An Optimal Algorithm for Reconstructing Images from Binary Measurements

Feng Yang[1], Yue M. Lu[1], Luciano Sbaiz[2], and Martin Vetterli[1,3]

[1]School of Computer and Communication Sciences

Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland

[2]Google, CH-8002 Zürich, Switzerland

[3]Department of Electrical Engineering and Computer Sciences

University of California, Berkeley, CA94720, USA

ABSTRACT

We have studied a camera with a very large number of binary pixels referred to as the gigavision camera [1] or the gigapixel digital film camera [2, 3]. Potential advantages of this new camera design include improved dynamic range, thanks to its logarithmic sensor response curve, and reduced exposure time in low light conditions, due to its highly sensitive photon detection mechanism.

We use maximum likelihood estimator (MLE) to reconstruct a high quality conventional image from the binary sensor measurements of the gigavision camera. We prove that when the threshold T is "1", the negative log-likelihood function is a convex function. Therefore, optimal solution can be achieved using convex optimization. Base on filter bank techniques, fast algorithms are given for computing the gradient and the multiplication of a vector and Hessian matrix of the negative log-likelihood function. We show that with a minor change, our algorithm also works for estimating conventional images from multiple binary images.

Numerical experiments with synthetic 1-D signals and images verify the effectiveness and quality of the proposed algorithm. Experimental results also show that estimation performance can be improved by increasing the oversampling factor or the number of binary images.

Keywords: Computational photography, high dynamic range imaging, low light level imaging, the gigavision camera, digital film, photon-limited imaging.

1. INTRODUCTION

Conventional digital camera comprises a single lens, an image sensor, an image processing unit, and readout circuitry. The image sensor converts photons to analog electrical signals, which are then quantized by an A/D converter into 8 or 12 bits.

In this paper, we study a camera having image sensor with 1-bit quantizer first proposed as the gigapixel digital film in [2, 3], then studied theoretically in the name of the gigavision sensor in [1]. The main differece between the gigavision camera and the conventional camera is that the A/D converter is 1-bit quantizer, where a threshold T determines how many photoelectrons are needed to switch the value of the binary pixel from "0" to "1". Due to the diffraction limit, Rayleigh criterion gives the minimum spatial resolution of an ideal lens [4]. Thanks to the scaling effects in CMOS technology, the state of the art pixel size is already smaller than the minimum spatial resolution of most consumer lens [5]. In this case, the image sensor can be considered as an oversampling device. Through using binary pixel, we can reduce the size of pixels further, hence increase the oversampling factor. This binary sensor works as a spatial oversampling A/D converters, thus potentially high resolution A/D converters [6]. Gray levels are determined, like in a conventional film, by the density of tiny digital pixels, and we can obtain these values by low-pass filtering the binary image. Owing to the number of photons hitting the sensor well-modeled by the Poisson process [7], as shown in [1], the sensor's response is non-linear and similar to a logarithmic function, which is suitable for acquiring high dynamic range scenes.
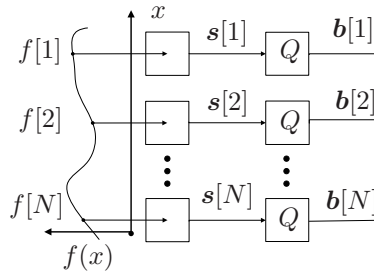
Figure 1. Simplified block diagram of a gigavision sensor. The light intensity function is $f(x)$. There are N pixels in the sensor. The light intensity on each pixel is $f[n], n = 1, \ldots, N$. The electrical signal $\boldsymbol{s}[n]$ is quantized by a one-bit quantizer Q with threshold T.

The most important problem in the gigavision camera is to reconstruct a conventional image using the binary image. Since the performance of reconstructing images by low-pass filtering the binary images is sub-optimal [1], we need to design new reconstruction algorithms according to the physical model of the gigavision camera. Section 2 gives the imaging model and problem statement. Section 3 describes the maximum likelihood estimator (MLE) used for reconstruction. We show that when the threshold of the gigavision camera is equal to "1", the negative log-likelihood function is a convex function. Thus, the optimal solution can be found using convex optimization. To accelerate the optimization speed, we propose fast algorithms to compute the negative log-likelihood function's gradient and the multiplication of its Hessian matrix and a vector based on filter bank techniques. We prove that our algorithm can be used for reconstructing images from multiple binary images with a minor change. Section 4 is the experimental results on synthesized 1-D signals and images which show the effectiveness of our algorithm. We also demonstrate that the estimation performance can be improved by increasing the oversampling factor or the number of binary images using numerical experiments. We conclude this paper in Section 5.

