6-25-2010

Information theoretic approach for assessing image fidelity in photon-counting imagers

Narravula R. Srikanth

Srikanth R. Narravula
Candidate

Electrical and Computer Engineering
Department

This thesis is approved, and it is acceptable in quality and form for publication:

Approved by the Thesis Committee:

Majid Hayat ,Chairperson

Sudhakar Prasad

7/8/2014

Information Theoretic Approach for Assessing Image Fidelity in Photon-counting Imagers

by

Srikanth R. Narravula

Bachelor of Technology, Electrical Engineering
Indian Institute of Technology, Madras, 2007

THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science
Electrical Engineering

The University of New Mexico

Albuquerque, New Mexico

May, 2010

Dedication

To my parents, Ramalinga Reddy and Radha Rani, for their patience and encouragement.

Acknowledgments

I want to thank my advisor, Professor Majeed M. Hayat, for his guidance and support. I have learned a lot from him during my stay here. I am grateful for having this opportunity to learn from and to work with him.

I would like to thank Professor Sudhakar Prasad for his time and input. Some of the discussions I had with him enabled me to polish my approach to solving problems in general. I would also like to thank my other thesis committee member, Professor Pradeep Sen, for his support.

I would like to thank Professor Bahram Javidi, for introducing me to the problem and also for his inputs. I would also like to thank Professor John A. Gubner for valuable discussions.

I take this opportunity to thank Jorge E. Pezoa for his help. My heartfelt gratitude to all those who helped me achieve my objective.

Information Theoretic Approach for Assessing Image Fidelity in Photon-counting Imagers

by

Srikanth R. Narravula

ABSTRACT OF THESIS

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

Electrical Engineering

The University of New Mexico

Albuquerque, New Mexico

May, 2010

Information Theoretic Approach for Assessing Image Fidelity in Photon-counting Imagers

by

Srikanth R. Narravula

Bachelor of Technology, Electrical Engineering
Indian Institute of Technology, Madras, 2007

M.S., Electrical Engineering, University of New Mexico, 2010

Abstract

The method of photon-counting integral imaging has been introduced recently for three-dimensional object sensing, visualization, recognition and classification of scenes under photon-starved conditions. In this work we present an information-theoretic model for the photon-counting imaging (PCI) method, thereby providing a rigorous foundation for the merits of PCI in terms of image fidelity. This can facilitate our understanding of the demonstrated success of photon-counting integral imaging in compressive imaging and classification. The mutual information between the source and photon-counted images is derived in a Markov random field setting and normalized by the source-image's entropy, yielding a fidelity metric that is between zero and unity, which respectively corresponds to complete loss of information and full preservation of information. Calculations suggest that the PCI fidelity metric increases with spatial correlation in source image, from which we infer that the PCI method is particularly effective for source images with high spatial correlation; the metric also increases with the reduction in photon-number uncertainty.

As an application to the theory, an image-classification problem is considered showing a congruous relationship between the fidelity metric and classifier's performance.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Three-dimensional (3D) imaging techniques have the potential for diverse applications in various fields, including medical, education, entertainment, commercial electronics, defense, communication, and manufacturing [1-5]. One of the promising methods for passive 3D sensing and visualization is integral imaging, which is based on the principle of integral photography [6-20]. In the integral imaging technique, in addition to irradiance, directional information of the rays is recorded by acquiring two-dimensional elemental images (2D projections) from different perspectives of the scene, thereby capturing depth information. Unlike holography, integral imaging can capture and reconstruct true 3D color images under ambient or incoherent light without the need for using viewing devices.

The motivation for this work comes from a special class of integral imaging, called photon-counting integral imaging, where images are formed by means of a photon-counting array. Applications for photon-counting integral imaging [21] include, low light level imaging [22, 23] single photon emission tomography [24] and astronomical imaging [25]. Photon counting for 3D object recognition, classification, and visualization with integral

imaging have been reported in [26-28].

It is intriguing that both 2D and 3D photon counting imagery capture significant portions of the information in an image even under photon-starved conditions, as demonstrated in [26-28]. This feature has become particularly evident and useful in the context of image classification, where very good performance is observed even when very few photons per pixel are available [26-28].

If we consider the physical process associated with a photon-counting imaging (PCI) system, we can identify three main components in the system (see Fig. 1.1). The first component is the true intensity image of the object. The second component is the transformation rule that governs the conversion of an intensity image into a stochastic stream of photons in space and time. The latter is characterized by a deterministic quantity called the photon-flux density [29]. The photon-flux density, ϕ, is the average number of photons per unit time and per unit area. It is well known [29] that for a coherent light source, the actual number of photons in the stochastic photon stream, present in an infinitesimal time-area rectangle $dtdA$, is a Poisson random variable with mean value $\phi dtdA$. The stochastic photon streams considered in PCI systems typically undergo severe random-deletion (or thinning) of photons due to absorption and scattering through the transmission medium. The third and last component of the PCI systems is the photon-counting array, which counts the photons impinging on each detector element. The totality of the photon counts in the array represents a degraded version of the source image, as seen by comparing the output and source images in Fig. 1.1. A key question is how to quantify the loss in image content starting from the object and ending with the degraded image generated by the photon-counting array. In light of the above three components of the PCI systems, we can identify the "source," "channel," and "output" of a communication system. As such, we can view the PCI problem shown in Fig. 1.1 as an information-theoretic problem. More precisely, the "source" is the ensemble of all possible intensity images of interest, which are selected according to a prescribed probability distribution function. The "output" is the

Figure 1.1: Schematic of the PCI system considered in this thesis. The figure shows a source image, the transformation rule and the output image. These components of the PCI system can be identified as source, channel and output of a communication system. In most scenarios of interest, the output image is a sparse, binary array since the photon stream is very weak.

ensemble of all digital photon-count images where the pixel values are non-negative integers representing the photon counts. The "channel" is simply the conditional probability that an output image is generated given that a specific source image was used.

With the above information-theoretic description, we observe that the mutual information between the source and the output of the aforementioned communication system is a measure of the change in the information present in an image prior to transmission and that present at the destination [30, 31]. Therefore, we can use the mutual information as a measure for the "similarity" between the source and output imagery. In particular, the mutual information can be utilized to quantify the preservation in image content in photon-counting imagers. More specifically, as a relative metric for image degradation we can consider the mutual information between the source and output normalized by the source's entropy, which yields a number between zero and unity, corresponding to complete loss of information and full preservation of information, respectively. For example, if the "degraded" image is statistically independent of the true image, then the metric returns a value of zero; on the other hand, if the "degraded" image is a deterministic and invertible transformation of the true image, then the metric will return a value of one. In the PCI

problem considered in this thesis, the transformation from the true image to the (degraded) elemental image is stochastic since intensity values are transformed into stochastic photon numbers, as dictated by the quantum nature of light. Moreover, the transformation that maps intensity to a photon-steam is generally non-invertible.

In this thesis, we will use the normalized mutual information metric applied to 2D elemental images in the PCI process to investigate the role of spatial correlation and photon statistics in how well image-content is preserved. Nevertheless, this model and formulation can be applied to the 3D images once we have modeled a 3D image scene in terms of the corresponding elemental images. In such a model, we need to consider the depth in addition to the intensities at a particular pixel location.

Chapter 2

Problem Formulation

2.1 Probabilistic model for PCI under Poisson photon statistics

Consider a stochastic column vector $\mathbf{X}$ whose entries, X_i, $i = 1, \ldots, n$, are discrete random variables in the interval $[0, 1]$ representing the reflectance of some unknown object or an unknown digital image. (For example, the vector $\mathbf{X}$ can be thought of as a lexicographic representation of a digital image.) Throughout this thesis, we will refer to $\mathbf{X}$ as the source image. In undertaking a digital image as the source image, we have implicitly assumed that the scene is imaged using an imaging system with a finite spatial resolution; as such, it is sufficient to consider a digital image sampled from the continuous-space acquired image with the sampling rate satisfying the Nyquist criterion consistent with the spatial resolution of the imaging system. (The quantization of levels is merely for simplicity.)

A probing beam carries an average photon flux of $\lambda\varepsilon$ photons per second per pixel, where λ is the photon flux of the un-attenuated light and $\varepsilon \in [0, 1]$ is an attenuation factor that we will use as a control parameter in this study. Clearly, the average photon flux is

reduced to $X_i\lambda\varepsilon$ upon reflection of the ith pixel of the image since each photon is reflected with a probability equal to the reflectance, X_i, associated with the ith pixel. The reflected light is detected using a detector array, operating in the photon counting mode, with quantum efficiency η; i.e., each photon is detected with probability η. Background stray light, dark-current noise, read-out noise, and other forms of noise (other than quantum noise) are ignored in this analysis.

The ith element of the detector array gives a measurement of the number of photons, Y_i, detected during integration time τ. According to the laws of photon optics for coherent light [29], conditional on a particular realization of X_i, say $X_i = x_i$, Y_i is a Poisson random variable with mean value $\eta x_i\lambda\varepsilon\tau \equiv x_i\varepsilon N_p$, where $N_p = \eta\lambda\tau$ is the mean number of photons per pixel and per unit integration time. Therefore, the *conditional* probability mass function of Y_i given that $X_i = x_i$ can be written as

$$P_{Y_i|X_i}(y_i|x_i) = \frac{(N_p x_i \varepsilon)^{y_i} e^{-N_p x_i \varepsilon}}{y_i!}, \quad y_i = 0, 1, 2, \ldots. \tag{2.1}$$

(Notation: We refer to $\mathsf{P}\{Y_i = y_i | X_i = x_i\}$, the probability that $Y_i = y_i$ given that $X_i = x_i$, as $P_{Y_i|X_i}(y_i|x_i)$; similarly, we refer to $\mathsf{P}\{\mathbf{X} = \mathbf{x}\}$ as $P_{\mathbf{X}}(\mathbf{x})$, etc.) We now define $\mathbf{Y}$ as a stochastic array whose entries are the integer-valued random variables Y_i $(i = 1, \ldots, n)$; this stochastic array represents the photon-count array. To find the probability mass function of the random-count array $\mathbf{Y}$, we observe that conditional on the gray levels $\mathbf{X} = \mathbf{x}$ of the totality of pixels, where $\mathbf{x}$ is an array with entries $x_1, \ldots, x_n$, the photon-count random variables Y_i corresponding to different pixels are statistically independent. Hence, it follows that the conditional probability mass function of the photon-count array $\mathbf{Y}$ given $\mathbf{X} = \mathbf{x}$ is

$$P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{n} P_{Y_i|X_i}(y_i|x_i), \tag{2.2}$$

where $\mathbf{y} \stackrel{\triangle}{=} [y_1, \ldots, y_n]'$ and for $i = 1, \ldots, n$, y_i is a nonnegative integer. Using the law of total probability, we can write the probability mass function $P_{\mathbf{Y}}(\mathbf{y})$ of the output image $\mathbf{Y}$ as

$$P_{\mathbf{Y}}(\mathbf{y}) = \sum_{\mathbf{x}} P_{\mathbf{X}}(\mathbf{x}) P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}), \tag{2.3}$$

where $P_{\mathbf{X}}(\mathbf{x})$ is the probability mass function of the source image $\mathbf{X}$, and the summation is over all realizations $\mathbf{x}$ of $\mathbf{X}$.

