a detection network and a classification network. The objective of the detection network was to identify only two very specific defects in the CCDs: traps and scratches or foreign material on the surface of the CCD. For the detection network to be able to identify such defects, it was trained using the features computed from a portion of a sample image and, unless the training data can encompass a rich and diverse set of features, the learning algorithm, in this case an ANN can only perform specific tasks—in this case only identify the two defects. They did not detect other types of defects such as hot or stuck pixels.

What emerges from the previous discussion is that very little work has been done in the way of devising a pattern recognition technique that can autonomously detect the defective pixels in image sensory arrays, without requiring any calibration data in the form of images or a priori knowledge of the defects in an array and requiring minimal use of ad hoc thresholds for discriminating defective pixels from healthy ones. This sets the context for our novel integrated technique that can autonomously identify the defective pixels in any detector using the raw
images captured with the detector; it can operate in an almost unsupervised setting and is robust in its performance.

4. Novel Integrated Technique

A. Modeling the Difference Estimates

Our novel integrated technique is applied to autonomously detect the defective pixels amongst $n$ pixels that lie within a region or form a block of $n$ pixels within an image. The choice of such a “local” technique is motivated by the fact that in an image containing rich scene information, pixels in one part of the image can report intensity values that might be significantly different from the intensity values of pixels in another different region of the image. Hence, by only focusing on small regions it becomes possible to identify the defective pixels that exhibit anomalous readings, since all normally functioning pixels within a small region report similar intensity values. The idea of using analysis pixels within small regions of images is related to idea of a “local neighborhood” of a pixel as explained in [22]. Figure 1 shows the different ways in which we select $n$ pixels for analysis. For the purposes of our integrated technique, a block can take on any shape such as a square or a rectangular block, but in effect, it is just a selection of $n$ pixels from a small region within the images as shown in Figs. 1(a)–1(c). In addition, our technique can also be applied to a block of pixels centered on a particular pixel as shown in Fig. 1(d) (e.g., a $3 \times 3$ block of $n = 9$ pixels centered on a particular pixel). In Sections 6 and 7, we show how both these configurations are used.

Consider a small region or window within an image containing $n$ pixels. In $p$ images, each of the $n$ pixels will report an intensity value in each of the $p$ images, and we can form a $n \times p$ matrix $M$. As the first step, the measurements of all $n$ pixels in every image is normalized as per Eq. (1), where $\text{std}(M_j)$ is the standard deviation of the measurements of all the $n$ pixels in image $j$. It is important that the sign of each $M_{ij}$ be preserved as a large positive value of $M_{ij}$—indicative of a hypersensitive measurement, which is very different to a large negative $M_{ij}$ = indicative of a hyposensitive measurement:

$$M_{ij} = \frac{M_{ij} - \text{mean}(M_j)}{\text{std}(M_j)}, \quad \text{where } i = 1, \ldots, n \text{ and } j = 1, \ldots, p.$$  

Every $1 \times p$ row in $M$, the row vector $M_a$ (where $a = 1, \ldots, n$), represents the measurements of a single pixel in $p$ images. For every row vector $M_a$, we then compute the difference between $M_a$ and all other $M_b$ (where $b = 1, \ldots, n$ and $a \neq b$). A single pairwise summed difference estimate or PSDE, $\Delta$, between the vectors of intensity values of two pixels is computed as per Eq. (2):

$$\Delta_{ab} = \sum_{j=1}^{p} |M_{aj} - M_{bj}|.$$  

The computation of PSDEs as per Eq. (2) in effect dispels with the need for the normalized pixel measurements as per Eq. (1) to be within a specific range; if a pixel’s measurement was not normalized, it would only scale the corresponding PSDE estimates.

For $n$ pixels within a block in an image, there will be $(n \times (n - 1))/2$ unique PSDEs if we discard the PSDE of a pixel with itself. The PSDE of a pixel with itself will always be zero and is discarded. Following the methodology shown in Fig. 2, given a random selection of images, we organize the images into image data sets $IS_1, \ldots, IS_N$, where each set $IS_i$ is composed of $p$ images each. To discover the faulty pixels amongst the $n$ pixels within a small region of the images, we compute the $(n \times (n - 1))/2$ PSDEs in each of the $N$ image data sets. Computing PSDEs between $n$ pixels is a transformation that leads to standardization of the density of the corresponding density of difference estimates. It also increases the number of points for density estimation, as opposed to just using individual pixel intensity values. Previously, Odom and Milanfar have shown how the
histogram of pixel differences are remarkably similar across different images, which is in stark contrast to the histogram of mere pixel intensity values, particularly for noisy images [33,34]. The parameter \( d \) in Fig. 2 refers to the number of image data sets used for the GMM step. Figure 2 explains the methodology of our integrated technique in the case where \( d = 1 \), i.e., we use the PSDEs from just one out of the \( N \) image data sets and model their density by means of a \( K \) component GMM; the total number of PSDEs are \( d \times (n \times (n-1))/2 \). The use of a mixture model in modeling pixel difference statistics for an image was shown by Odom and Milanfar in [33,34]. To model the pixel differences computed for single images, they proposed a two-component mixture model comprised of a mixture of a Gaussian component and a Laplacian or a generalized Gaussian distribution. The authors by means of chi-squared tests for goodness of fit, empirically showed that their two-component mixture model was a good fit to the density of the difference estimates. As explained by Xu and Jordan in [35], mixture models offer the flexibility of a nonparametric approach and, at the same time, retain some of the advantages of a parametric approach such as computational tractability. Hence, density estimation by mixture models provides a convenient method of density estimation that lies between parametric and kernel density estimators. Gaussian components are a popular choice for the individual components of a mixture model due to their computational convenience [36]. Since initially we do not have any \( a \) priori knowledge of how the PSDEs are distributed, modeling the density of the PSDEs by a GMM is akin to a model-based clustering framework, where the components of the mixture model then correspond to clusters to be imposed on the data [36,37].

Let \( \Delta_i \) represent a single PSDE between two sensors \( a \) and \( b \), which constitutes a single input to the GMM modeling process. Then, in the GMM formulation \( \Delta_i \) can be represented as \( \sum_{k=1}^{K} \pi_k P_k(\Delta_i|\theta_k) \), where \( K \) is the number of GMM components or clusters, \( \pi_k \) is the weighting factor of the \( k \)th component \( \pi_k \geq 0 \) and \( \sum_{k=1}^{K} \pi_k = 1 \) and \( \theta_k \) represents the complete parameter vector [10] of the \( K \)th component. The single PSDE, \( \Delta_i \), is a one-dimensional numeric value, and the density of all such PSDEs, consequently, is a univariate density. Therefore, for the purposes of our technique, the individual Gaussian components of the GMM are represented as univariate normal distributions, and \( P_k(\Delta_i|\theta_k) \) is the likelihood of the PSDE in the \( k \)th GMM component \((k = 1, \ldots, K)\) as per Eq. (3), where \( \mu_k \) and \( \sigma_k \) are the mean and the standard deviation of the \( k \)th GMM component:

\[
P_k(\Delta_i|\theta_k) = \frac{1}{\sigma_k \sqrt{2\pi}} e^{-\frac{(\Delta_i - \mu_k)^2}{2\sigma_k^2}}.
\]

Initially, the parameters of the \( K \)th component of the GMM are unknown. We use the standard expectation maximization (EM) algorithm to estimate the parameters of the GMM that models the density of PSDEs [38,39]. The EM algorithm maximizes the likelihood shown in Eq. (3), and the inputs to the EM algorithm are the PSDEs from \( d \) image data sets. As mentioned previously, there will be \( d \times (n \times (n-1))/2 \) PSDEs, and hence the input vector of PSDEs is \( \Delta_i \) for \( i = 1, \ldots, d \times (n \times (n-1))/2 \). In the EM formalism, the “complete” data \( y_i \) is subsequently represented as \( y_i = (\Delta_i, z_i) \), where \( z_i \) constitutes the “missing” data such that \( z_i = (z_{i1}, \ldots, z_{ik}) \); \( z_{ik} = 1 \) if \( \Delta_i \) belongs to cluster \( k \) and 0 otherwise. The \( z_{i1}, \ldots, z_{ik} \) are independently and identically distributed according to a multinomial distribution consisting of one draw on \( K \) categories with probabilities \( \pi_1, \ldots, \pi_K \). The resulting complete data log-likelihood is given by Eq. (4):

\[
L(\theta_k, \pi_k, z_{ik} | \Delta_i) = \sum_{i=1}^{(n^2-n)/2} \sum_{k=1}^{K} z_{ik} \log \pi_k P(\Delta_i|\theta_k).
\]