2. IMAGING MODEL AND PROBLEM STATEMENT

2.1 Imaging model

2.1.1 Pixel model

For simplicity of notation, we consider a one-dimensional (1-D) sensor array in this paper. All the results and discussions can be straightforwardly extended to the two-dimensional (2-D) case. Figure 1 shows the model of a gigavision sensor. The gigavision pixel is similar to a conventional camera pixel except that the quantizers Q are binary with threshold T. The number of photons impinging on the pixel can be modeled as a Poisson process with intensity $f(x)$, where x is the position parameter. Suppose there are N pixels and neglect the integration for simplicity, then the light intensity for every pixel is $f[n], n = 1, \ldots, N$, respectively. The response of each binary pixel $\boldsymbol{b}[n], n = 1, \ldots, N$ * is obtained by comparing the number of arrivals $\boldsymbol{s}[n]$, i.e. the electrons due to detected photons, with the threshold T. The quantities $\boldsymbol{b}[n]$ are binary random variables with parameter

$$p_{f[n]} \triangleq \mathbb{P}(\boldsymbol{s}[n] \geq T) = \sum_{k=T}^{\infty} e^{-f[n]} \frac{(f[n])^k}{k!}, n = 1, \ldots, N.$$

2.1.2 Camera model

In the gigavision camera, the incident light I is first low-pass filtered by the lens and then oversampled using the gigavision sensor, as shown in Figure 2. An estimation method is implemented to reconstruct the light intensity using the binary image captured by the gigavision sensor. The model of the gigavision camera is shown in Figure 3. Let $f = [f[1], f[2], \ldots, f[N]]^T$ be the sampled light intensity values and $\boldsymbol{b} = [\boldsymbol{b}[1], \boldsymbol{b}[2], \ldots, \boldsymbol{b}[N]]^T$ be the binary image captured by the gigavision sensor. The estimated light intensity values for each pixel $\widehat{\boldsymbol{f}} = [\widehat{\boldsymbol{f}}[1], \widehat{\boldsymbol{f}}[2], \ldots, \widehat{\boldsymbol{f}}[N]]^T$ is obtained using some reconstruction algorithm. Since the lens is a low-pass filter,

---

*we use bold letter to denote a random variable or vector and the corresponding normal-font letter is a realization.
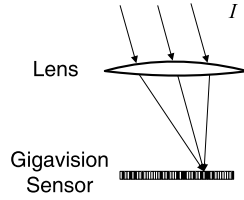
Figure 2. Simplified architecture of a gigavision camera. The incident light is focused by the lens and then impinges on the image sensors.
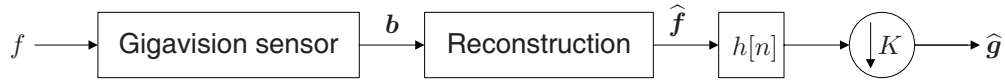

Figure 3. The model for the gigavision camera. $f = [f[1], f[2], \ldots, f[N]]^T$ is the sampled light intensity values. $b = [\boldsymbol{b}[1], \boldsymbol{b}[2], \ldots, \boldsymbol{b}[N]]^T$ is the binary image. $\widehat{\boldsymbol{f}} = [\widehat{\boldsymbol{f}}[1], \widehat{\boldsymbol{f}}[2], \ldots, \widehat{\boldsymbol{f}}[N]]^T$ is the reconstructed light intensity. $h[n]$ is a low-pass filter. $K = \frac{N}{M}$ is downsampling factor and is equal to the oversampling factor of the gigavision sensor. $\widehat{\boldsymbol{g}} = [\widehat{\boldsymbol{g}}[1], \widehat{\boldsymbol{g}}[2], \ldots, \widehat{\boldsymbol{g}}[M]]^T$ is the reconstructed non-blurred image.

$\widehat{\boldsymbol{f}}$ is a blurred image. Low-pass filtering and downsampling by a factor of K which is equal to the oversampling factor of the gigavision sensor are needed to generate a non-blurred image $\widehat{\boldsymbol{g}} = [\widehat{\boldsymbol{g}}[1], \widehat{\boldsymbol{g}}[2], \ldots, \widehat{\boldsymbol{g}}[M]]^T$. The downsampling factor $K = \frac{N}{M}$.

2.2 Problem statement

Since the final goal is to estimate the non-blurred image $g = [g[1], g[2], \ldots, g[M]]^T$, which is obtained by low-pass filtering and downsampling f, we want to design a reconstruction algorithm which can directly compute $\widehat{g}$ without $\widehat{\boldsymbol{f}}$. To achieve this, the model for the gigavision camera is described in an equivalent way as shown in Figure 4. g is critically sampled light intensity values. We upsample g with a factor of $K = \frac{N}{M}$ and filter it using a low-pass filter to get the oversampled light intensity values f. The binary image $\boldsymbol{b}$ are generated by the gigavision sensor. Finally a reconstruction algorithm is used to obtain $\widehat{\boldsymbol{g}}$.

3. IMAGE RECONSTRUCTION USING MAXIMUM LIKELIHOOD ESTIMATION

3.1 Maximum likelihood estimator

A maximum likelihood estimator(MLE) is proposed for solving the reconstruction problem. The relation between f and g can be written as $f = Hg$, where H is $N \times M$ matrix representing the upsampling and low-pass filtering operator, as shown in Figure 5. Suppose $H = [h_1, h_2, \ldots, h_N]^T$, then $f[n] = h_n^T g, n = 1, 2, \ldots, N$. We assume all the entries in H is larger than 0 to avoid negative light intensity values. $f[n]$ is a linear function of g and we can also write $f[n]$ as $f_n(g)$. According to the pixel model of the gigavision sensor and only considering the case when the threshold $T = 1$, we have

$$\begin{cases} \mathbb{P}(\boldsymbol{b}_n = 0) = e^{-f_n(g)} = e^{-h_n^T g} \\ \mathbb{P}(\boldsymbol{b}_n = 1) = 1 - e^{-f_n(g)} = 1 - e^{-h_n^T g} \end{cases}, n = 1, 2, \ldots N.$$