Approximation under photon-starved conditions: In situations when the attenuation factor ε is very small, the photon counts per pixel are either 0 or 1 with high probability [27]. Thus, the probability mass function of the photon count in each detector element can be approximated by a Bernoulli (binary) law. The Bernoulli law is also applicable to the scenario when the photon count is hard-limited (or thresholded) to 0 or 1. Under the Bernoulli assumption, the probability mass function of the photon count Y_i, conditional on the pixel's gray level $X_i = x_i$, takes the simpler form of

$$P_{Y_i|X_i}(0|x_i) = e^{-N_p x_i \varepsilon}, \tag{2.4a}$$

$$P_{Y_i|X_i}(1|x_i) = 1 - e^{-N_p x_i \varepsilon}. \tag{2.4b}$$

2.2 Probabilistic model for PCI under alternative photon statistics

In order to understand the role of the Poisson nature of photon numbers on the normalized mutual information, we considered two alternative models for the photon-counting channel: (1) the binomial-distribution model, and (2) the geometric-distribution model. In the former case, the probability mass function of Y_i conditional on X_i is binomial; in the latter case, the probability mass function is geometric (exponential). For consistency, we have assumed the same mean value for all three models. The binomial model is suitable in modeling photon statistics of non-classical light—when the probing light is maximally amplitude squeezed [32]. An amplitude-squeezed state can result in narrowing the distribution of the photon number, viz., reducing the photon-number uncertainty and hence reducing quantum noise below the classical shot-noise limit (associated with light with Poisson photon statistics). The geometric distribution, also termed the Boltzmann distri-

bution, is suitable for modeling the photon statistics of thermal light [29].

Binomial distribution model: The probability mass function for a binomial random variable, W, is given by

$$P_W(w) = \binom{n}{w} p^w (1-p)^{n-w}, \quad w = 0,1,2,\ldots,n, \tag{2.5}$$

where n and p are parameters; the mean of W is np. With the application of this model to the photon-count variable, Y_i, we maintain that conditional on $X_i = x_i$ $(i = 1,\ldots,n)$, Y_i is distributed according to a binomial probability mass function with mean $N_p x_i \varepsilon$. We must therefore set $np = N_p x_i \varepsilon$. This yields

$$P_{Y_i|X_i}(y_i|x_i) = \binom{n}{y_i} p^{y_i} (1-p)^{n-y_i}, \quad y_i = 0,1,2,\ldots,n, \tag{2.6}$$

where $p = N_p x_i \varepsilon / n$ is a parameter between 0 and 1. The case $p = 1$ corresponds to maximal amplitude squeezing for which the photon number is deterministic and equal to $N_p x_i \varepsilon$. Note also that as p approaches 0 (or $n \to \infty$), this binomial distribution converges to a Poisson probability mass function with mean $N_p x_i \varepsilon$. Hence, the Poisson model represents the limiting case of the binomial model.

For the case when the photon count is hard-limited to 0 or 1, we obtain

$$P_{Y_i|X_i}(0|x_i) = (1-p)^{N_p x_i \varepsilon p^{-1}} \tag{2.7a}$$

$$P_{Y_I|X_i}(1|x_i) = 1 - (1-p)^{N_p x_i \varepsilon p^{-1}}. \tag{2.7b}$$

Geometric distribution model: A geometrically distributed random variable, W, has the probability mass function

$$P_W(w) = p(1-p)^w, \quad w = 0,1,2,\ldots, \tag{2.8}$$

where p is a parameter and the mean of W is $(1-p)/p$. With the application of this model to the photon-count variable, Y_i, we maintain that conditional on $X_i = x_i$ $(i = 1,\ldots,n)$, Y_i is

distributed according to a geometric probability mass function with mean $N_p x_i \varepsilon$. We must therefore set $(1-p)/p = N_p x_i \varepsilon$, yielding $p = 1/(N_p x_i \varepsilon + 1)$ and

$$P_{Y_i|X_i}(y_i|x_i) = \frac{1}{N_p x_i \varepsilon + 1}\left(\frac{N_p x_i \varepsilon}{N_p x_i \varepsilon + 1}\right)^{y_i}, \quad y_i = 0,1,2,\ldots. \tag{2.9}$$

For the case when the photon count is hard-limited to 0 or 1, we obtain

$$P_{Y_i|X_i}(0|x_i) = \frac{1}{1+N_p x_i \varepsilon}, \tag{2.10a}$$

$$P_{Y_i|X_i}(1|x_i) = \frac{N_p x_i \varepsilon}{1+N_p x_i \varepsilon}. \tag{2.10b}$$

The probability mass functions from the three models are shown in Fig. 2.1.

Next, we use the PCI image-degradation probabilistic model described in this section to define a metric for the quality of image transmission in the PCI system based upon the mutual information between the input $\mathbf{X}$ and output $\mathbf{Y}$.

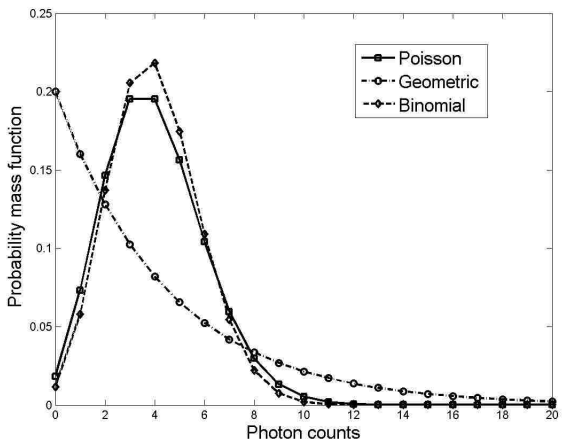


Figure 2.1: This figure shows Poisson, geometric and binomial probability mass functions. The various parameters used are $\lambda = 4$ for Poisson, $p = 0.2$ for geometric, and $n = 20, p = 0.2$ for binomial. In all three cases the mean value is 4.

2.3 Normalized mutual-information metric

We begin by reviewing the definitions of entropy and mutual information. The entropy, $H(\mathbf{X})$, of a random vector, $\mathbf{X}$, is defined as

$$H(\mathbf{X}) = -\mathsf{E}\Big[\log_2\big(P_{\mathbf{X}}(\mathbf{x})\big)\Big] = -\sum_{\mathbf{x}} P_{\mathbf{X}}(\mathbf{x})\log_2\big(P_{\mathbf{X}}(\mathbf{x})\big), \tag{2.11}$$

where "E" denotes expectation and the summation above is over all possible values that the random vector $\mathbf{X}$ can assume. The entropy $H(\mathbf{X})$ measures the amount of information conveyed in units of "bits," on average, by a random vector $\mathbf{X}$ [30]. The mutual information, $I(\mathbf{Y};\mathbf{X})$, is a measure of amount of information that one random vector, $\mathbf{Y}$, contains about another random vector, $\mathbf{X}$; it is defined by

$$I(\mathbf{Y};\mathbf{X}) = \mathsf{E}\left[\log_2\Big(\frac{P_{\mathbf{XY}}(\mathbf{xy})}{P_{\mathbf{Y}}(\mathbf{y})P_{\mathbf{X}}(\mathbf{x})}\Big)\right] = \sum_{\mathbf{y}} P_{\mathbf{Y}}(\mathbf{y})\log_2\big(P_{\mathbf{Y}}(\mathbf{y})\big) - \sum_{\mathbf{x},\mathbf{y}} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x})P_{\mathbf{X}}(\mathbf{x})\log_2\big(P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x})\big). \tag{2.12}$$

The *normalized mutual information*, denoted by ρ, is defined as

$$\rho = \frac{I(\mathbf{Y};\mathbf{X})}{H(\mathbf{X})}. \tag{2.13}$$

It can be shown that $0 \leq \rho \leq 1$ [30]. Moreover, $\rho = 1$ whenever $\mathbf{Y}$ is a deterministic, one-to-one function of $\mathbf{X}$. In particular, when $\mathbf{Y}$ can always be unambiguously transformed back to $\mathbf{X}$, the parameter ρ is at its maximum. The other extreme case is when $\mathbf{X}$ and $\mathbf{Y}$ are statistically independent, in which case $\rho = 0$. [30]. In this thesis use the terms normalized mutual information and fidelity metric interchangeably.