The EM algorithm alternates between an expectation (E) step in which the \( z_{ik} \) are estimated from the data and current parameter values and a maximization (M) step in which the log-likelihood given by Eq. (3) is maximized with respect to the parameters [40]. The input vector of PSDEs \( \Delta_i \), where \( i = 1, \ldots, d \times (n \times (n-1))/2 \), consists of PSDEs that have been estimated sequentially between sensor pairs. Before applying the EM algorithm, the vector \( \Delta_i \) is shuffled to randomize \( \Delta_i \) and avoid any positional bias due to computing pairwise difference estimates between adjacent pixels. The number of mixture components of the Gaussian mixture strongly influences how well the GMM models the distribution of PSDEs, which in turn may strongly influence the prediction of the detection of the pixels based on the estimated model. To determine the correct number of components, we use the Bayesian information criteria (BIC) to determine the GMM that best models a distribution of PSDEs. When there is a choice between multiple GMMs characterized by a different number of mixture components, BIC is known to perform well at determining the best fitting model as explained in [37,41]. Although several other criteria for model selection have been proposed, as explained by Biernacki et al. [42] and references therein, there is very little difference between many of them. Of particular interest to us is the Gaussian component that models the cluster of PSDEs that are toward the high end of the scale of PSDEs, and we refer to this component as the High component. This component is of particular importance because it models the group of PSDEs that are most aberrant. If a pixel is faulty, it will consistently produce PSDEs that belong to such a High component. The group of \( n = 30 \) pixels in Fig. 3(a) did not have any faulty pixel amongst them. On the contrary, in Fig. 3(c) there was a single hot pixel amongst the group of pixels. What can therefore be inferred from comparing Figs. 3(a) and 3(c) is that
that one of the two pixels $a$ or $b$ or even both of them are faulty. Hence, the probabilities of every PSDE in multiple components of the GMM over multiple image data sets is a key indicator of the aberrance of the pixels in the pixel pair associated with those PSDEs. Consequently, it is possible to classify a PSDE into one of the GMM components based on the probability of its membership in the components of the GMM, estimated over multiple image data sets, and we estimate such probabilities for all the PSDEs by applying the Bayes theorem. Furthermore, once we classify all the PSDEs of a single pixel, we then observe the proportion of the PSDEs that have been classified in the High component, as we can deduce whether or not the pixel is faulty; it will be faulty if a large proportion of its difference estimates are classified in the High component. Thus, in effect, this process of classifying PSDEs plays a key role in the identification of faulty pixels.

Let the components of the GMM be represented as $C_1,...,C_K$, where $C_K$ is the High component. These components represent the classes of PSDEs where each class is modeled as a Gaussian represented by the parameter vector $\theta_k = (\mathbf{x}_k, \mu_k, \sigma_k)$, where $k = 1, ..., K$. Our objective is to classify the PSDEs from $N - d$ image data sets into these classes. Let $j = d + 1, ..., N$ indicate the indices of the image data sets to be used for this classification step. The corresponding image data sets are $\mathbf{I}_{S_1}, ..., \mathbf{I}_{S_N}$. Each of these image data sets will have $(n \times (n - 1))/2$ PSDEs that have been computed between $n$ pixels within a block. For the purpose of classification, let $\Delta^j_i$ represent a single PSDE estimates between two pixels $a$ and $b$ in image data set $j$, where $j = (d + 1, ..., N)$ and $i = 1, ..., n \times (n - 1)/2$. For the complete $N - d$ sets, we can form a vector of inputs $\hat{\mathbf{d}}_i = (\Delta_i^{d+1}, \Delta_i^{d+2}, ..., \Delta_i^N)$. To classify this vector set of inputs we apply the Bayes theorem as shown in Eq. (5):

$$
P(C_k|\hat{\mathbf{d}}_i) = \frac{P(C_k) \times P(\hat{\mathbf{d}}_i|C_k)}{\sum_{k=1}^{K} P(C_k) \times P(\hat{\mathbf{d}}_i|C_k)}.
$$

The vector $\hat{\mathbf{d}}_i$ is classified into the class $C_k$ if its posterior probability $P(C_k|\hat{\mathbf{d}}_i)$ is the highest for that class. This is also called the maximum a posteriori (MAP) decision rule that minimizes the probability of error or error rate under zero-one loss [43] and is shown in Eq. (6):

$$
\text{classify}(\hat{\mathbf{d}}_i) = \arg \max_k P(C_k|\hat{\mathbf{d}}_i).
$$

It is important to note that while applying the Bayes theorem to vector of inputs $\hat{\mathbf{d}}_i$ as shown in Eq. (5), we assume that every $\Delta^j_i$ is independent of all the other $\Delta^j_i$'s, where $j = d + 1, ..., N$, $j \neq k$, i.e., a PSDE between two pixels in one image data set is independent of the PSDEs between the same two pixels in other image data sets as they have been computed from a completely independent set of

Fig. 3. Histograms of (a) and (c) two different samples of PSDEs and (b) and (d) the corresponding estimated GMMs.

the density of the PSDEs is strongly influenced by the specific nature of the faulty pixel. The more aberrant or faulty a pixel is, the larger are its PSDEs with the other $n - 1$ pixels, leading to a larger number of PSDEs nearer the tails of the density.

Figures 3(b) and 3(d) show two different densities are correspondingly modeled by a single component GMM (in effect a Gaussian) in Fig. 3(b), and a two-component GMM in Fig. 3(d), where the number of components $K$ have been determined by the BIC criterion. Each individual Gaussian component of the GMM represents a cluster in the density of the difference estimates. By varying the number of clusters or components of the GMM ($K$), we benefit from having more flexibility in modeling the PSDEs. As the density of PSDEs change, the value of $K$ can be varied to optimize the fit between the PSDEs and the model. This GMM modeling step serves as a useful precursor to the subsequent classification described in Subsection 4.B, where the individual Gaussian components of the PSDEs serve as the categories or clusters into which additional instances of PSDEs are classified. Without such partitioning by means of clustering, the classification step would only be possible if a separate training phase was introduced, where a predictive classification model might be trained from labeled training data, which is contradictory to our objective of developing an autonomous unsupervised machine learning technique.

B. Classification of the Difference Estimates

Once we have estimated the parameters of the GMM, the next step in the process is to use the GMM to draw inferences on additional instances of such PSDEs. Let $\Delta_i$ be a difference estimate computed between two pixels $a$ and $b$. If $\Delta_i$ exhibits consistently high probabilities in the High component of the GMM, across multiple image data sets, it is an indication that one of the two pixels $a$ or $b$ or even both of them are faulty. Hence, the probabilities of every PSDE in multiple components of the GMM over multiple image data sets is a key indicator of the aberrance of the pixels in the pixel pair associated with those PSDEs. Consequently, it is possible to classify a PSDE into one of the GMM components based on the probability of its membership in the components of the GMM, estimated over multiple image data sets, and we estimate such probabilities for all the PSDEs by applying the Bayes theorem. Furthermore, once we classify all the PSDEs of a single pixel, we then observe the proportion of the PSDEs that have been classified in the High component, as we can deduce whether or not the pixel is faulty; it will be faulty if a large proportion of its difference estimates are classified in the High component. Thus, in effect, this process of classifying PSDEs plays a key role in the identification of faulty pixels.