The MLE is as follows:

$$\begin{aligned} \widehat{\boldsymbol{g}} &= \arg\max_g \mathbb{P}(\boldsymbol{b}; g) \\ &= \arg\max_g \prod_{n=1}^N \left((1 - \boldsymbol{b}[n])e^{-f_n(g)} + \boldsymbol{b}[n](1 - e^{-f_n(g)}) \right) \\ &= \arg\max_g \ln\left(\prod_{n=1}^N \left((1 - \boldsymbol{b}[n])e^{-f_n(g)} + \boldsymbol{b}[n](1 - e^{-f_n(g)}) \right) \right) \\ &= \arg\min_g -\sum_{n=1}^N \ln\left((1 - \boldsymbol{b}[n])e^{-f_n(g)} + \boldsymbol{b}[n](1 - e^{-f_n(g)}) \right). \end{aligned}$$

Figure 4. The equivalent model for the gigavision camera. $g = [g[1], g[2], \ldots, g[M]]^T$ is critically sampled light intensity values. $K = \frac{N}{M}$ is the upsampling factor. $h[n]$ is a low-pass filter. $f = [f[1], f[2], \ldots, f[N]]^T$ is the oversampled light intensity values. $\boldsymbol{b} = [\boldsymbol{b}[1], \boldsymbol{b}[2], \ldots, \boldsymbol{b}[N]]^T$ is the binary image. $\widehat{g} = [\widehat{g}[1], \widehat{g}[2], \ldots, \widehat{g}[M]]^T$ is the reconstructed light intensity values.
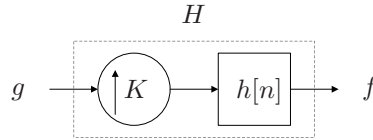

Figure 5. H is the matrix notation for upsampling and low-pass filtering operator. K is the upsampling factor. $h[n]$ is a low-pass filter.

The negative log-likelihood function is $O(g) = -\sum\limits_{n=1}^{N} \ln\left((1 - \boldsymbol{b}[n])e^{-f_n(g)} + \boldsymbol{b}[n](1 - e^{-f_n(g)})\right), g \in \mathbf{R}_+^M$

LEMMA 3.1. *The gradient of $O(g)$ is* $\nabla O(g) = H^T[(1 - \boldsymbol{b}[1]) + \boldsymbol{b}[1]\frac{-e^{-f_1(g)}}{1 - e^{-f_1(g)}}, \ldots, (1 - \boldsymbol{b}[N]) + \boldsymbol{b}[N]\frac{-e^{-f_N(g)}}{1 - e^{-f_N(g)}}]^T$.

Proof. $\frac{\partial O(g)}{\partial f_n(g)} = \begin{cases} 1, \boldsymbol{b}[n] = 0 \\ \frac{-e^{-f_n(g)}}{1 - e^{-f_n(g)}}, \boldsymbol{b}[n] = 1 \end{cases}, n = 1, 2, \ldots, N.$

According to the chain rule,

$$\nabla O(g) = \frac{\partial O(g)}{\partial g} = \frac{\partial f}{\partial g}\frac{\partial O(g)}{\partial f} = H^T[(1 - \boldsymbol{b}[1]) + \boldsymbol{b}[1]\frac{-e^{-f_1(g)}}{1 - e^{-f_1(g)}}, \ldots, (1 - \boldsymbol{b}[N]) + \boldsymbol{b}[N]\frac{-e^{-f_N(g)}}{1 - e^{-f_N(g)}}]^T.$$

$\square$

LEMMA 3.2. *The Hessian matrix of $O(g)$ is* $\boldsymbol{Q} = H^T \boldsymbol{A} H$, *where,*

$$\boldsymbol{A} = \begin{pmatrix} \frac{\boldsymbol{b}[1]e^{-f_1(g)}}{(1-e^{-f_1(g)})^2} & 0 & \cdots & 0 \\ 0 & \frac{\boldsymbol{b}[2]e^{-f_2(g)}}{(1-e^{-f_2(g)})^2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\boldsymbol{b}[N]e^{-f_N(g)}}{(1-e^{-f_N(g)})^2} \end{pmatrix}.$$

Proof. According to the chain rule,

$$\boldsymbol{Q} = \frac{\partial}{\partial g}\left(\frac{\partial O(g)}{\partial g}\right) = \frac{\partial f}{\partial g}\frac{\partial}{\partial f}\left(\frac{\partial O(g)}{\partial g}\right) = H^T \boldsymbol{A} H.$$

$\square$

Then the following proposition holds true.

PROPOSITION 1. $O(g)$ *is a convex function.*

Proof. Since the Hessian matrix of $O(g)$, $\boldsymbol{Q} \succeq 0$, i.e. $\boldsymbol{Q}$ is positive semidefinite, $O(g)$ is a convex function [8].