2.4 A classification example

Now we consider a classification example to show the application of the normalized mutual information metric defined in the previous section 2.3. We considered a $128 \times$

128 "line image," $\mathbf{X}$, as the input. The slope of the line, s, takes values from the set $\{-4,-3,-2,-1,1,2,3,4\}$ uniformly. The gray level of the pixels, representing the line, is assumed to be 255. We then added independent and identically distributed uniform noise, K, taking values in the interval $[0,k]$, $k \in [0,255]$, to the input image, $\mathbf{X}$. The addition of noise serves to reduce the spatial correlation in the image. In particular, as the value of k is increased, the variance of the noise increases reducing the the overall spatial correlation. Mathematically, conditional on the slope s, the input image pixels are independent of each other with the following probability mass function:

$$P_{\mathbf{X}|s}(\mathbf{x}|s) = \prod_{i=1}^{128}\prod_{j=1}^{128} P_{X_{i,j}|s}(x_{i,j}|s). \tag{2.14}$$

We then simulated the channel given by Eq. 2.4 to generate the photon-counted binary image $\mathbf{Y}$. To simulate the photon-starved conditions we have used $N_p\varepsilon = 0.3$. With the help of Eq. 2.2, we note that

$$P_{\mathbf{Y}|s}(\mathbf{y}|s) = \prod_{i=1}^{128}\prod_{j=1}^{128} P_{Y_{i,j}|s}(y_{i,j}|s). \tag{2.15}$$

The classification problem is then to find out whether the slope, s, of the line in the input "line image," $\mathbf{X}$, is positive or negative based on the noisy observation $\mathbf{Y}$. The classification is based upon the best least-square-error fit of a line to the noisy observation $\mathbf{Y}$. For a particular noise variance determined by k, the classifier was run for $10,000$ times, from which we estimated the classification error. The experiment was then repeated for for different values of k. Figure 2.2 shows a particular case of the photon-counted output array, $\mathbf{Y}$, for $k = 5$, and the best-fit estimate of the line based on the photon-counts. As we can see in the Table 2.1 and from Fig. 2.3, the classification error increases as we increase the noise variance, which is expected. In addition, we find that ρ decreases when the classification error increases. In other words, the fidelity metric decreases as the spatial correlation in the input image decreases. Hence, we find that there is a congruous relation between the fidelity metric and the classifier's performance. This observation indeed echoes the

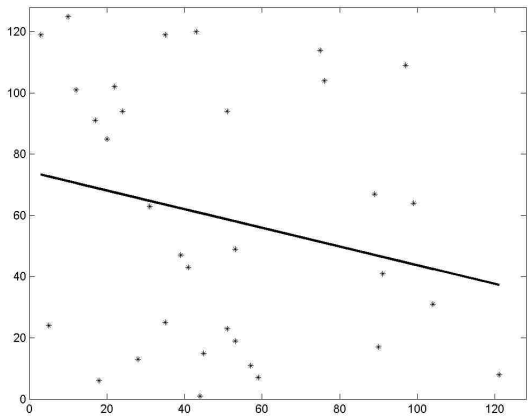


Figure 2.2: Figure showing the output 128×128 photon-counted array, where the photon counts are represented by the symbols '$*$', and the best-fit estimate of the line from the photon counts. The value of k is 5 and the corresponding noise variance is 2.08. Here $N_p\varepsilon = 0.3$.

role of Shannon's information (and channel capacity) in detection error in communication systems.

Table 2.1: Table showing the relation between the variance of the uniform noise in the input image, the normalized mutual information, ρ, and the average error of classification.

Noise variance	ρ	**Average classification error**
0.34	0.1157	0
0.75	0.1062	0.1356
3.00	0.0969	0.2218
6.75	0.0921	0.2516
18.75	0.0819	0.2912
27.00	0.0766	0.2979
33.34	0.0732	0.3015

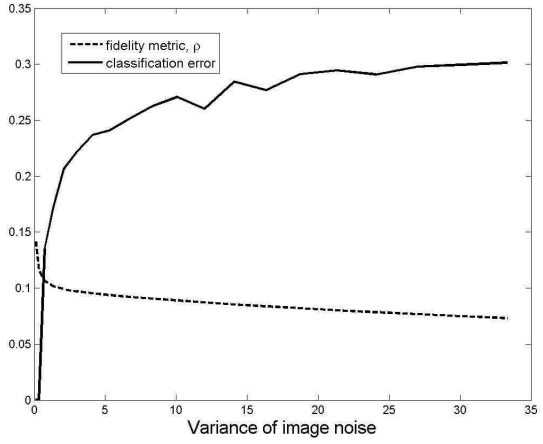


Figure 2.3: Figure showing the fidelity metric, ρ, and the classification error versus the noise variance in the input image. The classifier is run for $10,000$ times for each value of noise variance to average out the classification error. Here $N_p\varepsilon = 0.3$.

Chapter 3

Markov Models

3.1 One-dimensional case: Markov-chain model

To evaluate ρ, we need a model for the source probability distribution that can capture the correlation between the pixels. In this subsection we consider a one-dimensional image, represented by $X_1,\ldots,X_n$, which obeys a Markov-chain model. More precisely, the conditional probability mass function of a pixel given the value of an adjacent pixel is independent of all the other pixels prior to the adjacent pixel. Mathematically, we have

$$\mathsf{P}\{X_{i+1}=x_{i+1}|X_i=x_i,\ldots,X_1=x_1\}=\mathsf{P}\{X_{i+1}=x_{i+1}|X_i=x_i\},\quad i=1,\ldots,n-1. \quad (3.1)$$

For simplicity we will assume that each pixel X_i takes values from the set $\{0,1,\ldots,I\}$. The model described in (3.1) results in correlation between neighboring pixels, and the degree of correlation depends upon the specification of the transition probabilities $\mathsf{P}\{X_{i+1}=x_{i+1}|X_i=x_i\}$. In this thesis, we will assume the following form for the transition probabilities:

$$\mathsf{P}\{X_{i+1}=x_{i+1}|X_i=x_i\}=\begin{cases}\max\Big(mx_{i+1}-mx_i+c_{x_i},0\Big), & \text{for } x_{i+1}=0,1,\ldots x_i\\ \max\Big(-mx_{i+1}+mx_i+c_{x_i},0\Big), & \text{for } x_{i+1}=x_i+1,\ldots,I\end{cases},$$

(3.2)

where m is the spatial correlation index and c_{x_i} is selected so that $\mathsf{P}\{X_{i+1} = t | X_i = x_i\}$ is a probability mass function (as a function of t) for every x_i; more precisely,

$$c_{x_i} = \frac{1}{I+1}\Big(mx_i^2 - mIx_i + \frac{m}{2}I(I+1) + 1\Big). \tag{3.3}$$

Figure 3.1 depicts representative examples of the transition probabilities $\mathsf{P}\{X_{i+1} = x_{i+1} | X_i = x_i\}$ (for $x_i = 4$); these curves signify the correlation present between X_{i+1} and its neighbor X_i. As the correlation index increases from one curve to another, the spatial correlation increases between neighboring pixels. As such, by changing the correlation index m, we generate a range of correlation degrees among pixels, extending from independence ($m = 0$) to perfectly correlated ($m = \infty$). Finally, the Markov model allows us to write the probability mass function, $P_{\mathbf{X}}(\mathbf{x})$, for the image in terms of the transition probabilities and the probability mass function of X_1, which can be chosen arbitrarily,

$$P_{\mathbf{X}}(\mathbf{x}) = \mathsf{P}\{X_n = x_n | X_{n-1} = x_{n-1}\} \cdots \mathsf{P}\{X_2 = x_2 | X_1 = x_1\} P_{X_1}(x_1). \tag{3.4}$$

3.2 Two-dimensional case: Markov random field model

3.2.1 Background to MRF

A random field is a collection of random variables arranged on a lattice,

$$\mathbf{X} = \{X_{i,j}, (i,j) \in \mathbf{S}\}, \tag{3.5}$$

where $\mathbf{S}$ is a rectangular array of sites (lattice):

$$\mathbf{S} = \{(i,j) | 1 \le i \le N, 1 \le j \le N\}. \tag{3.6}$$

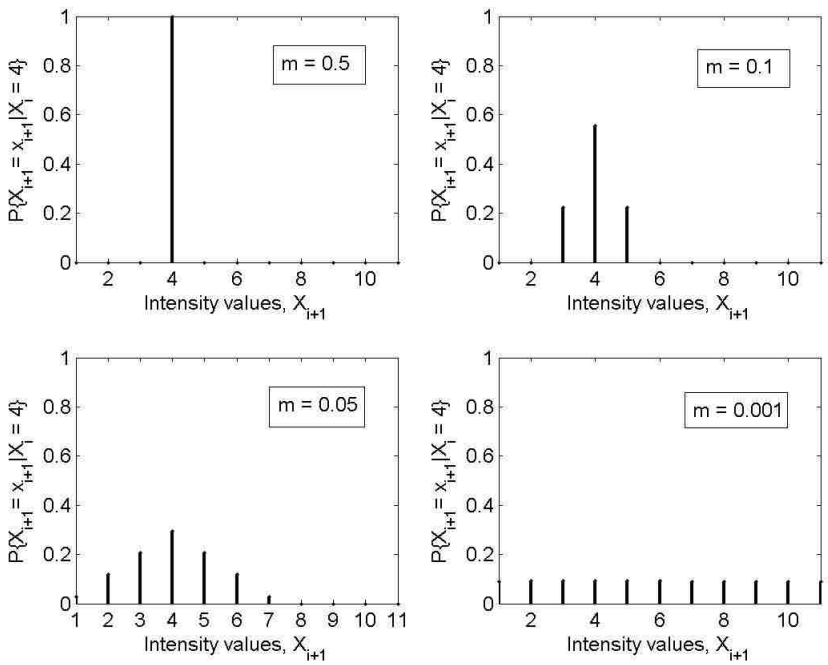


Figure 3.1: Representative spatial conditional probability mass function, $\mathsf{P}\{X_{i+1} = x_{i+1}|X_i = 4\}$, for different values of the correlation index m, which determines the amount of correlation present among source-image pixels.

A local neighborhood $\mathbf{N}_{i,j}$ of the site (i,j) can be defined as

$$\mathbf{N}_{i,j} = \{(k,l) \in \mathbf{S} : D[(i,j),(k,l)] \leq K, (k,l) \neq (i,j)\}, \tag{3.7}$$

where K is an integer representing the order of the neighborhood and D is some measurement of distance. A neighborhood system $\mathbf{N}$ for $\mathbf{S}$ is defined as

$$\mathbf{N} = \{\mathbf{N}_{i,j}, (i,j) \in \mathbf{S}\}. \tag{3.8}$$

Some typical neighborhood systems are shown in Fig 3.2(a).