Let the components of the GMM be represented as $C_1,...,C_K$, where $C_K$ is the High component. These components represent the classes of PSDEs where each class is modeled as a Gaussian represented by the parameter vector $\theta_k = (\mathbf{x}_k, \mu_k, \sigma_k)$, where $k = 1, ..., K$. Our objective is to classify the PSDEs from $N - d$ image data sets into these classes. Let $j = d + 1, ..., N$ indicate the indices of the image data sets to be used for this classification step. The corresponding image data sets are $\mathbf{I}_{S_1}, ..., \mathbf{I}_{S_N}$. Each of these image data sets will have $(n \times (n - 1))/2$ PSDEs that have been computed between $n$ pixels within a block. For the purpose of classification, let $\Delta^j_i$ represent a single PSDE estimates between two pixels $a$ and $b$ in image data set $j$, where $j = (d + 1, ..., N)$ and $i = 1, ..., n \times (n - 1)/2$. For the complete $N - d$ sets, we can form a vector of inputs $\hat{\mathbf{d}}_i = (\Delta_i^{d+1}, \Delta_i^{d+2}, ..., \Delta_i^N)$. To classify this vector set of inputs we apply the Bayes theorem as shown in Eq. (5):

$$
P(C_k|\hat{\mathbf{d}}_i) = \frac{P(C_k) \times P(\hat{\mathbf{d}}_i|C_k)}{\sum_{k=1}^{K} P(C_k) \times P(\hat{\mathbf{d}}_i|C_k)}.
$$

The vector $\hat{\mathbf{d}}_i$ is classified into the class $C_k$ if its posterior probability $P(C_k|\hat{\mathbf{d}}_i)$ is the highest for that class. This is also called the maximum a posteriori (MAP) decision rule that minimizes the probability of error or error rate under zero-one loss [43] and is shown in Eq. (6):

$$
\text{classify}(\hat{\mathbf{d}}_i) = \arg \max_k P(C_k|\hat{\mathbf{d}}_i).
$$

It is important to note that while applying the Bayes theorem to vector of inputs $\hat{\mathbf{d}}_i$ as shown in Eq. (5), we assume that every $\Delta^j_i$ is independent of all the other $\Delta^j_i$'s, where $j = d + 1, ..., N$, $j \neq k$, i.e., a PSDE between two pixels in one image data set is independent of the PSDEs between the same two pixels in other image data sets as they have been computed from a completely independent set of
images with no overlap. Figure 4 elucidates this independence property, the importance of which is explained in greater detail later in this section.

Subsequently, assuming the independence of the inputs given the estimated GMM as shown in Fig. 4, Eq. (5) can be simplified to Eq. (7):

$$P(C_k|\delta_i) = \frac{P(C_k) \times \prod_{j=d+1}^{N} P(\Delta_j^i|C_k)}{\sum_{k=1}^{K} P(C_k) \times \prod_{j=d+1}^{N} P(\Delta_j^i|C_k)}.$$  (7)

Each $P(\Delta_j^i|\theta_k)$ is estimated according to Eq. (3), replacing $\Delta_i$ by $\Delta_j^i$ for each of the $N-d$ image data sets indexed by $j$, where $j = d+1, \ldots, N$. Furthermore, the Bayes theorem in Eq. (7) is applied to every single vector of PSDEs, $\delta_i$, thereby producing the corresponding posterior probabilities of every PSDE. For the purpose of our technique, an important point to consider is the choice of the prior probability of the $K$ classes denoted by the term $P(C_k)$ in Eq. (7). Modeling by means of a GMM offers the natural advantage that every Gaussian component ($k$) has an associated weighting factor or mixing proportion denoted by $\alpha_k$, which represents the probability that a point generated randomly by a GMM is produced by the component $k$. Hence, $\alpha_k$ represents the influence of the component $k$ in the overall mixture [40]. For our purposes, we use $\alpha_k$ of every Gaussian component of the GMM that is estimated by the EM algorithm as our choice of the prior probability of the $k_{th}$ class, i.e., $P(C_k)$. This follows from the discussion in Subsection 4.A, where we described how the individual Gaussian components of the GMM partition the density of PSDEs and how every Gaussian component represents a class or category into which additional PSDEs are classified into by Eq. (7). Consequently, we regard the prior probability of each class, which is represented by a single Gaussian component of the GMM, as equivalent to the weighting factor of the corresponding Gaussian component in the overall mixture. The GMM therefore offers an advantage towards the application of the Bayes theorem; more specifically, towards the specification of the prior probability of a class, which in our case, is estimated from a sample of PSDEs by the EM algorithm. Consequently, replacing $P(C_k)$ with $\alpha_k$ and representing each class $C_k$ as a Gaussian with parameters $\mu_k, \sigma_k$,

$$P(C_k|\delta_i) = \frac{\alpha_k \times \prod_{j=d+1}^{N} P(\Delta_j^i|\mu_k, \sigma_k)}{\sum_{k=1}^{K} \alpha_k \times \prod_{j=d+1}^{N} P(\Delta_j^i|\mu_k, \sigma_k)}.$$  (8)

Table 1 is an excerpt from the table of the posterior class probabilities of a hypothetical problem where there are three classes $C_1, C_2$, and $C_3$. Each row of the table contains the posterior probabilities in the three classes of the PSDEs between a pixel and two others, computed from image data sets $I_1$ to $I_N$ – 1 according to Eq. (5). The last column of Table 1 is the final decision as to which of the three classes a difference estimate between a pixel pair belongs to, shown as Eq. (6), which is an application of the MAP decision rule of Eq. (6). While applying the Bayes theorem, it is possible that the posterior class probabilities in the last column of Table 1 may not be an accurate estimate of the true probabilities. However, as mentioned by Zhang [44] and Domingos and Pazzani [43] from their investigations into the performance of the Naive Bayes classifier, which is an application of the Bayes theorem that assumes independence of input attributes, the class with the maximum posterior probability is often unchanged. Even in cases where conditional independence of input attributes are violated, they show that application of Naive Bayes often leads to optimal classifications. In addition, they show that although the probability estimates produced in the Bayes formalism might be poor, the classifications are accurate. Zhang and Su in [45] mention that as long as the class with the maximum posterior probability estimate is identical to the actual class, the classification algorithm gives the correct classification. We wish to highlight that there is a subtle difference between our application of the Bayes theorem and the Naive Bayes classifier, although in both cases an assumption of inputs is made. The difference is that a Naive Bayes classifier is used to classify a set of feature or attributes, which are independent of one another. On the contrary, in our case, for every application of the Bayes theorem as per Eq. (8), the inputs are PSDEs between the same two pixels in multiple image data sets. Although they are independent, they cannot be regarded as independent features or attributes in a strict sense, as they represent multiple estimates of a single PSDE.

C. Identifying Faulty Pixels

In Subsection 4.B we explained how the Bayes theorem is applied to classify every PSDE of every pixel. Every single classification is indicative of the aber-

![Fig. 4. Independence of the inputs in $\delta_i$, given a class $C_k$.](Image)

| Pixel $a$ | Pixel $b$ | $P(C_1|\delta_i)$ | $P(C_2|\delta_i)$ | $P(C_3|\delta_i)$ | $\text{classify}(\delta_i)$ |
|----------|----------|---------------------|---------------------|---------------------|---------------------|
| 1        | 2        | 0.1                 | 0.7                 | 0.2                 | 2                   |
| 1        | 3        | 0.15                | 0.8                 | 0.05                | 2                   |
rurance of a pixel. By collating the classifications of a pixel it becomes possible to determine whether or not the pixel is defective. A pixel is defective if a large majority of its PSDEs are classified as being aberrant; they exhibit maximum posterior probabilities for the High component of the GMM. To determine this, we first amass the \((n - 1)\) final classifications of every pixel, each of them computed according to Eq. (6). For each pixel, we then determine the proportions of the classifications of its \(n - 1\) difference estimates in the different components of the GMM. For instance in the case of a 3 component GMM, a particular pixel \(P_i (i = 1, \ldots, n)\) may have 60% of its difference estimates classified in \(C_1\), 30% in \(C_2\), and 10% in \(C_3\). \(C_3\) then represents the High component, and defective pixels should have a higher proportion of their difference estimates classified in \(C_3\) as opposed to other healthy pixels. Subsequently, if we rank the \(n\) pixels in decreasing order of proportion of difference estimates classified in the High component, we find that defective pixels occupy the higher ranks relative to the pixels that are functioning normally.

From Fig. 5 we see how the ranking of pixels is based upon the proportion of difference estimates classified in the High component shown in Fig. 5(a). For our purposes, we consider those pixels that rank in the top \(\text{th}\)% of the \(n\) pixels analyzed. Figure 5(b) highlights the top \(\text{th}\)% of \(n = 30\) pixels (resulting in three pixels being highlighted) and these pixels have a large proportion of their difference estimates classified in the High component. In Section 6, we show how conspicuously faulty pixels of the Hot, Dead and Stuck types will consistently rank in the top \(\text{th}\)% in a group of \(n\) pixels. On the other hand, pixels that are abnormally sensitive do not always rank amongst the top \(\text{th}\)%, and their ranking will change depending on how aberrant they are, thereby, making them harder to detect.

