

# Generating Fluttering Patterns with Low Autocorrelation for Coded Exposure Imaging

Hae-Gon Jeon<sup>1</sup> · Joon-Young Lee<sup>1</sup> · Yudeog Han<sup>1</sup> · Seon Joo Kim<sup>2</sup> ·  
In So Kweon<sup>1</sup>

Received: date / Accepted: date

**Abstract** The performance of coded exposure imaging critically depends on finding good binary sequences. Previous coded exposure imaging methods have mostly relied on random searching to find the binary codes, but that approach can easily fail to find good long sequences, due to the exponentially expanding search space. In this paper, we present two algorithms for generating the binary sequences, which are especially well suited for generating short and long binary sequences, respectively. We show that the concept of low autocorrelation binary sequences, which has been successfully exploited in the field of information theory, can be applied to generate shutter fluttering patterns. We also propose a new measure for good binary sequences. Based on the new measure, we introduce two new algorithms for coded exposure imaging - a modified Legendre sequence method and a memetic algorithm. Experiments using both synthetic and real data show that our new algorithms consistently generate better binary sequences for the coded exposure problem, yielding better deblurring and resolution enhancement results com-

Hae-Gon Jeon  
E-mail: hgjeon@rcv.kaist.ac.kr

Joon-Young Lee  
E-mail: jolee@adobe.com

Yudeog Han  
E-mail: ydhan@add.re.kr

Seon Joo Kim  
E-mail: seonjookim@yonsei.ac.kr

In So Kweon (Corresponding author)  
E-mail: iskweon77@kaist.ac.kr

<sup>1</sup> The School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea (J.-Y Lee and Y. Han are currently at Adobe research and Agency for Defense Development, respectively.)

<sup>2</sup> Yonsei University, Seoul, Republic of Korea

pared to previous methods of generating the binary codes.

**Keywords** Coded exposure · Fluttering pattern · Motion deblurring · Computational photography

## 1 Introduction

Image deblurring is one of the most common problems in computer vision. The goal of deblurring is to recover a sharp latent image from an image that is blurred due to the motion of the subject or the camera. Although a solution for the image deblurring problem has been sought for the last few decades, it remains a challenging problem today.

One deblurring approach that has shown promising results is to tackle the problem in an active manner by changing the way images are captured in a camera.

The technique, which is called coded exposure photography ([Raskar et al. 2006](#)), flutters the camera's shutter open and closed in a special manner within the exposure time, in order to preserve the spatial frequencies, thereby simplifying the subsequent deblurring problem. A key aspect in coded exposure imaging is the generation of the fluttering pattern of the shutter (the binary sequence). Well-chosen fluttering patterns preserve the spatial frequencies in the captured blurred image while also minimizing deconvolution noise.

Although fluttering the camera's shutter results in less received light compared to a conventional shutter, the coded exposure process using a flutter sequence reported in ([Raskar et al. 2006](#)) still showed a performance gain when the incident illumination was lower than 100 lux (indoor lighting) in ([Cossairt et al. 2013](#)). Performance can also be significantly improved if fluttering patterns with a flatter spectrum are used.

Long binary sequences are essential for various applications of the coded exposure process. In previous works, a near optimal binary sequence was computed using a randomized linear search (Agrawal and Raskar 2007; Agrawal and Xu 2009; Raskar *et al.* 2006) or a priority search (McCloskey 2010) over the space of potential sequences. But while these methods can be used to generate short sequences, they are not suitable for computing long binary sequences because of the large search space.

For super-resolution from a single blurred image in (Agrawal and Raskar 2007), higher resolution enhancement was achieved when long fluttering patterns with flat spectrums were utilized. In (Gorthi *et al.* 2013), a coded fluorescence imaging device required long sequences (which can extend to 50 times the traditional exposure) to capture fast traveling cells with a high signal to noise ratio.

Finding binary sequences with low autocorrelation is a problem that has been deeply studied in the fields of information theory and physics, because it relates to many applications in telecommunications (e.g., synchronization, pulse compression and, especially, radar), physics (e.g., Ising spin glasses) and chemistry (Gallardo *et al.* 2009). The Barker sequence (Borwein *et al.* 2007) is considered to be the optimal binary sequence, however the sequence length only goes up to 13. Mertens (Mertens 1996) proposed an exhaustive search method to obtain near-optimal binary sequences up to length 48, and Gallardo *et al.* (Gallardo *et al.* 2009) introduced a memetic algorithm for generating near-optimal sequences up to 60. These methods are not universally applicable due to the heavy computational burden. Instead, theoretical methods have been widely used to generate near-optimal binary sequences of any length.