With a neighborhood system $\mathbf{N}$ defined on the lattice $\mathbf{S}$, we can restate the Markov

property for a MRF, $\mathbf{X}$. By defining

$$\mathbf{X}_{i,j}^{c} \stackrel{\triangle}{=} \{X_{k,l}, (k,l) \in \mathbf{S} \setminus (i,j)\}, \tag{3.9}$$

the Markov property can be stated as

$$\mathsf{P}\{X_{i,j} = x_{i,j} | \mathbf{X}_{i,j}^{c}\} = \mathsf{P}\{X_{i,j} = x_{i,j} | X_{k,l} = x_{k,l}, (k,l) \in \mathbf{N}_{i,j}\}. \tag{3.10}$$

From the Hammersley-Clifford theorem [33], we have an equivalence between a MRF and a Gibbs random field (GRF). The benefit of this theorem is that it allows the probability distribution to be stated explicitly.

A GRF is defined by Gibbs distribution:

$$P_{\mathbf{X}}(\mathbf{x}) = \frac{1}{Z} \exp\Big(-\frac{U(\mathbf{x})}{T}\Big), \tag{3.11}$$

where Z is a normalizing constant known as the partition function and it can be written as

$$Z = \sum_{\mathbf{x} \in \Omega} \exp\Big(-\frac{U(\mathbf{x})}{T}\Big), \tag{3.12}$$

T is a constant called temperature, $\mathbf{x}$ is an image and U is the energy function. The energy

$$U(\mathbf{x}) = \sum_{c \in C} V_c(\mathbf{x}) \tag{3.13}$$

is a sum of clique potentials, V_c over all possible cliques C associated with a neighborhood system. A clique, c, is a subset of sites in which every pair of distinct sites are neighbors. Some typical cliques are shown in Fig 3.2(b) (By definition, every set containing only one site is also a clique.). If the clique potential, $V_c(\mathbf{x})$, is independent of the position of the clique in the lattice, then we call such an MRF a homogeneous MRF. Figure 3.2(b) shows all the cliques that are present in a second-order neighborhood. In a first-order neighborhood, we have two categories of cliques based on the number of points they contain, as shown by the first three shapes of cliques (from top, left corner) in Fig. 3.2(b). In the second-order neighborhood, we have four categories and a total of ten shapes of cliques.

In this thesis, we have considered a commonly used special case of a homogeneous MRF termed the Ising model, [33]. A generalized Ising model is characterized by its clique potential function, which has the following form:

$$V_c(\mathbf{X}) = \begin{cases} \beta_c & \text{if the pixel values of } \mathbf{X} \text{ at the sites in } c \text{ are same} \\ -\beta_c & \text{otherwise.} \end{cases} \tag{3.14}$$

Here, β_c is a real number and it depends on the type of the clique; this parameter controls the spatial correlation in the image.

Finally, it is known that for an image modeled by a homogeneous MRF, it is sufficient to consider $H(\mathbf{X})$ and $I(\mathbf{Y};\mathbf{X})$ for a neighborhood of a pixel [31]. Here we justify the result for homogeneous Markov chains.

We call a Markov chain a homogeneous Markov chain if

$$\mathsf{P}\{X_j = x_j | X_{j-1} = x_{j-1}\} = \mathsf{P}\{X_i = x_i | X_{i-1} = x_{i-1}\} \quad \forall i, i \in \{1,2,\ldots,n\} \tag{3.15}$$

From Eq. (3.1) and (2.11) we can then write the entropy for a homogeneous Markov chain as

$$\begin{aligned} H(\mathbf{X}) &= \sum_{x_i} \cdots \sum_{x_n} P_{X_1,\ldots,X_n}(x_1,\ldots,x_n) \log_2 P_{X_1,\ldots,X_n}(x_1,\ldots,x_n) \\ &= \sum_{x_i} \cdots \sum_{x_n} P_{X_1,\ldots,X_n}(x_1,\ldots,x_n) \log_2 \big(P_{X_2|X_1}(x_2|x_1)\big)^{(n-1)} P_{X_1}(x_1) \\ &= (n-1) \sum_{x_1} \sum_{x_2} P_{X_1,X_2}(x_1,x_2) \log_2 P_{X_2|X_1}(x_2|x_1) + \sum_{x_1} P_{X_1}(x_1) \log 2 P_{X_1}(x_1) \\ &= (n-1) H(X_2|X_1) + H(X_1) \end{aligned} \tag{3.16}$$

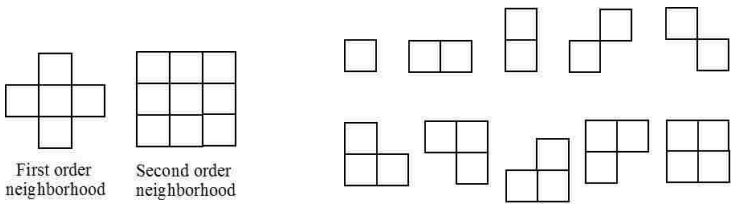
Hence, for large n, we can take the conditional entropy, $H(X_2|X_1)$, to be a good measure of the the entropy of the homogeneous Markov chain.

In the 2D case, the image spatial entropy is defined as [31]:

$$H\big(X_{i,j} | \mathbf{X}_{k,l}, (k,l) \in \mathbf{N}_{i,j}\}\big) = \sum_{x_{i,j}} \sum_{\mathbf{n}_{i,j}} \mathsf{P}\{X_{i,j} = x_{i,j}, \mathbf{N}_{i,j} = \mathbf{n}_{i,j}\} \log_2 \big(\mathsf{P}\{X_{i,j} = x_{i,j} | \mathbf{N}_{i,j} = \mathbf{n}_{i,j}\}\big),$$

(3.17)

where $\mathbf{n}_{i,j}$ is a particular configuration of the neighborhood. For a homogeneous MRF, the image spatial entropy defined by Eq. (3.17) is going to be the same irrespective of the pixel location (i, j). Hence we use this as a measure of the entropy for the image.



(a) (b)

Figure 3.2: 3.2(a) First-order and second-order neighborhoods (of the center pixel) considered in this thesis. 3.2(b) Cliques of various sizes up to four-point sites. The first three cliques shown (from top-left to top-right) correspond to a first-order neighborhood; all ten cliques correspond to the second-order neighborhood system.

3.3 Computing the fidelity metric

For a particular MRF model, there can be many neighborhood configurations which will yield the same energy. Following [31], let us denote the set of all configurations of the neighborhood which result in same energy (and there by same conditional probability) as α_x, and the state of the neighborhood by the random variable A_X. Now we can write the conditional entropy of a pixel given its neighborhood as [31]

$$H\big(X_{i,j}|\mathbf{X}_{k,l},(k,l)\in\mathbf{N}_{i,j}\big)=-\sum_{x_{i,j}}\sum_{\alpha_x}\mathsf{P}\{A_X=\alpha_x,X_{i,j}=x_{i,j}\}\log_2\big(\mathsf{P}\{X_{i,j}=x_{i,j}|A_X=\alpha_x\}\big), \tag{3.18}$$

where $\mathsf{P}\{A_X=\alpha_x,X_{i,j}=x_{i,j}\}$ is the joint probability of a site taking value $x_{i,j}$ and its neighbors being in state α_x and $\mathsf{P}\{X_{i,j}=x_{i,j}|A_X=\alpha_x\}$ is the conditional probability of a

site taking value $x_{i,j}$ knowing its neighbors are in state α_x. By approximating the probabilities with their corresponding histograms, $f_{A_X}(\alpha_x)$ and $f_{A_X X_{i,j}}(\alpha_x, x_{i,j})$, we can recast the conditional entropy as [31]

$$H\big(X_{i,j}|\mathbf{X}_{k,l},(k,l)\in\mathbf{N}_{i,j}\big) = -\sum_{x_{i,j}}\sum_{\alpha_x} f_{A_X X_{i,j}}(\alpha_x, x_{i,j}) \log_2\left(\frac{f_{A_X X_{i,j}}(\alpha_x, x_{i,j})}{f_{A_X}(\alpha_x)}\right). \quad (3.19)$$

Similarly, we can define the mutual information as [31]

$$I\big(X_{i,j};Y_{i,j}|\mathbf{X}_{k,l},\mathbf{Y}_{k,l},(k,l)\in\mathbf{N}_{i,j}\big) = \quad (3.20)$$
$$\sum_{x_{i,j}}\sum_{y_{i,j}}\sum_{\alpha_x}\sum_{\alpha_y} f_{A_X X_{i,j} A_Y Y_{i,j}}(\alpha_x, x_{i,j}, \alpha_y, y_{i,j}) \log_2\left(\frac{f_{A_X X_{i,j} A_Y Y_{i,j}}(\alpha_x, x_{i,j}, \alpha_y, y_{i,j}) f_{A_X}(\alpha_x) f_{A_Y}(\alpha_y)}{f_{A_X A_Y}(\alpha_x, \alpha_y) f_{A_X X_{i,j}}(\alpha_x, x_{i,j}) f_{A_Y Y_{i,j}}(\alpha_y, y_{i,j})}\right)$$

with similar interpretations of the histograms.

From Equations (3.19) and (3.20) we can calculate the metric ρ. For example, with the generalized Ising model, as given in (3.14), and a second order neighborhood with clique potentials assigned to the cliques as in Section 4.2, the number of states required in the estimation of entropy and mutual information is only nine. The possible energy levels for any of the eight cliques are β and $-\beta$. Therefore, the set of all states a second order neighborhood can take is $\{-8\beta,-6\beta,-4\beta,-2\beta,0,2\beta,4\beta,6\beta,8\beta\}$. Therefore we may consider only 9 states instead of 8^8 configurations, which simplifies the required calculations significantly.

Chapter 4

Results

4.1 Discussion of the normalized mutual information metric: Markov chain model

We now evaluate the normalized mutual information metric, ρ, assuming the above described Markov-chain model, section 3.1, for different values of the correlation index, m, and also for different channel probability distributions. In our calculations we have assumed that the first pixel, X_1, is uniformly distributed. Figure 4.1 shows ρ as a function of the transmission probability, ε, for different channel conditional distributions for a particular spatial correlation index, $m = 0.01$. We find that the case of Poisson photon statistics yields a higher normalized mutual information than that for the Boltzmann (geometric) photon statistics. We also observe that, as expected, the normalized mutual information from the binomial photon-statistics case converges to that for the Poisson statistics as the parameter p of the binomial distribution tends to zero (while keeping its mean fixed).