Although it becomes necessary to introduce a threshold to distinguish the more aberrant and faulty pixels from the less faulty ones, which is less than ideal, we shall see later that in most cases faulty pixels can be detected well with such a threshold, as depending on how faulty they are, a large proportion or all of their PSDEs are classified in the High component. Rather than using the posterior probability values of the classified PSDEs of a pixel, we rely on their classification. Zhang and Su, in the context of evaluating a Naive Bayes classifier [45], showed that a combination of classification and ranking based on the application of Naive Bayes is more accurate as opposed to simply estimating the posterior probabilities. In light of this, we believe we too benefit from relying on the ranking of pixels based on the classifications of their PSDEs, as opposed to drawing inferences based on the posterior probabilities.

5. Ground-Truth Technique

We use a Videology monochrome digital camera equipped with a 1/4 in. Sony Ex-View CCD, having 768 × 494 active pixels. We use a ground-truth technique to discover the defective pixels in the CCD array of this camera, and it serves as a precursor to our more sophisticated novel integrated technique. In addition to discovering the faulty pixels in calibration images, this also assigns a deviance value to such pixels, which serves to indicate how defective those pixels are. The importance of this assignment is discussed in Section 6, where our novel integrated technique is evaluated on the basis of its performance at detecting the defective pixels with different deviance values assigned by the ground-truth technique. For the ground-truth technique, we take flatfield calibration images in a laboratory setup. These images are devoid of any scene information, and the reported pixel intensities are closely representative of the behavior of the pixels. This experimental setup also helps us obtain calibration images under the control of two important parameters: the temperature of the camera and the exposure setting. To precisely control the temperature of the camera, we augment our camera with a thermoelectric cooler module [46,47], which is controlled by a thermostat. The exposure setting is controlled by the software that comes with the camera. Figure 6 offers a front view of our experimental setup. The flatfield calibra-

Fig. 5. Identifying the faulty pixels based on the ranking in terms of proportions of classification in the ‘High’ component.

![Fig. 5](image_url)

![Fig. 6](image_url)
tion images are taken in a dark room and an LED is used as a light source. The light from the LED is allowed to pass through a pair of optical diffuser plates before reaching the camera and falling on the CCD. We defocus the camera lens and this experimental setup enables us to take highly precise flatfield calibration images. Figure 7 shows the overall setup.

Let \( M \) be a \( n \times q \) matrix composed of the measurements of \( n \) pixels in \( q \) calibration images. Since the ground truth is based on a simple test for normality, we use the measurements of a large number of pixels (\( n > 10,000 \)). If we average the measurements of every pixel across the \( p \) images, we get a vector of averaged pixel measurements \( \langle M \rangle = (\langle M \rangle_1, \ldots, \langle M \rangle_p) \), where \( \langle \cdot \rangle \) implies averaging. For every trial, each \( \langle M \rangle_i \) is obtained by averaging the measurement of pixel \( i \), \( (i = 1, \ldots, n) \) over \( q > 20 \) images. Averaging reduces pixel noise and increases the signal to noise ratio, which results in vector \( \langle M \rangle \) being devoid of noise in individual images and is truly representative of the measurements of the \( n \) pixels. We compute the trimmed mean \( c_p \), which is found to be more robust than the standard mean \([48,49]\) and the corresponding standard deviation \( c_o \) for the vector \langle M \rangle. We remove the highest and the lowest values from the vector \( M \) to inhibit the influence of the most stark outliers only and then compute the summary statistics \( c_p \) and \( c_o \). Subsequently, for every averaged pixel measurement in \( M_i \), we then estimate its normalized deviance from the trimmed mean \( c_p \) as per Eq. (9), resulting in the vector \( Z = (Z_1, \ldots, Z_n) \). We then verify if the vector \( Z \) is normally distributed by applying the robust Kolmogorov–Smirnov test for normality \([50]\), which compares the values in the vector \( Z \) to the standard normal distribution. The null hypothesis is that \( Z \) has a normal distribution, and the alternative hypothesis is that it is not normally distributed. For our trials, we conduct this test at the 1% significance level and reject the null hypothesis if the test is significant at this level. The statistical properties of the normal distribution make it possible to easily identify the outliers in normally distributed data as explained in \([51,52]\):

\[
Z_i = \frac{M_i - c_p}{c_o} \quad \text{where} \quad i = 1, \ldots, n. \tag{9}
\]

In the same vein, pixels that have abnormal \( Z_i \) values, which are greater than certain fixed thresholds represent the outlier pixels. To identify such defective pixels, we use a simple heuristic in the form of a threshold—\( 3c_o \) (precisely, measurements \( \geq 3c_o \)) in every normality trial to identify the outlier normalized deviances as the ones that are greater than the threshold. Since, it is possible for a pixel’s measurement to be greater than the thresholds purely by chance, we conduct multiple (around 30) independent trials of normality to mitigate experimental error. As an alternative to using normality statistics, it is possible to use a statistic such as median absolute deviation or MAD based on the median of \( \langle M \rangle \) to identify defective pixels, as this statistic is more robust to outliers and more generally when underlying assumptions such as normality of data are incorrect \([53]\). However, in our case, since we are able to obtain flatfield images that are similarly distributed and exhibit good normality properties, and because we mitigate the influence of outliers and the experimental error, we abide by this simple ground truth that is based on well-known statistics based on an assumption of normality and using simple heuristics to discriminate outlier pixels. Assumption of similar distribution of pixel values in flatfield images also forms the core of Lopez-Alonso and Alda’s PCA-based defective pixel identification technique \([28]\).

For every trial, we use a completely different set of images during the averaging process, and for every pixel we estimate the normalized deviance, \( Z = (Z_1, \ldots, Z_n) \). Each \( Z_i (i = 1, \ldots, n) \) is then compared with the threshold \( 3c_o \), and all those pixels whose \( Z_i \)'s are \( > \) than the threshold in every trial are extracted. As a consequence of the multiple trials, for every pixel we can estimate the detection rate \( \text{DR}_{gt} \); the proportion of the total number of trials in which their normalized deviances are greater than the threshold. To eliminate the possibility of a pixel being greater than the threshold in some trials purely by chance, we regard pixels to be defective if they exhibit \( \text{DR}_{gt} \) values greater than 40%. From our experiments we find that this limit is robust enough and helps discriminate the truly defective pixels. For these defective pixels, we average their normalized deviances across the multiple trials, to get their mean normalized deviances \( \text{avg}(Z) = (\text{avg}(Z_1), \ldots, \text{avg}(Z_j)), (j < n) \), which serve to indicate how faulty the pixels really are, and by assigning a numerical value, it is possible to accurately quantify their aberrance. At the end of this exercise, we get a list of the defective pixels in our CCD, along with an indication of how faulty they are based on their \( \text{DR}_{gt} \) and \( \text{avg}(Z) \) values. To illustrate the point that these estimates are robust at detecting the truly defective pixels: the probability of a pixel having a \( \text{DR}_{gt} = 50\% \) and \( \text{avg}(Z) = 3 \) can be estimated as per the binomial distribution, where the probability \( p \) per trial, for \( \text{avg}(Z) = 3 \), according to normal statistics is approximately 0.003. If we conduct 10 independent trials, then the probability of getting \( \text{avg}(Z) = 3 \) in five
trials (DR$_{gt} = 50\%$), according to the binomial distribution is extremely small.

6. Application to CCD

Here we review the performance of our integrated technique at the detection of defective pixels in the CCD detector of our experimental camera. Our integrated technique is applied to regular images captured with our experimental camera that have rich scene information, as shown in Fig. 8. We evaluate our integrated technique primarily at the detection of both faulty pixels that have been discovered by the ground-truth technique in Subsection 6.B. In our earlier work [9], we reported the performance of our technique at detecting defective pixels and the influence of some of the parameters involved in our methodology. However, in light of having overhauled our ground-truth technique and having made changes to the GMM modeling steps, we redo all our experiments and consequently, in this paper, we report the results from the new set of experiments. In addition, we investigate the influence of some more parameters, which were not discussed in [9]. We also perform simulations where we evaluate our technique’s performance at the detection of defective pixels, where the defects are artificially introduced in the CCD as per the defect levels proposed by Dudas et al. [30]. All the experiments with the CCD are applications of our technique to detecting specific pixels in the CCD only that we know from the ground-truth technique are defective. The main objective of doing so is to get a thorough insight into how our technique performs at the detection of different known defective pixels. Although, for the experiments with the CCD we do not analyze entire images captured with the camera and limit our attention to analyzing specific defects. This is addressed in Section 7, where we apply our technique to the detection of the defective pixels in the image sensor arrays of an astronomical camera and also evaluate our technique’s impact on the photometry.