Representative methods include the Jacobi (Jedwab 2005), the modified Jacobi (Xiong and Hall 2011), and the Legendre sequence. Among those methods, the Legendre sequence has shown advantages over the other methods, especially in terms of computational time and the autocorrelation measure (Jedwab 2005). Baden (Baden 2011) presented a sequence optimization algorithm using those theoretical sequences as initial estimates.

In this paper, we introduce two algorithms for generating binary sequences for coded exposure imaging. We show that the low autocorrelation binary sequence concept can be applied to generate the fluttering pattern of a shutter, and propose the modified Legendre sequence and a memetic algorithm. The proposed algorithms consistently generate better binary sequences for the coded exposure problem, yielding better deblurring and resolution enhancement results compared to previous methods of generating the binary codes.

The two algorithms that we propose in this paper are complementary to each other in terms of sequence length. For generating short sequences, the memetic search algorithm has the advantages of sequence quality and reasonable computational time, while the modified Legendre sequence is useful for generating longer sequences because of its computational efficiency.

We found that the memetic algorithm outperformed previous search-based algorithms (Raskar *et al.* 2006; Agrawal and Raskar 2007; McCloskey *et al.* 2012) with similar computational time by efficiently exploring the search space. The modified Legendre sequence generated good sequences in a much shorter time (several orders of magnitude), especially when the sequence length was large, in cases where the previous methods failed to find good binary codes due to the exponentially expanding search space.

## 2 Previous Works

Image deblurring is a classic problem in computer vision and has been actively studied for the last several decades. Traditional solutions to the problem include the Richardson-Lucy (Lucy 1974; Richardson 1972) and Wiener filters (Wiener 1964), but several new directions have been explored recently to enhance deblurring performance. Fergus *et al.* (Fergus *et al.* 2006) took a statistical approach in a variational Bayesian framework by using a natural image prior to the image gradients, while Shan *et al.* (Shan *et al.* 2008) incorporated spatial parameters to enforce the natural image statistics, using a local ringing suppression. In (Levin *et al.* 2007), Levin *et al.* proposed a solution for defocus blur using a coded aperture and a sparse natural image prior to produce sharper edges and reduce undesirable ringing artifacts. We refer to (Levin *et al.* 2009) for a comprehensive review of the deblurring literature.

In (Raskar *et al.* 2006), Raskar *et al.* introduced coded exposure photography, a motion deblurring method using the fluttering shutter. Rather than having the shutter open for the entire exposure duration, they fluttered the camera's shutter open and closed during the exposure using a binary pseudo-random sequence (Raskar *et al.* 2006). With the fluttered shutter, spatial details in the blurred image are preserved, making the deconvolution a well-posed problem. Tai *et al.* presented a spatially varying PSF estimation algorithm which jointly utilized a coded exposure camera and simple user interactions in (Tai *et al.* 2010), while McCloskey *et al.* further addressed the problem of motion deblurring using coded exposure, by analyzing the design and estimation of the coded exposure PSF in (McCloskey *et al.* 2012).

The idea of coded exposure photography was also extended to a resolution enhancement application in (Agrawal and Raskar 2007). As expected, the fluttering pattern of the camera shutter plays a critical role in determining the performance of the coded exposure imaging. Agrawal and Xu proposed a method for finding optimized codes for both PSF estimation and invertibility in (Agrawal and Xu 2009). McCloskey presented the idea that the shutter sequence must be dependent on the object velocity, and proposed a method for computing the velocity-dependent sequences in (McCloskey 2010).

To actually compute the binary sequence, previous works have relied on either a random sample search (Agrawal and Raskar 2007; Agrawal and Xu 2009; Raskar et al. 2006) or a priority search (McCloskey 2010) over the space of potential sequences. Natural image statistics were incorporated to generate binary sequences for a coded aperture (Zhou et al. 2011) and coded exposure (McCloskey et al. 2012). While these search based methods can produce good binary sequences of short lengths, they are computationally infeasible for long sequences because of the large search space.