Figure 4.2 shows ρ as a function of the transmission probability, ε, parameterized by different spatial correlation indices, m, for both the Poisson and geometric photon statis-

tics. Here too we find that the Poisson-statistic model offers higher fidelity metric, ρ, than that offered by the geometric photon-statistics model. Notably, it is seen that fidelity metric increases with spatial correlation in the source image. In particular, Fig. 4.2 suggests that the highly correlated case, corresponding to $m = 0.5$, yields higher fidelity metric than that offered by the nearly independent-pixel case ($m = 0.0005$).

In summary, we draw the following observations from the results. First, the fidelity metric ρ increases as the variance of the photon number decreases (from that corresponding to a Boltzmann distribution, to a Poisson distribution, and finally to a binomial distribution). This is expected since a reduced quantum noise implies that the photon count resembles the image more closely and, in turn, implying higher correlation between the source image and the photon-counted output image. Second, a more important observation is that ρ increases as the spatial correlation in the image becomes stronger. This suggests that the ability of the PCI approach to retain spatial information improves with the spatial correlation in the source image. Namely, the PCI approach seems to be inherently geared toward "images" rather than individual pixels. These observations are confirmed in the 2D simulations considered next.

4.2 Discussion of the normalized mutual information metric in the 2D setting

We have generated MRF images according to the generalized Ising model described in (3.14) with the following specification of the parameter β_c. For any one-site clique c, $\beta_c = 1$; for any two-site clique c, $\beta_c = \beta$, a constant. Finally, for all three-site and four-site cliques, $\beta_c = 0$. We followed the Metropolis sampling algorithm [33] to generate 3-bit images of size 128×128 with varying spatial-correlation parameter, β. (The temperature parameter, T, as discussed in Appendix 3.2.1, is set to 3.) The algorithm is run for 1000

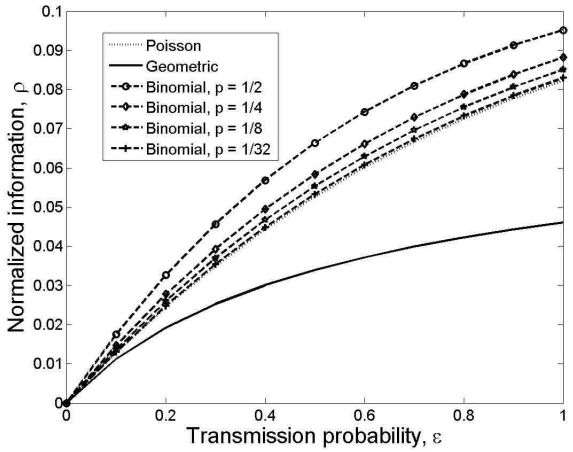
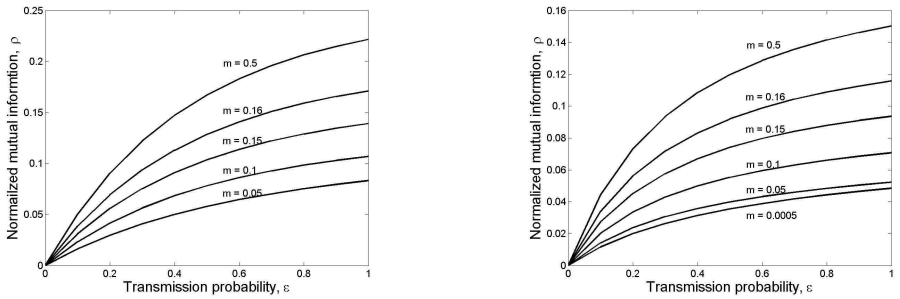


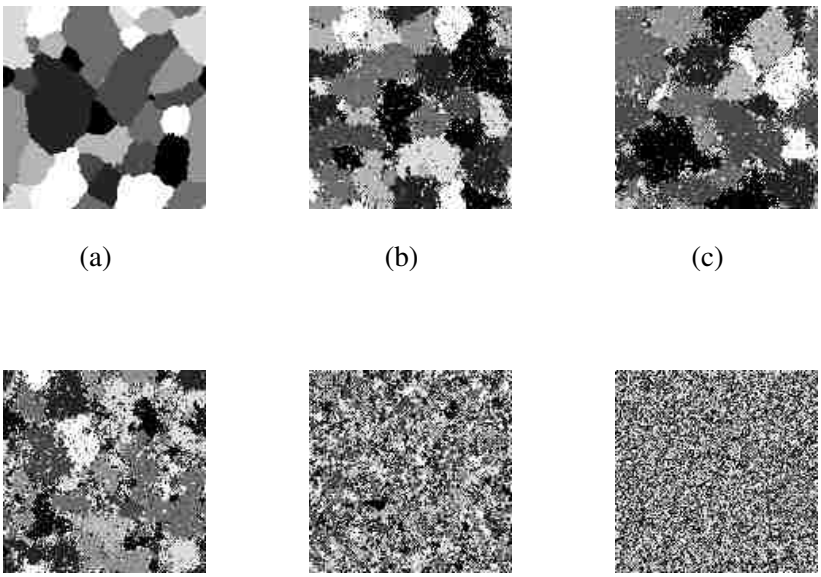
Figure 4.1: Normalized mutual information for different possible channel distributions while using a Markov-chain model for $P_{\mathbf{X}}(\mathbf{x})$, with a specific correlation index, $m = 0.01$. Here we used $N_p = 3$.



(a) (b)

Figure 4.2: Family of curves showing that presence of spatial correlation in the source image results in an increase in the normalized mutual information for (a) the Poisson photon-counting channel and (b) the geometric photon-counting channel. Here we used $N_p = 3$.

iterations for each image generated; examples are shown in Fig. 4.3 showing the change in spatial correlation in the image as β is varied.



(a) (b) (c)

(d) (e) (f)

Figure 4.3: Sample MRF images generated according to a generalized Ising model with potential energies given by (3.14) using various values for the parameter β. We employed a Metropolis sampler using 1000 iterations. The values used for the parameter β are (a) $\beta = -2$, (b) $\beta = -1$, (c) $\beta = -0.975$, (d) $\beta = -0.95$, and (e) $\beta = -0.9$, (f) $\beta = -0.4$.

To quantify the spatial correlation present in the source images, we estimated the auto-covariance of the image. More precisely, for each selection of the parameter β we generated 50 realizations of the image. We then formed the covariance matrix corresponding to the 50 sample images and generated the L_1 norm of it, which is a measure of spatial correlation. Next, we normalized the spatial correlation in the image by that for a constant (perfectly correlated image), which resulted in a number between 0 and 1. We took this number, termed the *correlation metric* as a measure of the amount of spatial correlation present in a the MRF image.

From Table 4.1, we can see that as the parameter β decreases, the correlation metric increases. In particular, from Fig. 4.4, we observe that, when β increases from -1 to

-0.9, the correlation metric drops rapidly. This result can also be seen in from Figure 4.3: as β increases, the image begins to loose its spatial structure and begins to resemble white noise. We calculated ρ as described in section 3.3. It is also evident that as the the correlation metric increases (by increasing the parameter β), the fidelity metric, ρ, also increases.

Next, as in the case of the 1D Markov-chain model, we plot ρ as a function of transmission probability, ε, parameterized by β, as shown in Fig. 4.5(a). It is seen that ρ increases monotonically with ε. Moreover, for a fixed ε, ρ increases as β is decreased (i.e., as correlation metric is increased). To see the dependence of ρ on spatial correlation of the image more clearly, we plotted ρ as a function of the correlation metric, as shown in Fig 4.5(b). It is seen that there is a nearly linear relationship between the correlation metric and ρ.

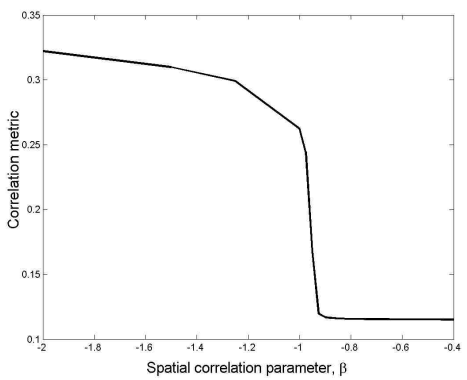
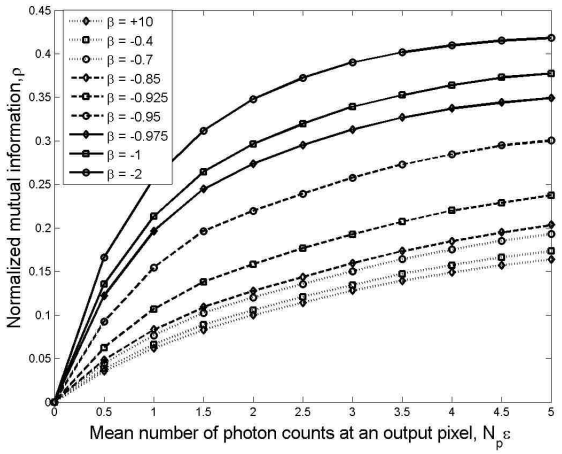


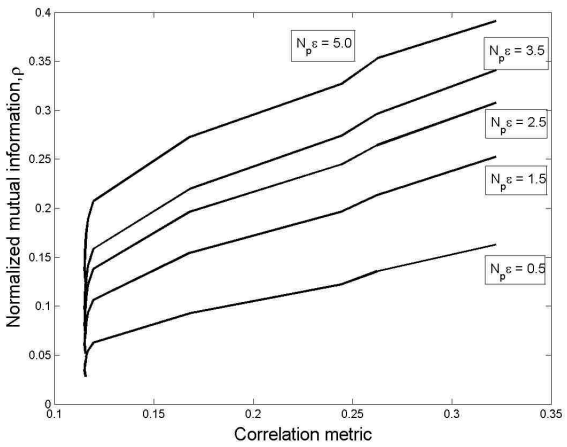
Figure 4.4: Figure showing the relation between correlation metric, and the spatial correlation parameter β.

Table 4.1: Dependence of the spatial-correlation and fidelity metrics on the parameter β for the case $N_p\varepsilon = 3$.