A. Preliminary Procedures

Before we apply our integrated technique, we create multiple image banks, each composed of on average 200 random images containing rich scene information taken in a particular setting, such as a room or a laboratory. Every image bank is composed of images that have been taken for a unique combination of exposure and camera temperature, which are controlled during the image acquisition. As per our methodology described in Section 4 for an image bank chosen for analysis, we construct $N$ separate sets of image data sets, each comprising of the same number $p$ of randomly selected images each, and we compute PSDEs using those $p$ images. For every trial of our analysis on images from a chosen image bank, we randomly select $N \times p$ images. For all our trials, we use PSDEs from only one set of the $N$ sets for the GMM modeling step. The Bayes theorem is then applied to the PSDEs from the remaining $N - 1$ sets. For instance, given an image bank consisting of 200 images in total, we can randomly select 60 images. We can then form 5 image data sets, each containing $p = 12$ images—IS$_1$, ..., IS$_5$. We can use IS$_1$ for the GMM modeling step and IS$_2$, ..., IS$_5$ for the Bayes classification step. In our earlier paper [9], we used the K-means algorithm to initialize the EM algorithm, a step that was necessitated by EM’s slow convergence otherwise. However, for all the experiments presented in this paper, we use small runs of the EM algorithm to initialize EM. This strategy is known as the eM strategy for initializing the EM algorithm proposed by Biernacki et al., which, as shown by the authors, yields good starting values for the EM algorithm [54]. The EM algorithm is subsequently run for 100 iterations. To evaluate the performance of our integrated technique at the detection of faulty pixels using images taken under a particular combination of exposure and camera temperature, we conduct 30 independent trials and use the “detection rate” (DR) as a metric for evaluating our technique’s performance in all the 30 trials.

Our technique is applied to a block of $n$ pixels centered on the one of interest, i.e., a pixel deemed to be defective by the ground-truth technique, which we wish to detect by means of our integrated technique. This selection of a block of pixels is as per the configuration shown in Fig. 1(d). After applying our technique, a single detection is recorded if the pixel under test ranks amongst the top $th = 10\%$ of $n$ pixels analyzed and ranked as per the methodology described in Subsection 4.C. The DR is the number of such detections divided by the total number of trials. We choose DR as the metric primarily because, with regards to the CCD, we try to empirically estimate the performance of our technique at detecting very specific pixels. It also makes it possible to compare the DR to the DR$_{gt}$ assigned by the ground-truth technique to the same defective pixels, thereby allowing us to get a measure of how accurately the same defective pixel is detected by our integrated technique in real images. For all trials in Subsection 6.B, we use a fixed $th = 10\%$ threshold at the end of the ranking step. We show that this threshold results in good diagnostic accuracy in terms of the DR values at detecting the defective pixels in the CCD. However, when we apply our technique to the broader application of autonomously detecting the defective pixels in

Fig. 8. Sample images captured with experimental camera.
the detectors of the astronomical camera in Section 7, we vary \( \theta \) and we closely investigate its effect on our technique's performance. From our ground-truth experiments, we also find that there are no dead or stuck pixels in the CCD. Hence, for evaluating the performance of our technique at detecting such pixels, we artificially introduce dead and stuck pixels in the CCD. Consequently, the \( DR_{gt} \) values of the dead and stuck pixels is deemed to be not recorded (NR).

### B. Experimental Results for CCD

Our novel integrated technique uses the \((n \times (n-1))/2\) PSDEs from \(d\) image data sets and models their density by means of a GMM; in effect, a total of \(d \times (n \times (n-1))/2\) PSDEs are used. We first observe the influence of parameter \(d\) on our overall methodology and its influence in the detection of two different defective pixels in terms of their \(DR\).

For this purpose, we use 72 images that are divided into 6 image data sets containing \(p = 12\) images each. The parameter \(d\) is varied from \(d = 1\) to \(d = 4\). It is clear that as \(d\) increases more data is available for the GMM step and less for the applying Bayes. Table 2 presents the results of the trials. As the value of \(d\) is increased, a greater number of points are used for the density estimation. However, it is clearly evident from Table 2, that this does not lead to an improvement in the corresponding \(DR\). In the case of the \(avg(Z) = 4.682\) pixel, its \(DR\) reduces with increasing \(d\). We believe that this can be explained in terms of the interplay between the GMM step and the Bayes step. The primary purpose of the GMM step is to model a sample of the PSDEs, which serves as a prior model for the Bayes step. This prior model is subsequently refined by the application of the Bayes theorem to additional PSDEs. By using \(d = 1\) for all our trials, we leverage the reduced complexity derived from EM estimating a good prior model from relatively few data points.

We then determine how many GMM components \(K\) are selected by BIC to correctly model the density of PSDEs in the case of five different defective pixels in the CCD. For every defective pixel type, we conduct 30 independent trials, with each trial run for a choice of \(1, 2, 3\), and \(3\) components, and we observe how many times BIC selects one of the three choices. Table 3 shows the result. From Table 3, we can clearly infer that in most cases the BIC criteria for model selection selects 2 Gaussian components or \(K = 2\) to model the density of PSDEs. In some trials \(K = 3\) components are selected, but a choice of \(K = 2\) is most favored. We also observe that trials where our technique fails to detect defective pixels, the most notable of which is the case of the \(avg(Z) = 4.682\) pixel and is due to BIC selecting \(K = 1\) or a single component to model the density of PSDEs. The diagnosis fails for a single GMM component because there is no distinct high component to serve as the outlier component. Consequently, all pixels have their PSDEs classified in the single component, and therefore it is not possible to distinguish a defective pixel from a normally functioning pixel. We therefore run all remaining trials with BIC having to choose between GMMs with \(K = 2\) or \(K = 3\) components.

We then evaluate the influence of four different parameters on the detection of the defective pixels: number of images, number of pixels, defective cluster size, and the parameter \(p\). Table 4 shows the performance of our technique for varying numbers of images at the detection of four different kinds of faulty pixels discovered by the ground truth. We perform this analysis for a fixed number of \(n = 25\) pixels, one of which is the defective pixel under test. Our results are compared with the \(DR_{gt}\) of those pixels assigned by the ground truth. From the results presented in Table 4, it is possible to infer that the number of images needed by our technique to accurately detect defective pixels is dependent on how faulty the pixel is in terms of its \(avg(Z)\) value assigned by the ground-truth technique. In the case of the two pixels that are not conspicuously faulty, such as the two pixels with \(avg(Z) = 7\) and \(avg(Z) = 3.5\), our technique converges to the \(DR_{gt}\) values assigned by the ground-truth technique in less than 48 images. However, for the conspicuously faulty pixels, such as dead and stuck pixels, our integrated technique converges

### Table 2. Detection Rate for Different Numbers of Image Data Sets

<table>
<thead>
<tr>
<th>Data Sets</th>
<th>Ground-Truth</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel Type</td>
<td>DR&lt;sub&gt;gt&lt;/sub&gt;</td>
<td>Detection Rate</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>avg(Z) = 11.143</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>avg(Z) = 4.682</td>
<td>78.5</td>
<td>56.7</td>
<td>50</td>
<td>36.67</td>
</tr>
</tbody>
</table>

### Table 3. Numbers of Selections of Different Numbers of GMM Components by the BIC Criterion (K) in 30 Trials

<table>
<thead>
<tr>
<th>Pixel Type</th>
<th>DR&lt;sub&gt;gt&lt;/sub&gt;</th>
<th>Result</th>
<th>No. of Selections</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>avg(Z) = 11.143</td>
<td>100</td>
<td>96.7</td>
</tr>
<tr>
<td></td>
<td>avg(Z) = 4.682</td>
<td>78.5</td>
<td>3.3</td>
</tr>
<tr>
<td></td>
<td>avg(Z) = -5.0543</td>
<td>63.15</td>
<td>96.7</td>
</tr>
<tr>
<td>Stuck</td>
<td>NR</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>Dead</td>
<td>NR</td>
<td>100</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table 4. Effect of Number of Images on Performance<sup>a</sup>

<table>
<thead>
<tr>
<th>No. of Images</th>
<th>Pixel Type</th>
<th>DR&lt;sub&gt;gt&lt;/sub&gt;</th>
<th>Detection Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg(Z) = 7</td>
<td>100</td>
<td>40</td>
<td>83.3</td>
</tr>
<tr>
<td>avg(Z) = 3.5</td>
<td>50</td>
<td>26.7</td>
<td>33.3</td>
</tr>
<tr>
<td>Stuck</td>
<td>NR</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Dead</td>
<td>NR</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

<sup>a</sup>Blocks of 25 pixels containing the defective pixels were analyzed.
rapidly and can detect such pixels with as few as 10 images.