$\square$

Figure 6 shows the negative log-likelihood function in a 2-D case, when $g = [g[1], g[2]]^T = [2, 3]^T$, oversampling factor is 200 and H is a randomly generated 400×2 matrix, which also demonstrates that the negative log-likelihood function is a convex function. Thus, any method for solving convex optimization problem such as the interior-point method can be used to solve the above optimization problem.
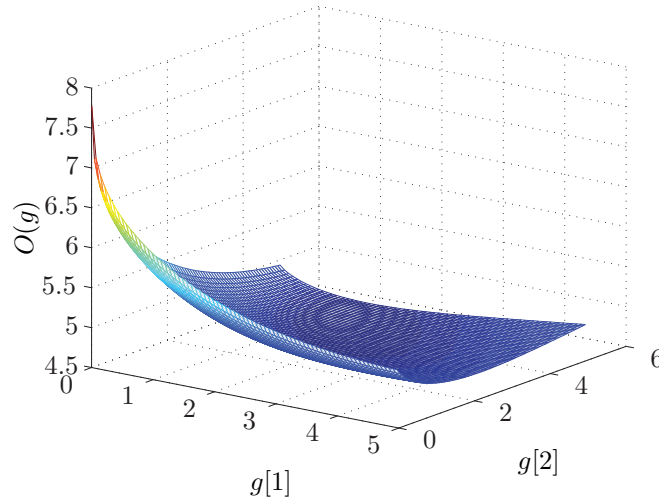
Figure 6. The objective function, when $g = [g[1], g[2]]^T = [2, 3]^T$, and the oversampling factor is 200.
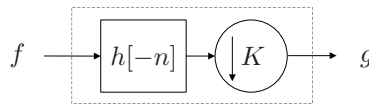

Figure 7. H^T is the matrix notation for low-pass filtering and downsampling operator. $h[-n]$ is a low-pass filter. K is the downsampling factor.

3.2 Implementation

3.2.1 Calculation of gradient and Hessian matrix times a vector

In most methods for solving convex optimization problem, like interior-point algorithm or trust-region algorithm, we need to provide the negative log-likelihood function's gradient, Hessian matrix multiplication with a vector. These can be computed according to lemma 3.1 and 3.2. The problem for directly calculating using the above two equations is that if the upsampling factor is large, the matrix H is big and huge storage space is required for H which makes it impractical. Since H is the matrix notation for upsampling and low-pass filtering operator, as in Figure 5, there is no requirement for storing the matrix H. We only need to know the parameters of the upsampling and low-pass filtering operator, i.e the upsampling factor K and the coefficients of the low-pass filter $h[n]$. Let $H = LU$, where L indicates low-pass filtering operator and U denotes upsampling operator. Then $H^T = (LU)^T = U^T L^T = DG$, where D is the matrix notation for downsampling operator and G is the matrix notation for low-pass filtering operator. The downsampling factor of D is equal to the upsampling factor of U. If the filter coefficients of L is $h[n]$, then the filter coefficient of G is $h[-n]$. In the case of symmetric low-pass filter, the filter coefficients of L and G are the same. Thus H^T is the matrix notation for low-pass filtering and downsampling operator, as in Figure 7.

According to lemma 3.1, the gradient of the negative log-likelihood function can be computed by first low-pass filtering the vector $[(1 - \boldsymbol{b}[1]) + \boldsymbol{b}[1]\frac{-e^{-f_1(g)}}{1-e^{-f_1(g)}}, \ldots, (1 - \boldsymbol{b}[N]) + \boldsymbol{b}[N]\frac{-e^{-f_N(g)}}{1-e^{-f_N(g)}}]^T$, then downsampling it by a factor of $K = \frac{N}{M}$. According to lemma 3.2, the negative log-likelihood function's Hessian matrix $\boldsymbol{Q}$ times a vector v is $\boldsymbol{Q}v = H^T\boldsymbol{A}Hv$. Figure 8 shows the diagram for computing the above equation. We upsample v with upsampling factor $K = \frac{N}{M}$, then low-pass filter it using a filter $h[n]$. Since the matrix $\boldsymbol{A}$ is a diagonal matrix, the multiplication of $\boldsymbol{A}$ and vector Hv is equal to the elementwise multiplication of the diagonal of $\boldsymbol{A}$ and Hv. After that, we low-pass filter the obtained vector with a filter $h[-n]$ and downsample by a factor of $K = \frac{N}{M}$ to get $\boldsymbol{Q}v$.
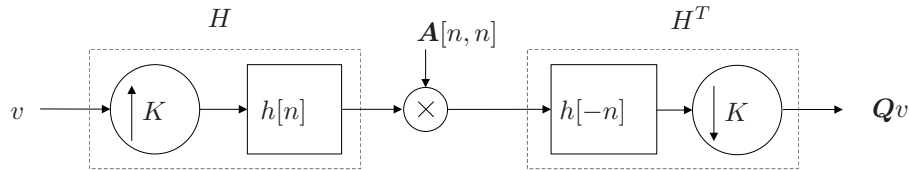
Figure 8. The diagram from computing the negative log-likelihood function's Hessian matrix $\boldsymbol{Q}\boldsymbol{v}$ times a vector $\boldsymbol{v}$. $K = \frac{N}{M}$ is upsampling and downsampling factor. $h[n]$ and $h[-n]$ are low-pass filters.
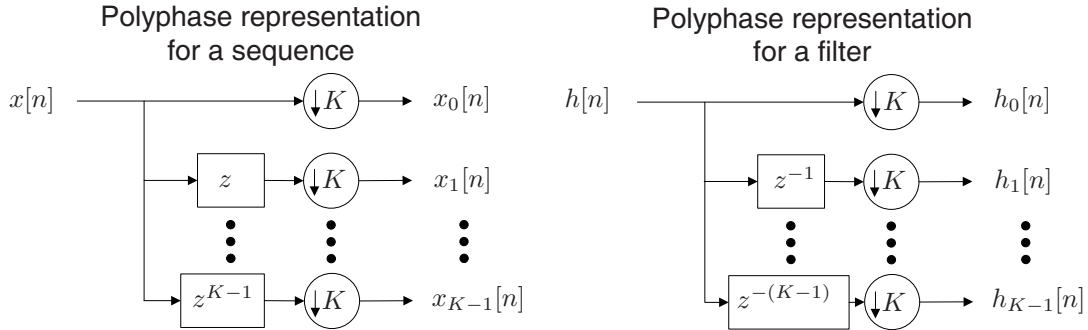