β	**Correlation Metric**	ρ
-2	0.32	0.36
-1.25	0.29	0.35
-1	0.26	0.31
-0.975	0.24	0.29
-0.95	0.16	0.23
-0.925	0.119	0.17
-0.9	0.1169	0.15
-0.85	0.1161	0.14
-0.8	0.1158	0.13
-0.7	0.1155	0.126
-0.6	0.11544	0.120
-0.5	0.11535	0.116
-0.4	0.11531	0.114
+10	0.1152	0.112



(a)



(b)

Figure 4.5: Figure showing the relation between correlation metric, ρ and the mean number of photon counts at the out put per pixel per unit integration time.

Chapter 5

Conclusions

In this thesis we have viewed the PCI approach as a communication system over a stochastic channel; this allowed us to model and analyze the performance of the PCI method within a rigorous information-theoretic framework. In our model, we regard the source as the ensemble of all possible intensity images of interest with a known probability distribution function and the output as the ensemble of all digital photon-count images. The channel is governed by laws of statistical optics and photodetection, which together allow us to characterize the conditional probability that an output image is generated given that a specific source image has been used. Normalized mutual information between the source and the output images was proposed as a fidelity metric, measuring the loss of information content in the output image relative to the source image. The analysis specifically captures the role of spatial correlation present in the source image by means of Markov image models. Calculations suggest that the effectiveness of the imaging PCI approach is enhanced with the presence of spatial correlation in the source image. In addition, we examined the performance of the PCI approach under Poisson and alternative photon statistics (Boltzmann and photon-number squeezed distributions). Calculations suggest that the performance improves as the variance of the photon number is reduced, which is consistent with our understanding of the role of quantum noise in imaging.

While this thesis considers only the analysis of elemental images in the PCI approach, our model can be extended in a straightforward manner to sequences of elemental images and 3D images by expanding the sizes of the vectors. However, we believe that the insight brought about by our analysis of elemental images can justify the tenet of the PCI approach as a compressive sensing tool. There are many possible directions that can be pursued based upon the foundational work provided in this thesis. For example, since image compression is an inherent property of the PCI approach, it will be interesting to look into the fundamental tradeoff between compression and preserving 3D image fidelity in the PCI approach as well as compressive 3D imaging and visualization. It is also possible to employ recognition-specific metrics such as the Mahalanobis and Bhattacharya distances to further characterize the use of PCI-generated imagery in object classification.

Appendices

Appendix A

Matlab codes

A.1 Code for the classification example

```
% We generate two types of lines here. We add noise to it. From them we
% find the simulated photon count output image. Then we will find the best
% fit to the data points of the output. We then find the error that is
% generated between the data points and the 'line-fit'. We see that as the
% variance of the noise added to it increases, the error increases. We also
% find the 'normalized mutual information metric' (Rho) of the images and
% expect an inverse proportional relation between the error and Rho.
clear all
clc
N = 128;
Np = 0.3;
S = 5000; % number of times the classifcation is done
g = 255; % number of gray levels in the input image would be g+1, [0, 255]
n1 = 0;
```

```
for n2 = 1:30 % this controls the variance of the uniform independent noise
    for j = 1:S % this is the number of times we are simulating the program so that we can
average the classification error.
        X = zeros(N,N);
        %Here we define slope and intercept of the line. If the slope'm' is
        %1, then we define 'c' to be high up (for example c = 2). The line
        %looks like a top-left (TL) to bottom-right (BR) type of a line. If
        %the slope 'm' is -1, then we define 'c' to be low down (for
        %example c = 125). The line looks like a bottom-left (BL) to
        %top-right (TR) type od a line. NOTE: 'c' is not y-intercept, but
        %it is x-intercept. When we later find the equation of the line
        %from the data points, we actually find the y-intercept.
        a = -4;
        b = 4;
        m = floor(a + (b-a)*rand) ;% slope of the line which is random between [-a,b-1]
        % Now, y intercep in y = m*x+c
        if m<0
            c = 125;
        elseif m == 0
            m = 4;
            c =2;
        else
            c = 2;
        end
        for i = 1:N
            if ((0 < m*i+c) && (m*i+c < N) && (m>0))
                X(i,m*i+c) = g; % Use this to generate TL to BR type of a line
            elseif ((0 < m*i+c) && (m*i+c < N) && (m<0))
```

```
            X(m*i+c,i) = g; % Use this to generate BL to TR type of a line
        end
    end
    % Now we add noise to the image, pixel wise. This noise is uniform
    % between n1 and n2. If we increase n2-n1, we are automatically
    % increasing the variance of the noise.
    noise_l = n1 + floor((n2-n1)*rand(N)); % We truncate the numbers as they have to
correspond to a gray level
    Xn = X+noise_l;
    % here we make sure the grey levels are in [0,g].
    k = find(Xn>g);
    Xn(k) = g;
    l = find(Xn<0);
    Xn(l) = 0;

    Y = zeros(N,N);
    for i = 1:(N*N) % runs through the entire input image file, X
        P = exp(-Np*((Xn(i)-1)/(g+1))); % count zero channel probability
        p = rand;
        Y(i) =  (p<P); % selects zero with probability P.
    end
    % Now fitting a line to the data points Y.
    [x,y] = find(Y ==1);
    D(:,:) = [x,y];
    temp = polyfit(x,y,1);
    a1 = temp(2); % y-intercept of the fitted line
    a2 = temp(1); % slope of fitted lines
    fit = a1+a2*x;
```

```
        % We now calculate the classification error. We check the slope of
        % the line 'a2' and find out whether the line is TL-BR or BL-TR.
        Test(j) = m*a2;
        % If 'Test' is positive then we are classifying without any error.
        % If it is negative then it means that we are saying that the
        % estimate 'a2' is negative when actually the slope 'm' is
        % positive. It would then be a classification error.
        clear x y D;
    end
    % Here we calculate the normalized mutual information metric in the
    % classification example of two lines. This is the actual calculation,
    % not estimating entropies. These are actually conditional
    % calculations, conditional on the slope of the image.
    n = n2-n1+1; % support of the uniform noise
    Hx = log2(n); % conditional entropy of a pixel
    xi = n1:n2; % this is used to calculate P(y=0—m)
    Py_0 = (1/n)*sum(exp(-Np*xi));
    Py_1 = 1-Py_0;
    Hy = -Py_0*log2(Py_0)-Py_1*log2(Py_1);
    temp0 = exp(-Np*xi);
    temp1 = 1-temp0;
    q = find(temp0 ==0);
    temp0(q) = 1;
    r = find(temp1 ==0);
    temp1(r) = 1;
    Hyx = -(1/n)*sum(temp0.*log2(temp0) + (temp1).*log2(temp1));
    Ixy = Hy-Hyx;
    Rho(n2) = Ixy/Hx; % this is the normalized mutual information metric
```

```
    var(n2) = (n2-n1)^2/12; % this is the noise variance that we are adding
    k = find(Test < 0);
    s = size(k);
    ave_error(n2) = s(2)/S; % this is the average classification error
end
T(1,:) = var(:);
T(2,:) = Rho(:);
T(3,:) = ave_error(:);
figure; plot(T(1,:),T(2,:), 'r'); hold on; plot(T(1,:),T(3,:), 'g')
```

A.2 Markov chains

A.2.1 Conditional probability function

```
 % This function calculates the conditional probabilities in the Markov
% chain model: P = P(X2—X1)
function [P] = cond_prob(x,th,I)
P = zeros(1,I);
P(1:x) = tan(th)*[1:x] + ((2/(I+1)) - tan(th)*x);
P(x+1:I) = -tan(th)*[x+1:I] + ((2/(I+1)) + tan(th)*x);
k = find(P < 0);
P(k) = 0;
P = P/sum(P); % normalizing the probability
```