Table 5 shows the performance of our integrated technique for varying numbers of pixels, for a fixed number of 72 images. It is evident that even for a small number of pixels, the performance of our integrated technique in terms of DR is comparable to the ground-truth technique's assigned $DR_{gt}$. Although, for a higher number of pixels analyzed, there is an improvement in the DR due to having a $th = 10\%$ threshold, which leads to a corresponding increase in the likelihood of a faulty pixel being found in the higher ranks. Table 6 shows that $p$—the number of images used to estimate a PSDE—does not have a significant impact on the DR of all the defective pixels except in the case of the $avg(Z) = 3.5$ pixel. In this case, we do see that higher values of $p$ makes it slightly easier to detect the 3.5 pixels, which is only very subtly aberrant, almost at the verge of being a conspicuously defective pixel. This parameter $p$ can be varied depending on the number of images available for our technique. In the event that not many images are available for analysis, we believe that the value of the parameter $p$ can be lowered without significantly deteriorating the performance of our technique for conspicuously defective pixels. This results in a flexible technique that can be adapted to a multitude of applications constrained by the number of images available.

Once we evaluate the influence of the most salient parameters on the performance of our integrated technique, we then apply our integrated technique to the task of detecting a large number of different defective pixels in images captured under different combinations of exposure and temperature. Table 7 presents the results of the performance of our integrated technique at such a task. Although we do not check for dead and stuck pixels in this case, some of the pixels investigated are definitely hot pixels, which appear as bright spots at higher camera temperatures. Also, many of the pixels shown in Table 7 are pixels, which exhibit different $avg(Z)$ and $DR_{gt}$ values at different combinations of exposure setting and camera temperatures, with both DR and $avg(Z)$ of those pixels experiencing an increase due to a rise in the temperature of the camera and at longer exposure setting. The results in Table 7 show that our technique performs well in most cases and its performance at detecting the defective pixels is comparable to the ground truth. Only in two cases ($avg(Z) = 3.43$ and $avg(Z) = 4.7$) does our technique perform significantly worse than the ground truth and both those cases are of two different hypersensitive pixels that are not conspicuously defective.

C. Simulation with Artificial Defects

For the purpose of evaluating our integrated technique at simulated defective pixels, we consider the methodology followed by Chapman et al. [8] and Dudans et al. [30]. They model the output of a pixel at a particular $X$-$Y$ spatial location in the detector as $y_{ij} = mx_{ij} + b$. By altering the gain $m$ in the equation, they artificially introduce defects in the image sensor array. Like Chapman et al. we do not consider different values of the bias $b$ because, as they mention, it can easily be removed by using calibrated or flat-fielded images. We work on the premise that the gain or sensitivity $m$ influences the response of a pixel as different values of $m$ and gives rise to two different kinds of faulty pixels. A pixel having gain ($m < 1$) is hyposensitive; the lower the value of $m$, the more grossly hyposensitive the pixel will be. In the same

Table 5. Effect of Number of Pixels on Performance

<table>
<thead>
<tr>
<th>Pixel Type</th>
<th>$DR_{gt}$</th>
<th>Detection Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>$avg(Z) = 7$</td>
<td>100</td>
<td>93.3 93.3 100 100</td>
</tr>
<tr>
<td>$avg(Z) = 3.5$</td>
<td>50</td>
<td>46.7 46.7 60 60</td>
</tr>
<tr>
<td>Stuck</td>
<td>NR</td>
<td>100 100 100 100</td>
</tr>
<tr>
<td>Dead</td>
<td>NR</td>
<td>100 100 100 100</td>
</tr>
</tbody>
</table>

Note: *In total 72 images per trial, with $p = 12$, resulting in six image data sets used.*

Table 6. Effect of Parameter $p$ on Detection Rate

<table>
<thead>
<tr>
<th>$p$</th>
<th>$5$</th>
<th>$8$</th>
<th>$10$</th>
<th>$12$</th>
<th>$15$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel Type</td>
<td>$DR_{gt}$</td>
<td>Detection Rate</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$avg(Z) = 7$</td>
<td>100</td>
<td>100 100 100 100 96.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$avg(Z) = 3.5$</td>
<td>50</td>
<td>46.7 46.7 50 63.3 60</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stuck</td>
<td>NR</td>
<td>100 100 100 100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dead</td>
<td>NR</td>
<td>100 100 100 100</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: *Blocks of 25 pixels containing the defective pixels were analyzed. For every $p$ total number of images chosen was such that the number of image data sets constructed was seven in all cases. For $p = 5$, for instance, $5 \times 7 = 35$ images were analyzed.*

Table 7. Performance at the Detection of Different Faulty Pixels Detected by the Ground-Truth Technique

<table>
<thead>
<tr>
<th>Pixel Type</th>
<th>$DR_{gt}$</th>
<th>Exposure Temperature</th>
<th>Detection Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>$avg(Z) = 6.69$</td>
<td>100</td>
<td>1/50</td>
<td>36°C</td>
</tr>
<tr>
<td>$avg(Z) = 3.43$</td>
<td>65</td>
<td>1/50</td>
<td>36°C</td>
</tr>
<tr>
<td>$avg(Z) = 10.18$</td>
<td>100</td>
<td>1/50</td>
<td>46°C</td>
</tr>
<tr>
<td>$avg(Z) = 5.26$</td>
<td>82.3</td>
<td>1/50</td>
<td>46°C</td>
</tr>
<tr>
<td>$avg(Z) = 3.65$</td>
<td>100</td>
<td>1/50</td>
<td>55°C</td>
</tr>
<tr>
<td>$avg(Z) = 6.45$</td>
<td>100</td>
<td>1/50</td>
<td>55°C</td>
</tr>
<tr>
<td>$avg(Z) = 10.69$</td>
<td>100</td>
<td>1/25</td>
<td>36°C</td>
</tr>
<tr>
<td>$avg(Z) = 5.24$</td>
<td>100</td>
<td>1/25</td>
<td>36°C</td>
</tr>
<tr>
<td>$avg(Z) = 15.45$</td>
<td>100</td>
<td>1/25</td>
<td>46°C</td>
</tr>
<tr>
<td>$avg(Z) = 6.7$</td>
<td>100</td>
<td>1/25</td>
<td>46°C</td>
</tr>
<tr>
<td>$avg(Z) = -3.4573$</td>
<td>41.17</td>
<td>1/25</td>
<td>46°C</td>
</tr>
<tr>
<td>$avg(Z) = 11.14$</td>
<td>100</td>
<td>1/25</td>
<td>55°C</td>
</tr>
<tr>
<td>$avg(Z) = -5.05$</td>
<td>63.15</td>
<td>1/25</td>
<td>55°C</td>
</tr>
<tr>
<td>$avg(Z) = 4.7$</td>
<td>78.5</td>
<td>1/25</td>
<td>55°C</td>
</tr>
</tbody>
</table>

Note: *In total 72 images per trial, with $p = 12$, resulting in six image data sets used and blocks of 25 pixels containing the defective pixels were analyzed.*
In total 60 images per trial, with \( p = 12 \), resulting in six image data sets (one used for GMM step and the remaining five for the Bayes step). Windows containing 25 pixels are analyzed.

### Table 8. Performance of our Technique at Detecting Different Pixels Subject to Different Values of Gain (m)\(^a\)

<table>
<thead>
<tr>
<th>Exposure</th>
<th>Detection Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9–0.93</td>
<td>0.94</td>
</tr>
<tr>
<td>1/50</td>
<td>100</td>
</tr>
<tr>
<td>1/25</td>
<td>100</td>
</tr>
</tbody>
</table>

\(^a\)In total 60 images per trial, with \( p = 12 \), resulting in six image data sets (one used for GMM step and the remaining five for the Bayes step). Windows containing 25 pixels are analyzed.