Figure 9. The polyphase representation of a sequence $x[n]$ (left) and a filter $h[n]$ (right).

3.2.2 Polyphase representation

To further increase the speed of the optimization process, we propose to use polyphase representation [9] to reduce the computing time of the upsampling and low-pass filtering operator, also the low-pass filtering and downsampling operator. Polyphase representation is a very useful tool in multirate signal processing. We will introduce the definition and properties of the polyphase representation here to those readers who are not familiar with this topic. Different polyphase representations are defined for a sequence and a filter.

DEFINITION 3.3. *A 1-D sequence $x[n]$ or filter $h[n]$ can be decomposed into K polyphase components, defined as,*

$$x_i[n] \triangleq x[Kn+i], h_i[n] \triangleq h[Kn-i], i = 0, 1, \ldots, K-1.$$

Figure 9 shows the polyphase representation of a sequence $x[n]$ and a filter $h[n]$. In the z-domain, we have $X(z) = \sum_{k=0}^{K-1} z^{-k} X_k(z^K)$, and $H(z) = \sum_{k=0}^{K-1} z^k H_k(z^K)$.

PROPOSITION 2. *Let $Y(z)$ be the z transform of the sequence $y[n]$, which is the output when we implement the low-pass filtering and downsampling operator on a signal $x[n]$, then we have $Y(z) = \sum_{k=0}^{K-1} H_k(z) X_k(z)$.*

Proof. Omitted. □

REMARK 1. *This means that the process for the implementation of the operator on a sequence $x[n]$ is decomposing the filter and sequence into K polyphase components, filtering the kth polyphase component of the sequence separately with the corresponding kth polyphase component of the filter, and summing all the filtered results to generate $y[n]$. During this process, we avoid to compute the sequence value which will be discarded during the downsampling process, therefore the computing time can be saved.*

PROPOSITION 3. *If we decompose $h[n]$ using the definition of polyphase representation for a sequence, then $H(z) = \sum_{k=0}^{K-1} z^{-k} H_k(z^K)$ and the output of the implementing the upsampling and low-pass filtering operator on $x[n]$ can be written as*

$$\begin{pmatrix} Y_0(z) \\ Y_1(z) \\ \vdots \\ Y_K(z) \end{pmatrix} = \begin{pmatrix} H_0(z) \\ H_1(z) \\ \vdots \\ H_K(z) \end{pmatrix} X(z),$$

where $Y_k(z)$ is the z transform of the kth polyphase component of the output sequence $y[n]$.

Proof. Omitted. □

REMARK 2. *This means that kth polyphase component of $y[n]$ can be calculated by filtering $x[n]$ using the kth polyphase component of $h[n]$, and $y[n]$ can be obtained according to $Y(z) = \sum_{k=0}^{K-1} z^{-k} Y_k(z^K)$. Since we avoid to compute the multiplication of the filter coefficient and the "0"s generated during the upsampling process, we can increase the speed of the algorithm by a factor of K.*

3.3 Multiple exposures

In the above, we only discuss about the reconstruction problem when a single binary image is captured by the gigavision camera under one exposure. In some cases the resolution of the gigavision camera is not large enough or the high speed is not required, we can capture many binary images using multiple exposures and estimate light intensity using all the binary images to achieve better reconstruction performance. In what follows, we show that the proposed reconstrucion algorithm can be straightforwardly extended to the multi-frame case, with no increase in the computational complexity. When taking S binary images, $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_S$, the maximum likelihood estimator is

$$
\begin{aligned}
\widehat{\boldsymbol{g}} &= \arg\max_g \mathbb{P}\left(\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_S; g\right) = \arg\max_g \prod_{s=1}^{S} \prod_{n=1}^{N} \left((1 - \boldsymbol{b}_s[n])e^{-f_n(g)} + \boldsymbol{b}_s[n](1 - e^{-f_n(g)})\right) \\
&= \arg\max_g \ln \prod_{s=1}^{S} \prod_{n=1}^{N} \left((1 - \boldsymbol{b}_s[n])e^{-f_n(g)} + \boldsymbol{b}_s[n](1 - e^{-f_n(g)})\right) \\
&= \arg\min_g -\sum_{s=1}^{S} \sum_{n=1}^{N} \ln \left((1 - \boldsymbol{b}_s[n])e^{-f_n(g)} + \boldsymbol{b}_s[n](1 - e^{-f_n(g)})\right).
\end{aligned}
$$

The negative log-likelihood function is $O(g) = -\sum_{s=1}^{S} \sum_{n=1}^{N} \ln \left((1 - \boldsymbol{b}_s[n])e^{-f_n(g)} + \boldsymbol{b}_s[n](1 - e^{-f_n(g)})\right)$.