A.2.2 Calculation of ρ for a Markov chain

```
% Here we generate plots of Rho = Hx/Ixy with 'f' (fraction of intensity
% coming out) in a Markov chain setting. Hx –> Entropy of the input, X,
% which is a Markov chain of size, N. Hy –> Entropy of the photon-counted
% output, Y. Ixy –> Mutual information between par –> partially
% correlated; refers to the correlation in X.
clear all
clc
% Initializing the variables.
N = 4; % no. of pixels
I = 10; % no. of irradiance values
Bp = [1/1, 1/(2), 1/(4), 1/(8), 1/(16), 1/(32), 1/(64)]; % the values of 'p' in the binomial
channel distribution
sbp=size(Bp);
F = 0:0.1:1; % the values of 'f', transmission probability
Hx_par = zeros(1, I+1); % Entropy of the input
% 'parp' refers to partially correlated, Poisson channel, 'parg' refers to
% partially correlated, geometric channel and 'parb' refers to partially
% correlated binomial channel.
Hy_parp = zeros(1, I+1);
Hyx_parp = zeros(1, I+1);
Ixy_parp = zeros(1, I+1);
Hy_parg = zeros(1, I+1);
Hyx_parg = zeros(1, I+1);
Ixy_parg = zeros(1, I+1);
Hy_parb = zeros(sbp(2), I+1);
Hyx_parb = zeros(sbp(2), I+1);
```

```
Ixy_parb = zeros(sbp(2), I+1);
b = 0;
%theta = [0.5, 0.01]
theta = 0.01; % 'theta' controls the spatial correlation in Px.
b = b+1;
% This code can be used to calculate Hx/Ixy as a function of 'theta'.
Px = zeros(I+1,I+1,I+1,I+1);
Py_p = zeros(2,2,2,2); % the photon-counts of 0 and 1; two values
Pyx_p = zeros(2,2,2,2,I+1,I+1,I+1,I+1);
Py_g = zeros(2,2,2,2);
Pyx_g = zeros(2,2,2,2,I+1,I+1,I+1,I+1);
Py_b = zeros(sbp(2),2,2,2,2);
Pyx_b = zeros(sbp(2),2,2,2,2,I+1,I+1,I+1,I+1);
for x1 = 1:1:I+1
    P2g1 = cond_prob(x1,theta,I+1);
    for x2 = 1:1:I+1
        P3g2 = cond_prob(x2,theta,I+1);
        for x3 = 1:1:I+1
            P4g3 = cond_prob(x3,theta,I+1);
            for x4 = 1:1:I+1
                Px(x1,x2,x3,x4) = P4g3(x4)*P3g2(x3)*P2g1(x2)/(I+1);
            end
        end
    end
end
P = Px(:);
jlt = find(P == 0);
P(jlt) = 1;
```

```
Hx = - sum(P.*log2(P));
Hx_par(b,:) = Hx_par(b,:) + Hx;
c = 0;
for f=F
    c=c+1;
    Np=3*f; % the average number of photons in the input image scene
    % Now we calculate the conditional pdfs and then the marginals which
    % are used to calculate entropies and thereby mutual information
    for c1=0:1
        for c2=0:1
            for c3=0:1
                for c4=0:1
                    for x1=1:1:I+1
                        for x2=1:1:I+1
                            for x3=1:1:I+1
                                for x4=1:1:I+1
                                    Pyx_p(c1+1,c2+1,c3+1,c4+1,x1,x2,x3,x4)=...
                                    ((1-c1)*exp(-Np*((x1-1)/(I+1)))+...
                                    c1*(1-exp(-Np*((x1-1)/(I+1)))))*...
                                    ((1-c2)*exp(-Np*((x2-1)/(I+1)))+...
                                    c2*(1-exp(-Np*((x2-1)/(I+1)))))*...
                                    ((1-c3)*exp(-Np*((x3-1)/(I+1)))+...
                                    c3*(1-exp(-Np*((x3-1)/(I+1)))))*...
                                    ((1-c4)*exp(-Np*((x4-1)/(I+1)))+...
                                    c4*(1-exp(-Np*((x4-1)/(I+1)))));
                                    Pyx_g(c1+1,c2+1,c3+1,c4+1,x1,x2,x3,x4)=...
                                    ((1-c1+Np*c1*((x1-1)/(I+1)))/(1+Np*((x1-1)/(I+1))))* ...
                                    ((1-c2+Np*c2*((x2-1)/(I+1)))/(1+Np*((x2-1)/(I+1))))* ...
```

```
((1-c3+Np*c3*((x3-1)/(I+1)))/(1+Np*((x3-1)/(I+1))))* ...
((1-c4+Np*c4*((x4-1)/(I+1)))/(1+Np*((x4-1)/(I+1))));
a=0;
for p=Bp
    a=a+1;
    Pyx_b(a,c1+1,c2+1,c3+1,c4+1,x1,x2,x3,x4)=...
    ((1-c1)*((1-p)^((Np/p)*((x1-1)/(I+1))))+...
    c1*(1-((1-p)^((Np/p)*((x1-1)/(I+1))))))*...
    ((1-c2)*((1-p)^((Np/p)*((x2-1)/(I+1))))+...
    c2*(1-((1-p)^((Np/p)*((x2-1)/(I+1))))))*...
    ((1-c3)*((1-p)^((Np/p)*((x3-1)/(I+1))))+...
    c3*(1-((1-p)^((Np/p)*((x3-1)/(I+1))))))*...
    ((1-c4)*((1-p)^((Np/p)*((x4-1)/(I+1))))+...
    c4*(1-((1-p)^((Np/p)*((x4-1)/(I+1))))));
    if Pyx_b(a,c1+1,c2+1,c3+1,c4+1,x1,x2,x3,x4)==0
        Hyx_parb(a,c)=Hyx_parb(a,c);
    else
        Hyx_parb(a,c)=Hyx_parb(a,c)-...
        Pyx_b(a,c1+1,c2+1,c3+1,c4+1,x1,x2,x3,x4)*...
        Px(x1,x2,x3,x4)*...
        log2(Pyx_b(a,c1+1,c2+1,c3+1,c4+1,x1,x2,x3,x4));
    end
end
if Pyx_p(c1+1,c2+1,c3+1,c4+1,x1,x2,x3,x4)==0
    Hyx_parp(b,c)=Hyx_parp(b,c);
else
    Hyx_parp(b,c)=Hyx_parp(b,c)-...
    Pyx_p(c1+1,c2+1,c3+1,c4+1,x1,x2,x3,x4)*...
```

```
                    Px(x1,x2,x3,x4)*...
                    log2(Pyx_p(c1+1,c2+1,c3+1,c4+1,x1,x2,x3,x4));
                end
                if Pyx_g(c1+1,c2+1,c3+1,c4+1,x1,x2,x3,x4)==0
                    Hyx_parg(b,c)=Hyx_parg(b,c);
                else
                    Hyx_parg(b,c)=Hyx_parg(b,c)-...
                    Pyx_g(c1+1,c2+1,c3+1,c4+1,x1,x2,x3,x4)*...
                    Px(x1,x2,x3,x4)*...
                    log2(Pyx_g(c1+1,c2+1,c3+1,c4+1,x1,x2,x3,x4));
                end
            end
        end
    end
end
K = Pyx_p(c1+1,c2+1,c3+1,c4+1,:);
K = K(:);
L = Px(:);
Py_p(c1+1,c2+1,c3+1,c4+1) = sum(K.*L);
KG = Pyx_g(c1+1,c2+1,c3+1,c4+1,:);
KG = KG(:);
LG = Px(:);
Py_g(c1+1,c2+1,c3+1,c4+1) = sum(KG.*LG);
for k = 1:sbp(2)
    KB = Pyx_b(k,c1+1,c2+1,c3+1,c4+1,:);
    KB = KB(:);
    LB = Px(:);
    Py_b(k,c1+1,c2+1,c3+1,c4+1) = sum(KB.*LB);
```

```
                    end
                end
            end
        end
    end
    Hy_parp(b,c) = -sum(Py_p(:).*log2(Py_p(:)));
    Hy_parg(b,c) = -sum(Py_g(:).*log2(Py_g(:)));
    for l = 1:sbp(2)
        Hy_parb(l,c) = -sum(Py_b(l,:).*log2(Py_b(l,:)));
    end
end
Hy_parp(b,1) = 0;
Ixy_parp(b,:) = Hy_parp(b,:) - Hyx_parp(b,:);
Hy_parg(b,1) = 0;
Ixy_parg(b,:) = Hy_parg(b,:) - Hyx_parg(b,:);
for m = 1:sbp(2)
    Hy_parb(m,1) = 0;
    Ixy_parb(m,:) = Hy_parb(m,:) - Hyx_parb(m,:);
end
% Rho = Hx/Ixy, now can be plotted in all the different cases.
```

A.3 Markov random fields

A.3.1 Generating a sample from a Markov random field

```
 % Here we generate samples of Markov random fields using Metropolis sampler
% algorithm.
clear all
clc
N = 128;
G = 8; % number of intensity levels
alpha = ones(1,G); % single-pixel clique potential
T = 3;% Temperature parameter in the Gibbs energy function
% k1, k2, k3, k4 are the 2-pixel clique potentials
k1 = -5;% horizontal clique
k2 = -5;% vertical cliqie
k3 = 0;% diagonal clique
k4 = 0;% the other diagonal clique
M = 250;% number of times the alorithm is run
X = G * rand(N,N); % initializing X
X = ceil(X);
for m = 1:M
    for i = 1:N
        for j = 1:N
            ip= rem(N+i,N)+1;
            im= rem(N+i-2,N)+1;
            jp= rem(N+j,N)+1;
            jm= rem(N+j-2,N)+1;
            clique_pot_old = alpha(X(i,j)) + 2 * ...
```

```
            (k1*(((X(i,j)==X(i,jp))-0.5)+((X(i,j)==X(i,jm))-0.5))+...
            k2*(((X(i,j)==X(ip,j))-0.5)+((X(i,j)==X(im,j))-0.5))+...
            k3*(((X(i,j)==X(ip,jm))-0.5)+((X(i,j)==X(im,jp))-0.5))+...
            k4*(((X(i,j)==X(im,jm))-0.5)+((X(i,j)==X(ip,jp))-0.5)));

            a = ceil(G * rand(1));

            clique_pot_new = alpha(a) + 2 *...
            (k1*(((a==X(i,jp))-0.5)+((a==X(i,jm))-0.5))+...
            k2*(((a==X(ip,j))-0.5)+((a==X(im,j))-0.5))+...
            k3*(((a==X(ip,jm))-0.5)+((a==X(im,jp))-0.5))+...
            k4*(((a==X(im,jm))-0.5)+((a==X(ip,jp))-0.5)));

            delta_V = (clique_pot_new - clique_pot_old)/T;
            prob = exp(-delta_V);
            p = min(1,prob);
            if (rand(1) < p)
                X(i,j) = a;
            end
        end
    end
end
```

A.3.2 Function to estimate the states in a neighborhood of an MRF

```
% Function to estimate the probabilities used in calculating the entropies
% and mutual information of a neighborhood in a Markov random field
```

```
function [P,R_x,R_y,Q_x,Q_y,U] = MRF_state_histogram(X,Y,G)
s = size(X);
n = 9; % the number of states in a second order neighborhood with two-pair cliques
P = zeros(G,2,n,n); % G is the no.of grey levels in the input image. Since output image is
photon counted we have only 2 levels.
R_x = zeros(1,n);
R_y = zeros(1,n);
Q_x = zeros(G,n);
Q_y = zeros(2,n);
U = zeros(n,n);
for i = 1:s(1)
    for j = 1:s(2)
        ip= rem(s(1)+i,s(1))+1;
        im= rem(s(1)+i-2,s(1))+1;
        jp= rem(s(2)+j,s(2))+1;
        jm= rem(s(2)+j-2,s(2))+1;

        k(1) = X(i,j) == X(i,jp);
        k(2) = X(i,j) == X(i,jm);
        k(3) = X(i,j) == X(ip,j);
        k(4) = X(i,j) == X(im,j);
        k(5) = X(i,j) == X(ip,jp);
        k(6) = X(i,j) == X(ip,jm);
        k(7) = X(i,j) == X(im,jp);
        k(8) = X(i,j) == X(im,jm);

        l(1) = Y(i,j) == Y(i,jp);
        l(2) = Y(i,j) == Y(i,jm);
```

```
        l(3) = Y(i,j) == Y(ip,j);
        l(4) = Y(i,j) == Y(im,j);
        l(5) = Y(i,j) == Y(ip,jp);
        l(6) = Y(i,j) == Y(ip,jm);
        l(7) = Y(i,j) == Y(im,jp);
        l(8) = Y(i,j) == Y(im,jm);

        beta_x = sum(k);
        beta_y = sum(l);
        %since Y(i,j) takes values in {0,1}, we add 1 to it.
        P(X(i,j),Y(i,j)+1,beta_x+1,beta_y+1)=P(X(i,j),Y(i,j)+1,beta_x+1,beta_y+1)+1;
        R_x(beta_x+1)=R_x(beta_x+1)+1;%estimate of the states in neighborhood of X
        Q_x(X(i,j),beta_x+1)=Q_x(X(i,j),beta_x+1)+1;%estimate of pixel and states in X
        R_y(beta_y+1)=R_y(beta_y+1)+1;%estimate of the states in neighborhood of Y
        Q_y(Y(i,j)+1,beta_y+1)=Q_y(Y(i,j)+1,beta_y+1)+1;%estimate of pixel and states
in Y
        U(beta_x+1,beta_y+1)=U(beta_x+1,beta_y+1)+1;
    end
end
R_x = R_x/sum(R_x);
Q_y = Q_y/sum(sum(Q_y));
R_y = R_y/sum(R_y);
Q_x = Q_x/sum(sum(Q_x));
U = U/sum(sum(U));
P = P/sum(sum(sum(sum(P))));
```