Table 9. Performance of our Technique at Detecting Clusters of Defective Pixels, Subject to Different Values of Gain (m)\(^a\)

<table>
<thead>
<tr>
<th>Cluster Size</th>
<th>DR</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>96.7</td>
</tr>
<tr>
<td>4</td>
<td>96.7</td>
</tr>
</tbody>
</table>

\(^a\)Total 60 images per trial and \( p = 12 \), resulting in six image data sets (one used for GMM step and the remaining five for the Bayes step). Windows containing 25 pixels are analyzed.

### 7. Application to Astronomical Data

To demonstrate its utility, we apply our novel integrated technique to the autonomous detection of the defective pixels in two out of the four detectors of the WFCAM NIR camera on the UK Infrared Telescope on the summit of Mauna Kea in Hawaii, which when commissioned represented a significant leap in deep NIR survey capability [12]. WFCAM has been designed specifically to carry out large-scale survey observations several magnitudes fainter than current capabilities [55]. The WFCAM IR camera is equipped with four 2048 × 2048 Rockwell Hawaii-II (HgCdTe 2048 × 2048) arrays with a pixel size of 0.4 in. [12]. Figure 9 shows the layout of the four detector arrays with an autoguider CCD located in the center of the instrument focal plane, at an angle of 45° with respect to the NIR detectors. The autoguider used to send tip-tilt commands to the UKIRT secondary mirror unit. The image in Fig. 9 was obtained from the Joint Astronomy Centre (JAC) website (http://www.jach.hawaii.edu/UKIRT/instruments/wfcam/user_guide/description.html).

![Four detector arrays on WFCAM.](image_url)
A. NIR Image Data

Near infrared (NIR) images of the small cloud IC 1396 W in Cepheus were used as a test case for our data analysis. The cloud is situated close to the galactic plane (right ascension: 21°26′06.1″, declination +57°56′17″) and thought to be a sight of active star formation [56]. The images were taken at the United Kingdom Infra-Red Telescope (UKIRT) using WFCAM. We used the H-band filter (\(\lambda = 1.65\, \mu m\) and \(\Delta \lambda = 0.4\, \mu m\)) for our observations. Images were taken as a sequence. Each of the images was integrated by 5 s to prevent saturation from the background. A 5 position random jitter pattern with off-sets of a few arc seconds was applied. At each of these a 4 position microstepping pattern (a 0.2″ shift of the telescope) was added to fully sample the point spread function (PSF) of the stellar images. The images are stored as FITS multie xtension files and there are 163 images in total that were downloaded from the Cambridge Astronomical Survey Unit’s (CASU) archive of raw images. Each three-dimensional FITS data cube contains four planes, where each plane holds an image captured by one of the four detectors.

B. Applying Our Technique

We evaluate our technique at the detection of the defective pixels in the inner 512 × 512 patch of WFCAM’s detectors 1 and 4. We restrict our application to this patch in the NIR images described in Subsection 7.4 for sheer computational reasons only. To apply our integrated technique to such images, we followed similar procedures to the ones explained in Subsection 6.4. We apply our technique to small 8 × 8 blocks containing 64 pixels within the 512 × 512 patch, as per the configuration shown in Fig. 1(a). This results in a total of 4096 independent blocks being analyzed. We use BIC to determine the appropriate number of components to model the density of PSDEs for each block of \(n = 64\) pixels and their corresponding 2016 PSDEs \((64 \times 63)/2\). To evaluate the performance of our technique, we conduct 10 independent trials, and for every trial we use 60 randomly selected images. We compute PSDE using \(p = 12\) images, which results in five image data sets in total. From the five image data sets \(I_{S_1}, \ldots, I_{S_5}\), we use the PSDEs from \(d = 1\) of the sets for GMM estimation and apply the Bayes theorem to each of the PSDEs in the remaining four sets. At the end of each run of our technique, every pixel in a 64 pixel block is ranked on basis of the proportion of its PSDEs ranked in the high component. We find that the number of components \(K\) deemed by BIC to accurately model the density of PSDEs in the various blocks ranges from 4 to 6 components. To identify the most defective pixels amongst those 64 pixels, we use threshold \(th\) to identify the most defective pixels, which results in a fixed, but at the same time, a very small proportion of all the pixels in the 512 × 512 patch being identified as defective.

C. Evaluating Diagnostic Performance

To evaluate the performance of our algorithm, we use a ground truth in the form of confidence maps supplied by UKIRT. These confidence maps are for a specific detector/filter combination. As explained in [25], the confidence \(c_{ij}\) assigned to the intensity value in each pixel \(j\) of frame \(i\) is the normalized (to a median of 100%) inverse variance weight map. The same map is used to encode for hot, bad, or dead pixels by assigning zero confidence. From the confidence map provided, we create a confidence mask for both detectors 1 and 4: a mask \(m_{gt}\) containing the indices of all pixels that have been assigned zero confidence in the 512 × 512 patch. There are 549 zero confidence pixels in detectors 1s mask \(m_{gt}^s\) and 1213 zero confidence pixels in Detector 4s mask \(m_{gt}^d\). They include all kinds of defective pixels, many of which occur in clusters. In addition, Detector 4 has a bad column, two columns wide, in the patch we analyze. We then compare the list of defective pixels detected by our integrated technique to the mask. The parameter \(th\) governs the number of defective pixels selected by our integrated technique. Hence, to determine the optimum value of the parameter \(th\) and its effect on our technique’s diagnostic performance, we make use of the receiver operating characteristic (ROC) curve [57,58]. In the ROC framework, the detection performance of our integrated technique at diagnosing the defective pixels is a trade off between the hit rate (true positive rate) and the false alarm rate (false positive rate). As explained by Swets [59], the ROC curve is the only measure available that is uninfluenced by decision biases and prior probabilities, and it places the performances of diverse systems on a common easily interpreted scale. Swets explains the utility of ROC curves in analyzing the behavior of diagnostic systems. In the same vein, we draw the ROC curve for the diagnostic performance of our technique. We define the following terms.

- **True Positives (TP)**: Number of defective pixels detected by our integrated technique that are in the confidence mask.
- **False Positive (FP)**: Number of defective pixels detected by our integrated technique that are not in the confidence mask.
- **True Negative (TN)**: Number of normally functioning pixels that are not in the confidence mask and are not detected by our technique.
- **False Negative (FN)**: Number of normally functioning pixels that are in the confidence mask, but are not detected by our technique.

We estimate the true positive rate (TPR) and false positive rate (FPR) as per Eq. (10):

\[
TPR = \frac{TP}{TP + FN}, \quad FPR = \frac{FP}{FP + TN}.
\]

The two ROC curves are produced by averaging the TPR and the FPR for values of threshold \(th\) ranging

20 December 2008 / Vol. 47, No. 36 / APPLIED OPTICS 6919
from \( th = 1\% \), (corresponding to selecting 1\%) of the pixels to \( th = 95\% \) over the 10 trials, akin to the threshold averaging procedure described by Fawcett [58]. A \( th = 5\% \) threshold, for instance, implies that we select 5% of 64 pixels = 3 pixels in every block. Figures 10(a) and 10(b) show the two ROC curves, along with error bars. In Fig. 10(a) the ROC curve for Detector 1 shows that a \( th = 10\% \) threshold yields good detection performance in terms of the trade off between TPR and FPR. We also see that a defective pixel mask produced for this threshold leads to a good masking of the defective pixels during photometry in Section 8. On the other hand, for Detector 4, as per the ROC curve in Fig. 10(b) our technique performs well for a \( th = 12\% \) threshold. Detector 4 has a greater number of defective pixels in the 512 x 512 patch as compared to Detector 1, because of which, a higher threshold is required to detect all the defective pixels. What is also evident that our technique is very good at autonomously detecting the defective pixels with an accuracy (TPR) of greater than 99% in both cases. The ROC curve therefore enables us to visualize the influence of different values of threshold \( th \). By selecting the right threshold value, we can create a defective pixel mask for the detectors containing a list of the most significantly defective pixels. We also compute the area under curve (AUC) from the ROC curves [60], which is a good indicator of the performance of any diagnostic test such as ours and has been found to be statistically more discriminating as compared to accuracy [61]. The AUC from Fig. 10(a) for Detector 1 is found to be 0.9351 and for Detector 4 from Fig. 10(b) is 0.9. Comparison of the two AUC estimates clearly tells us that our technique performs slightly better at diagnosing the defective pixels of Detector 1 as compared to Detector 4.