LEMMA 3.4. *The gradient of $O(g)$ is*

$$
\begin{aligned}
\triangledown O(g) &= H^T\Big[\sum_{s=1}^{S}(1 - \boldsymbol{b}_s[1]) + \boldsymbol{b}_s[1]\frac{-e^{-f_1(g)}}{1 - e^{-f_1(g)}}, \sum_{s=1}^{S}(1 - \boldsymbol{b}_s[2]) + \boldsymbol{b}_s[2]\frac{-e^{-f_2(g)}}{1 - e^{-f_2(g)}}, \\
&\quad \ldots, \sum_{s=1}^{S}(1 - \boldsymbol{b}_s[N]) + \boldsymbol{b}_s[N]\frac{-e^{-f_N(g)}}{1 - e^{-f_N(g)}}\Big]^T.
\end{aligned}
$$

Proof. $\frac{\partial O(g)}{\partial f_n(g)} = \sum_{s=1}^{S} \left((1 - \boldsymbol{b}_s[n]) + \boldsymbol{b}_s[n]\frac{-e^{-f_n(g)}}{1-e^{-f_n(g)}}\right)$

According to the chain rule,

$$
\begin{aligned}
\triangledown O(g) &= \frac{\partial O(g)}{\partial g} = \frac{\partial f}{\partial g}\frac{\partial O(g)}{\partial f} = H^T\frac{\partial O(g)}{\partial f} \\
&= H^T\Big[\sum_{s=1}^{S}(1 - \boldsymbol{b}_s[1]) + \boldsymbol{b}_s[1]\frac{-e^{-f_1(g)}}{1 - e^{-f_1(g)}}, \sum_{s=1}^{S}(1 - \boldsymbol{b}_s[2]) + \boldsymbol{b}_s[2]\frac{-e^{-f_2(g)}}{1 - e^{-f_2(g)}}, \\
&\quad \ldots, \sum_{s=1}^{S}(1 - \boldsymbol{b}_s[N]) + \boldsymbol{b}_s[N]\frac{-e^{-f_N(g)}}{1 - e^{-f_N(g)}}\Big]^T.
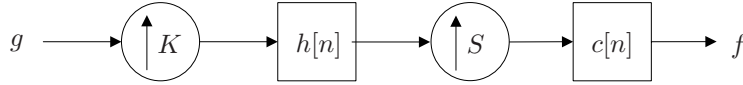\end{aligned}
$$

□

Figure 10. The model for reconstructing images from multiple binary images. $g = [g[1], g[2], \ldots, g[M]]^T$ is critically sampled light intensity values. $K = \frac{N}{M}$ is the upsampling factor. $h[n]$ is a low-pass filter. $f = [f[1], f[2], \ldots, f[NS]]^T$ is the oversampled light intensity values. $c[n] = 1, n = 1, 2, \ldots, S$.

LEMMA 3.5. *The Hessian matrix of $O(g)$ is $\boldsymbol{Q} = H^T \boldsymbol{R} H$, where,*

$$
\boldsymbol{R} = \begin{pmatrix}
\frac{\sum\limits_{s=1}^{S} \boldsymbol{b}_s[1] e^{-f_1(g)}}{(1 - e^{-f_1(g)})^2} & 0 & \cdots & 0 \\
0 & \frac{\sum\limits_{s=1}^{S} \boldsymbol{b}_s[2] e^{-f_2(g)}}{(1 - e^{-f_2(g)})^2} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \frac{\sum\limits_{s=1}^{S} \boldsymbol{b}_s[N] e^{-f_N(g)}}{(1 - e^{-f_N(g)})^2}
\end{pmatrix}.
$$

Proof. According to the chain rule,

$$
\boldsymbol{Q} = \frac{\partial}{\partial g} \left(\frac{\partial O(g)}{\partial g} \right) = \frac{\partial f}{\partial g} \frac{\partial}{\partial f} \left(\frac{\partial O(g)}{\partial g} \right) = H^T \boldsymbol{R} H.
$$

□

The calculation of gradient and Hessian matrix is almost the same as that for the single exposure.

There is also an insight linking temporal oversampling with a special form of spatial oversampling.

PROPOSITION 4. *The estimation result $\widehat{\boldsymbol{g}}(\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_S)$ with spatial oversampling factor $K = \frac{N}{M}$ and low-pass filter $h[n]$ is equivalent to $\widehat{\boldsymbol{g}}(\boldsymbol{b})$ with spatial oversampling factor $W = S \frac{N}{M}$ and low-pass filter $l[n]$, where $l[n]$ equals to the convolution between a filter $c[n] = 1, n = 1, 2, \ldots, S$ with the upsampled version of $h[n]$ by a factor of S.*

Proof. The model for reconstructing images from multiple binary images is shown in Figure 10. If we exchange the positions of the second upsampling operator and low-pass filter $h[n]$, the resulting model is equivalent to the model used to generate $\widehat{\boldsymbol{g}}(\boldsymbol{b})$ with spatial oversampling factor $W = S \frac{N}{M}$ and low-pass filter $l[n]$. So the above proposition is correct. □

4. EXPERIMENTAL RESULTS

4.1 1-D signals

We first did some experiments on 1-D synthesized signals. The gigavision model used is the model shown in Figure 4. We first generate a 1-D synthesized signal. Then it is upsampled by 100 and low-pass filtered by a Gaussian low-pass filter with variance $\sigma = 30$ and filter length 301. After that, a binary sequence is generated according to the pixel model of the gigavision sensor. The estimated signal is obtained using the binary sequence and the MLE method described above. The optimization method used is the interior-point algorithm. The estimation result is shown in Figure 11. The mean square error (MSE) is 0.3477. Figure 12 shows the estimation result when the oversampling factor is 200, the Gaussian low-pass filter's σ is 50 and filter length is 401. The MSE is 0.2060. Figure 13 shows the result using the same parameters as in Figure 11, except that we take 20 binary images. The MSE is 0.0158.