A.3.3 Calculation of ρ for images modeled as Markov random fields

```
% For images modeled as MRFs, we will now generate rho, i.e., the
% normalized mutual information metric. The images are loaded from a
% previously generated markov models and then we will estimate the
% probabilities here to generate the metric.
clear all
clc
% load the data file containing the MRF
load F:\alfa\Hayat\PAPER\real1000_1.mat
t = 0:1:1000; % different values of Np, the average number of photons in the input scene
st = size(t);
G = 8; % number of grey levels. In general, we should take max(max(Y)).
x = 1:G; % represents the number of bins in the input image.
y = 0:1; % represents the number of bins in the output image.
S = size(X_mrf); % X_mrf is the data file that we have loaded.
N = S(2); % N*N image. Size of the square images in the .mat file.
Hx = zeros(1,S(1));
Hy = zeros(1,S(1));
Ixy = zeros(1,S(1));
Rho_all = zeros(1,S(1));
Rho = zeros(st);
count = 0;
for Np = t
    count = count+1;
    for s=1:S(1)%run through the number of samples
        X_=X_mrf(s,:,:);%this will be a 1x128x128 matrix (related to the image file)
        X(:,:)=X_;%this will be a 128x128 image file.
```

```
Y=ones(N,N);%initializing the output photon count.
for i=1:(N*N)%runs through the entire input image file, X
    P=exp(-Np*((X(i)-1)/(G+1)));%count zero channel probability
    p=rand;
    Y(i)=  (p<P);%selects zero with probability P.
end
% At this point we have both X and the simulated photon counts. Now
% we have to estimate the probabilities that will be used in
% calculating the normalized mutual information metric. For this
% purpose we will use 'MRF_state_histogram' which will
% estimate the probability of the states in the neighborhood.
[T,R_x,R_y,Q_x,Q_y,U] = MRF_state_histogram(X,Y,G);
si = size(Q_x);
Rx = repmat(R_x,si(1),1);
si = size(Q_y);
Ry = repmat(R_y,si(1),1);
cond_x = Q_x./Rx;
cond_y = Q_y./Ry;
% The states which are not present in X should contribute nothin to
% the entropy
lx = find(Rx == 0);
cond_x(lx) = 1;
kx = find(cond_x==0);%Takes care of (pixel, state) case.
cond_x(kx) = 1;
Hx(s) = -sum(sum(Q_x.*log2(cond_x)));
% The states which are not present in Y should contribute nothin to
% the entropy
ly = find(Rx == 0);
```

```
        cond_y(ly) = 1;
        ky = find(cond_y==0);%Takes care of (pixel, state) case.
        cond_y(ky) = 1;
        Hy(s) = -sum(sum(Q_y.*log2(cond_y)));
        % let us create matrices of equal dimensions to calculate Ixy
        si = size(T);
        U_ = repmat(U,[1 1 si(1) si(2)]);
        Uf = permute(U_,[3 4 1 2]);
        Rx_ = repmat(cond_x,[1 1 si(2) si(4)]);
        Rxf = permute(Rx_, [1 3 2 4]);
        Ry_ = repmat(cond_y,[1 1 si(1) si(3)]);
        Ryf = permute(Ry_, [3 1 4 2]);
        % Now to calculate Ixy
        k = find(T == 0);
        T(k) = 1;
        Uf(k) = 1;
        Rxf(k) = 1;
        Ryf(k) = 1;
        Ixy(s) = sum(sum(sum(sum(T.*log2(T./(Uf.*Rxf.*Ryf))))));
        Rho_all(s) = Ixy(s)/Hx(s);
    end
    Rho(count) = sum(Rho_all)/S(1);
end
```

References

[1] B. Javidi, F. Okano, and J. Y. Son eds. Three dimensional imaging, visualization, and display. *Springer*, 2009.

[2] A. R. L. Travis. The display of three dimensional video images. *Proc. IEEE*, 85(11):1817–1832, November 1997.

[3] T. Okoshi. Three-dimensional displays. *Proc. IEEE*, 68:548–564, 1980.

[4] Y. Frauel, T. Naughton, O. Matoba, E. Tahajuerce, and B. Javidi. Three dimensional imaging and display using computational holographic imaging. *Proc. IEEE*, 94(3):636–654, 2006.

[5] S. Benton. Selected papers on 3d displays. *SPIE Press Book*, May 2001.

[6] G. Lippmann. Epreuves reversibles donnant la sensation du relief. *Journal of Physics (Paris)*, 7:821–825, 1908.

[7] H. E. Ives. Optical properties of a lippmann lenticulated sheet. *J. Opt. Soc. Am.*, 21:171–176, 1931.

[8] C. B. Burckhardt. Optimum parameters and resolution limitation of integral photography. *J. Opt. Soc. Am.*, 58:71–76, 1968.

[9] A. Stern and B. Javidi. 3d image sensing, visualization, and processing using integral imaging. *Proceedings of the IEEE Journal*, 94(3):591–608, March 2006.

[10] L. Yang, M. McCornick, and N. Davies. Discussion of the optics of a new 3-d imaging system. *Appl. Opt.*, 27:4529–4534, 1988.

[11] R. Martnez-Cuenca, H. Navarro, G. Saavedra, B. Javidi, and M. Martnez-Corral. Enhanced viewing-angle integral imaging by multiple-axis telecentric relay system. *Opt. Express*, 15:16255–16260, 2007.

[12] R. Martinez-Cuenca, G. Saavedra, M. Martinez-Corral, and B. Javidi. Progress in 3-d multiperspective display by integral imaging. *The Proceedings of the IEEE Journal*, 97(6):1067–1077, June 2009.

[13] Y. Igarishi, H. Murata, and M. Ueda. 3d display system using a computer-generated integral photograph. *Jpn. J. Appl. Phys.*, 17:1683–1684, 1978.

[14] B. Lee, S. Jung, and J.-H. Park. Viewing-angle-enhanced integral imaging using lens switching. *Opt. Lett.*, 27:818–820, May 2002.

[15] L. Erdmann and K. J. Gabriel. High resolution digital photography by use of a scanning microlens array. *Appl. Opt.*, 40:5592–5599, November 2001.

[16] F. Okano, J. Arai, H. Hoshino, and I. Yuyama. Three-dimensional video system based on integral photography. *Opt. Eng.*, 38:1072–1078, 1999.

[17] B. Javidi, S.-H. Hong, and O. Matoba. Multi dimensional optical sensors and imaging systems. *Appl. Opt.*, 45:2986–2994, 2006.

[18] H. Arimoto and B. Javidi. Integrate three-dimensional imaging with computed reconstruction. *Opt. Lett.*, 26:157–159, 2001.

[19] Marc Levoy and Pat Hanrahan. Light field rendering. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, New York, NY, USA, 1996. ACM.

[20] J. S. Jang and B. Javidi. Three-dimensional integral imaging of micro-objects. *Optics Letters*, 29(11):1230–1232, June 2004.

[21] G. M. Morris. Scene matching using photon-limited images. *J. Opt. Soc. Am. A.*, 1:482–488, 1984.

[22] J. W. Goodman. *Statistical optics*. John Wiley & sons, 1985.

[23] E. A. Watson and G. M. Morris. Comparison of infrared up conversion methods for photon-limited imaging. *J. Appl. Phys.*, 67:6075–6084, 1990.

[24] K. Lange and R. Carson. Em reconstruction algorithms for emission and transmission tomography. *J. Comput. Assist. Tomogr.*, 8(2):306–316, 1984.

[25] M. Guillaume, P. Melon, and P. Refregier. Maximum-likelihood estimation of an astronomical image from a sequence at low photon levels. *J. Opt. Soc. Am. A.*, 15:2841–2848, 1998.

[26] S. Yeoma, B. Javidi, and E. Watson. Three-dimensional distortion-tolerant object recognition using photon-counting integral imaging. *OPTICS EXPRESS*, 15(4):1513–1533, 2007.

[27] B. Tavakoli, B. Javidi, and E. Watson. Three-dimensional visualization by photon counting computational integral imaging. *OPTICS EXPRESS*, 16(7):4426–4436, 2008.

[28] I. Moon and B. Javidi. Three-dimensional recognition of photon starved events using computational integral imaging and statistical sampling. *Optics Letters*, 34(6):731–733, March 2009.

[29] B. E. A. Saleh and M. C. Teich. *Fundamentals of Photonics*. New York: Wiley, 2007.

[30] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & sons, 1991.

[31] E. Volden, G. Giraudon, and M. Berthod. Information in markov random fields and image redundancy. In *Selected Papers from the 4th Canadian Workshop on Information Theory and Applications II*, pages 250–268, London, UK, 1996. Springer-Verlag.

[32] L. Mandel. Sub-poissonian photon statistics in resonance fluorescence. *Opt. Lett.*, 4:205–207, 1979.

[33] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.