8. Astronomical Photometry

Here we investigate the effect our defective pixel detection technique has on astronomical photometry using only the images from Detector 1. Having applied our technique to detecting the defective pixels in Detector 1, we create a defective pixel mask corresponding to a good selection of threshold \( th \), which as we saw for Detector 1 is for \( th = 10\% \). We observe the influence of this mask on the relative change of the stellar intensities in the images.
A. Preparatory Procedures

First, to measure the intensities of stars we need to correct for the different sensitivity of the individual pixels (flatfield) and to coadd the sequence of images. The observation procedure followed by the astronomers ensures that in each of the images the stars are situated in slightly different positions. We can hence create a flatfield image by determining the median pixel value at each position. This will create a flatfield image, since despite the high density of stars in this area of the sky, in a majority of the individual images the pixel values correspond to the sky background, which can be assumed to represent a homogeneously illuminated screen. After correcting for the flatfield, the images are aligned using the positions of the individual stars. Positions are measured using a fit of a Gaussian to the stellar images. Stars were then matched in each image and the individual frames are coadded. Masks, identifying potentially corrupt or otherwise influenced pixels, were also coadded in the same manner. The final coadded image was then divided by the coadded mask to correct for the different integration times at each pixel. We select the same inner 512 × 512 pixels of Detector 1 to perform our final image analysis. We identify all the stars in this area using the Source Extractor software described by Bertin and Arnouts. The software identifies regions of a minimum number of pixels that are a certain amount above the local noise level (e.g., 3σ). These regions are identified as stars and their brightness (identical to −2.5 × log (intensity)) is measured. We measure the brightness of the stars in all final coadded images that are created using the defective pixel mask, mask_{10}. The measured brightness of the stars is not calibrated, but this is not required for the analysis performed here.

B. Observation from Coadded Images

The final coadded images differ slightly depending on which mask for defective pixels has been used. Despite the high quality of the detector, a number of defective pixels are evident. In Figs. 11(a)–11(c) we show a magnification of a small region of the entire image in three different cases: for Fig. 11(a) no mask is used during the coaddition of images; for Fig. 11(b) mask_{512} is used; for Fig. 11(c) mask_{10} is used and the figures show a part of the final mosaic created. The image size is 20 in. × 20 in. on the sky, North is up and East is to the left. Stars are visible as circular dark areas in this negative presentation. One can clearly see that in the image obtained using no mask in Fig. 11(a) a large number of defective pixels occur. The pattern of the distribution of this defective pixel is caused by the dither pattern and microstepping described above. All defective pixels in this region are basically caused by one and the same small cluster of defective pixels. The dark and white, doughnut shaped object at the bottom is an artifact caused by bright stars and electronic cross talk in the read-out electronics. We see in Fig. 11(b) that the standard UKIRT mask does not eliminate all the obvious defective pixels in the image. A number of apparently high noise level pixels east (left) of the bright star actually disappear when using our mask mask_{10}. This shows that indeed they are caused by intrinsically not reliable pixels and are not created by the noise of the sky background emission.

The source detection software could identify 367 stars in the region that are 3σ above the noise level and where reliable photometry can be obtained. Given the full width at half-maximum (FWHM) of the the PSF of the stellar images of about 3 pixels, almost 4% of the area are covered by stellar images. This crowding is above the limit of about 1% or 2%, for which reliable photometry can be obtained for all stars in the field. The magnitudes of the detected stars vary by about 10 magnitudes (−2.5 log (intensity)), corresponding to a intensity ratio of the brightest to the faintest of about 10,000. The uncertainty of the intensity measurements depends strongly on the brightness of the stars. For the brightest stars (magnitude below 13), the error (error provided by the source extractor, which denotes error in the intensity measurement) is about 0.001 mag (0.092%). This error increases to 0.01 mag for stars with a magnitude between 15 and 16, and reaches about 0.1 mag (8.8%) for stars at our detection limit of 19 magnitudes. The typical errors are shown in all the graphs as solid black lines. Errors for individual stars can, of course, be larger in case they have a close-by neighbor.

We compare the intensities of the stars measured in the different images obtained using two masks mask_{512} and mask_{10} in Fig. 12(a) and 12(b), respectively. If in one of the individual images a star is influenced by a defective pixel and this defective pixel is not masked out, then its intensity in the coadded final image will be different. The amount of the intensity change will depend on the brightness of the star and how faulty a defective pixel actually is. Obviously, the fainter the star, the larger the fractional change in the measured intensity if it is hit by a defective pixel. We investigate how many stars show changes in the intensity, which are above the uncertainties in the intensity measurements itself. When comparing the intensities of the stars obtained when using no mask to using the standard UKIRT mask in Fig. 12(a), we see that six stars (one marginally) show changes in the intensity which are above the error. Most stars show almost completely unchanged intensities. In general, the change/scatter in intensity due to the masking of the defective pixels increases towards fainter stars, as expected due to their smaller intensities. Comparing the change in the intensity of the stars with no mask and our mask leads to a much increased scatter compared to the standard UKIRT masks as seen in Fig. 12(b). This is mainly caused by the larger number of defective pixels that have been masked. We also see that the scatter in brightness caused by the masking is smaller than the errors of the photometry. In other words, at most only about 10% to 15% of the stars...
change their intensity by more than their estimated intensity error. Considering the errors, a star will need to change its intensity actually by at least $2\sigma$ to consider this as significant. At most, 5% of the stars show such a change for mask $10$ in Fig. 12(b).

C. Discussion

Is the influence of applying the bad pixel mask significant for the analysis of the photometry in our images? This is, in general, a difficult question to answer, even for our particular field, and strongly depends on what one actually wants to do with the photometry. If a general analysis of all stars is required, then certainly the application of the mask does not introduce any significant effect. Note that the $1\sigma$ error bars on the magnitudes of the stars mean that only in about 60% of the cases the real magnitude of the star is within those error bars. The application of the mask shifts only 5% of the stars by more than $2\sigma$. This will hence not introduce any statistically significant change in the photometry of all stars. If one, however, is interested in high-precision photometry of a particular star in the field, then indeed the application of the mask could (with a small but nonzero probability) change the measured flux significantly. However, in such cases one ensures during the planning process of the observations that the star of interest is situated in the best and the most stable and reliable area of a detector.

9. Conclusion

In this paper, we have given insight into a novel integrated technique based on a probabilistic methodology, which can identify a multitude of different defective pixels in different kinds of image sensor arrays. First, we demonstrate its performance at the detection of the defective pixels in the CCD array of an experimental camera, where it can reliably detect such pixels from images with real scene information. Its performance in line with the performance of a ground truth, which is only applied to calibration flatfield images. Second, we demonstrate its impressive performance at the detection of the defective pixels in 2 out of the 4 high-grade detectors of the WFCAM camera. We also demonstrate the effect of the defective pixel detection on the photometry done with the images captured with those detectors, where we show that our technique can remove the cosmetic spatial errors. This can have a significant impact on high-precision photometry of a particular star in the field. Moreover, our technique is able to uncover not very subtle defects, some of which UKIRT’s bad pixel detection mask misses. The main utility of this technique lies in the fact that it can be autonomously used to generate a defective pixel mask for a detector using the raw images as opposed to using special calibration images. Once the defects are detected and a defective pixel map is created, the reported values of the defective pixels can either be ignored or they may be interpolated from their neighbors’ values. Although we have only demonstrated its performance at the detection of the defective pixels in CCD and IR arrays, it can be applied to other sensor arrays such as CMOS. This is because, in essence, this technique relies on the images captured with the detectors. Although in this paper we have only conducted trials with monochrome images, this technique can be applied to the independent R, G, B matrices of color images. In addition, our technique can potentially be improved, particularly at the density estimation stage. Although, for our purposes using EM for the GMM estimation works well, it is perhaps not optimal because of the need for a criteria such as BIC to determine the number of GMM components $K$. Recently, much improved techniques for GMM estimation have been developed that can automatically infer $K$ from the data and can potentially be experimented with [63,64]. Alternatively, using a mixture of Student–$t$ distribution as opposed to a GMM is another possibility as it has been found to be robust as they have longer tails and are less susceptible to outliers [65]. With Student–$t$ mixtures, $K$ can be automatically inferred from the data in a Bayesian setting as shown in [66,67]. However, our focus in this paper has been on proposing a technique that can autonomously identify defective pixels in detectors from real images in an almost unsupervised setting, without the need for any training, and in this regard we believe we have been successful. More importantly, we provide a more holistic treatment of the problem of defective pixel detection that exists...
in different strands of imaging that use solid-state detector arrays, unlike previous work in this regard.

References