In Figure 14, we shows the relation between oversampling factor and MSE for the above synthesized sequence. We only use 1 binary image and run the experiment 50 times for each oversampling factor. The MSE is an average MSE of the 50 experiments. Figure 15 shows the relation between number of binary images and MSE for the
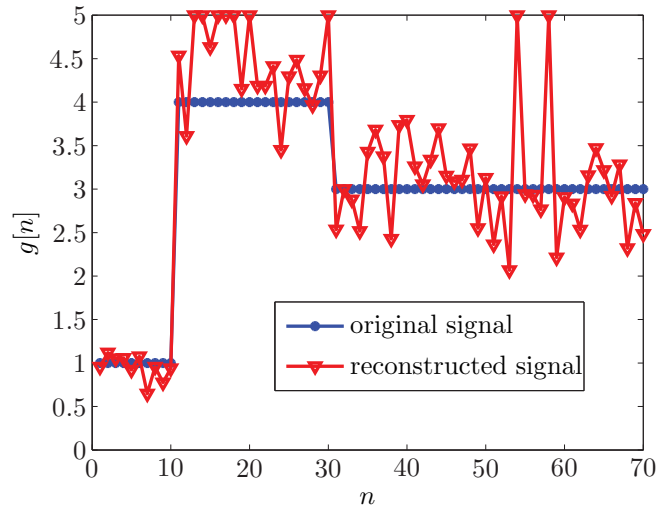
Figure 11. The estimation result when oversampling factor is 100 and we only take 1 binary image. The MSE is 0.3477.
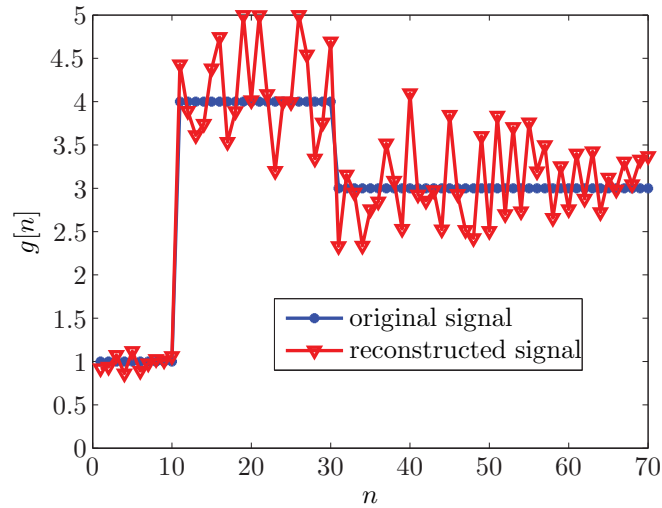

Figure 12. The estimation results when the oversampling factor is 200 and we only take 1 binary image. The MSE is 0.2060.

synthesized sequence. The oversampling factor is equal to 100 for each number of binary images. We run the experiments 50 times. The MSE is an average MSE of the 50 experiments. We can conclude that with higher oversampling factor or more binary images, the estimation performance can be improved.

4.2 2-D images

A color image with resolution 2016×3024 is used for simulation. The color image contains three channels, R, G, and B. We reconstruct each channel seperately. The image is first filtered by a 1-D Gaussian low-pass filter with $\sigma = 4$ and filter length 61 horizontally and vertically. After that, it is downsampled using downsampling factor 8 by 8. The resulting image is shown in Figure 16(a). A binary image is generated according to the pixel model of the gigavision sensor using the original image. Then we use the proposed reconstruction method to estimate the reconstructed image using the binary image. The results when using 1 binary image and 255 binary images are shown in Figure 16(b) and (c) respectively. We can see that when using 1 binary image, the reconstructed image is a bit noisy, but when using 255 binary images, the quality of the reconstructed image becomes better.
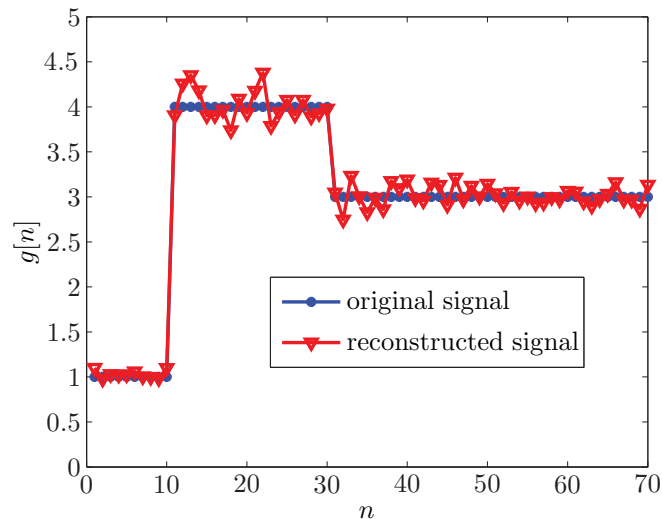
Figure 13. The estimation results when the oversampling factor is 100 and we take 20 binary images. The MSE is 0.0158.
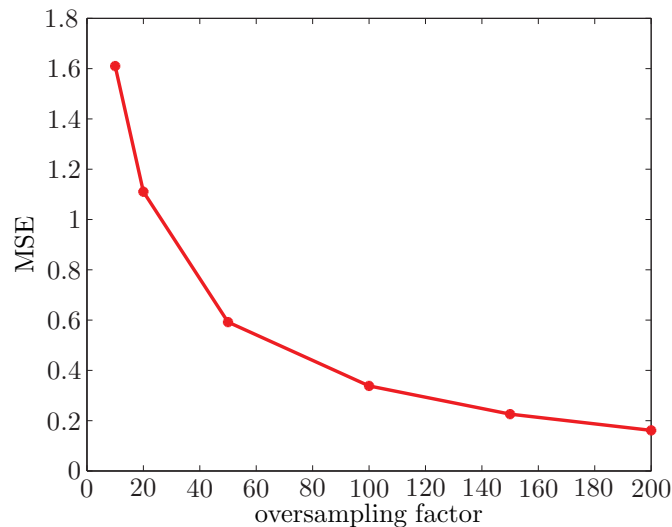

Figure 14. The relation between the oversampling factor and MSE for the synthetic sequence. We only use 1 binary image and run the experiment 50 times for each oversampling factor.

5. CONCLUSIONS

In this paper, we propose a maximum likelihood estimator algorithm for reconstructing images from binary measurements. When the threshold of the gigavision camera is equal to "1", we show that the problem is a convex optimization problem, and hence we can reach the global optimal solution. To increase the speed of the optimization, fast algorithms based on filter bank techniques are proposed. We prove that our algorithm also works for reconstructing images from multiple binary images with a small change. Experimental results verify the effectiveness of our algorithm. We show that the estimation performance can be improved by increasing the oversampling factor or the number of independent exposures.

REFERENCES

[1] L. Sbaiz, F. Yang, E. Charbon, S. Süsstrunk, and M. Vetterli, "The gigavision camera," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1093–1096, (Taipei), April 2009.
[2] E. R. Fossum, "What to do with sub-diffraction-limit (SDL) pixels? – a proposal for a gigapixel digital film
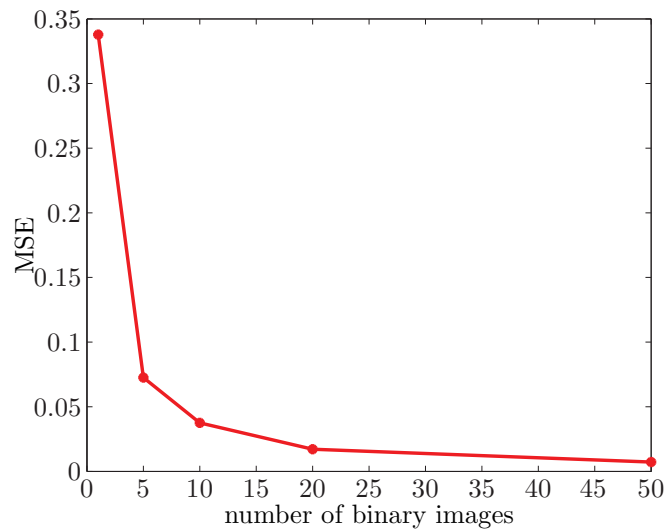
Figure 15. The relation between the number of binary images and MSE for the synthetic sequence. The oversampling factor is equal to 100 and for each number of binary images, we run the experiments 50 times.

(a) Original image

(b) Oversampling factor is 8 by 8,
1 binary image.

(c) Oversampling factor is 8 by 8,
255 binary images.

Figure 16. Reconstructed color images.

sensor (DFS)," in *IEEE Workshop on Charge-Coupled Devices and Advanced Image Sensors*, pp. 214–217, (Nagano), June 2005.

[3] E. R. Fossum, "Gigapixel digital film sensor," in *Nanospace Manipulation of Photons and Electrons for Nanovision Systems, the* 7*th Takayanagi Kenjiro Memorial Symposium and the* 2*nd International Symposium on Nanovision Science*, (Hamamatsu), Oct. 2005.

[4] M. Born and E. Wolf, *Principles of Optics*, Cambridge University Press, Cambridge, 7th ed., 1999.

[5] A. J. Theuwissen, "CMOS image sensors: State-of-the-art," *Solid-state Electronics* **52**, pp. 1401–1406, Sept. 2008.

[6] J. C. Candy and G. C. Temes, *Oversampling Delta-Sigma Data Converters — Theory, Design and Simulation*, IEEE Press, New York, 1992.

[7] J. W. Goodman, *Statistical Optics*, Wiley-Interscience, New York, 1985.

[8] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, New York, 2004.

[9] M. Vetterli, J. Kovačević, and V. K. Goyal, *The World of Fourier and Wavelets: Theory, Algorithm and Applications*, 2010.