This paper is an extended version of (Jeon et al. 2013). We additionally introduce a new search based algorithm, which will be presented in Section 5. The new algorithm is especially suited for generating short sequences, which is complementary to the modified Legendre sequence method presented in (Jeon et al. 2013). We also present additional experiments as well as more detailed discussion.

### 3 Measure of a Good Binary Sequence

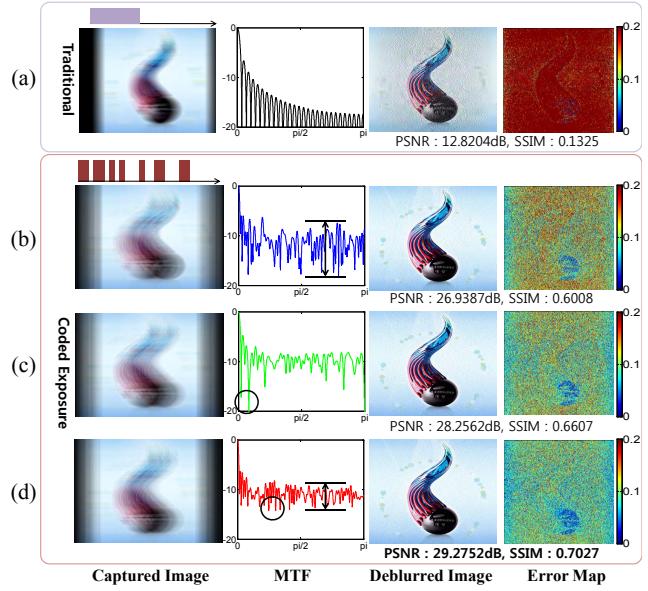
Assuming spatially-invariant motion blur, the blur process is modeled as follows:

$$\mathbf{B} = \mathbf{AI} + \mathbf{n}, \quad (1)$$

where  $\mathbf{B}$ ,  $\mathbf{I}$  and  $\mathbf{n}$  represent the blurred image, the latent sharp image, and the noise, respectively. The matrix  $\mathbf{A}$  is called the smearing matrix, which describes the convolution of the latent input image with a point spread function.

The principal idea behind the coded exposure is to improve the invertibility of the imaging process (the invertibility of the smearing matrix  $\mathbf{A}$ ) through the fluttered shutter (see Fig. 1 (a)(b)). Denoting a binary sequence of length  $n$  as  $U = [u_0, \dots, u_{n-1}]$ , a near-optimal binary code is computed through a randomized linear search with the following conditions in (Raskar et al. 2006):

$$(i) \underset{U}{\operatorname{argmax}} \min(|\mathcal{F}(U)|),$$



**Fig. 1** Coded exposure and the measure of a good binary sequence. (a) In a traditional camera, some information are lost at frequencies with MTF value of 0, making the deblurring problem an ill-posed problem. (b) By using the coded exposure, the information are preserved and the deblurring problem becomes invertible. (c) The merit factor is a good measure of the variance of the MTF (deconvolution noise), but it may make the spectrum partially peaky. (d) The proposed *coded factor* is a good measure of a binary sequence, which minimizes the variance of the MTF while maximizing the lowest MTF value.

$$(ii) \underset{U}{\operatorname{argmin}} \operatorname{var}(|\mathcal{F}(U)|) \text{ or } \underset{U}{\operatorname{argmin}} \operatorname{mean}(\mathbf{A}^T \mathbf{A})^{-1}$$

where  $\mathcal{F}(U)$  is the discrete Fourier transform of the binary sequence and its absolute value  $|\mathcal{F}(U)|$  is a magnitude of frequency response of binary sequence (MTF: Modulation Transfer Function). Condition (i) relates to preserving the spatial frequency in a blurred image, and condition (ii) describes the variance of the MTF or the deconvolution noise.

#### 3.1 Merit Factor

As mentioned earlier, finding binary sequences is also of importance in the field of information (coding) theory. In information theory, the *merit factor* is widely used as the criterion of “goodness” for binary sequences whose aperiodic autocorrelations are collectively small. For a binary sequence  $U = [u_0, u_1, \dots, u_{n-1}]$ , the merit factor  $M(U)$  is defined as follows:

$$M(U) = \frac{n^2}{2 \sum_{k=1}^{n-1} a_k^2}, \quad (2)$$

where  $a_k$  is the *aperiodic autocorrelation* at shift  $k$  given by

$$a_k = \sum_{i=0}^{n-k-1} u_i u_{i+k}. \quad (3)$$

The merit factor is closely related to the signal to self-generated noise ratio, which corresponds to the deconvolution noise in the coded exposure imaging. In (Jensen et al. 1991), the relation between the merit factor and the spectral properties of the sequence is denoted as

$$\sum_{k=1}^{n-1} a_k^2 = \frac{1}{2} \int_{-\pi}^{\pi} [|F(U)|^2 - n]^2 d\omega, \quad (4)$$

where  $\omega$  represents frequency. For a fixed sequence of length  $n$ , Eq. (4) shows that the merit factor measures how much the amplitude spectrum of the sequence deviates from the constant value  $n$ . Therefore, a sequence with a higher merit factor has a flatter MTF. This corresponds to condition (ii) of the binary sequence measure for coded exposure and we can rewrite the merit factor as follows:

$$M(U) = \frac{n^2}{\int_0^1 [|F(U)|^2 - n]^2 d\theta} \simeq \frac{n^2}{\text{var}(|F(U)|)}. \quad (5)$$

### 3.2 Coded Factor

While the merit factor is a good criterion for measuring the MTF variance, it can make the amplitude spectrum partially peaky, which prevents the system from preserving the details of a blurred image. As shown in Fig. 1(c), the MTF of the binary sequence shows a peaky spectrum even though its merit factor is about 6.1. In contrast, the binary sequence in Fig. 1(d) is more suitable for coded exposure imaging, even though its merit factor is about 5.8. To deal with this problem, we define a new measure called the *coded factor* ( $F_C$ ) to measure the quality of a binary sequence for coded exposure imaging:

$$F_C(U) = M(U) + \lambda \min[\log(|F(U)|)], \quad (6)$$

where  $\lambda$  is the weighting parameter for balancing the two terms and  $\log$  is used for normalizing the scales between the two terms.

We should note that Eq. (3) is derived with the binary sequence taking the value  $\{-1, 1\}$ . However, the binary sequence for the coded exposure should take the value  $\{0, 1\}$ , since the value -1 is physically infeasible. If we change the sequence  $u_i \in \{-1, 1\}$  to  $\hat{u}_i \in \{0, 1\}$

by substituting 0 for -1, the aperiodic autocorrelation of  $U$  is computed as

$$a_k = 4\hat{a}_k + 4m \sum_{i=0}^{n-k-1} (\hat{u}_i + \hat{u}_{i+k} + 3\mu - 0.5), \quad (7)$$

where  $\hat{a}$  and  $\mu$  represent the *autocovariance* and the mean of  $\hat{C}$ , respectively. The derivation of Eq. (7) is provided in Appendix 1. In Eq. (7),  $m = \mu - 0.5$ , which becomes 0 with the assumption that the sequence is balanced with an equal number of zeros and ones for optimal autocorrelation properties (Lempel et al. 1977). Therefore, we use the following equation for computing the merit factor from a binary sequence of 0's and 1's.

$$a_k \approx 4\hat{a}_k. \quad (8)$$

### 4 Modified Legendre Sequence for Coded Exposure

Although the importance of both terms in Eq. (6) are addressed in (Raskar et al. 2006), a solution for finding a good binary sequence that simultaneously satisfies both conditions is not provided. Instead, they rely on a randomized linear search that only considers  $\min[\log(|F(U)|)]$  in Eq. (6). To deal with this issue, we find a solution that can take both terms into account and return the maximum coded factor  $F_C$ . For this, we turn to the Legendre sequence.

The Legendre sequence (Golay 1983) is a binary sequence with a high merit factor, and is among the most popular choices for generating binary sequences in many different fields. The Legendre sequence of a prime length  $n$  is defined as

$$u_i = \begin{cases} 1 & \text{if } i = 0, \\ \left(\frac{i}{n}\right) & \text{if } i > 0, \end{cases} \quad (9)$$

where  $u$  and  $i$  represent an element value and the index of the sequence, respectively.  $\left(\frac{i}{n}\right)$  is the *Legendre symbol* that takes the value 1 if  $i$  is a *quadratic residue* modulo  $n$  and the value 0 otherwise.<sup>1</sup>

The advantages of using the Legendre sequence over a random binary sequence search for the coded exposure include higher quality sequences with high merit factor, as well as much less computational load, since the Legendre sequence is solved in a closed form. Although the Legendre sequence can ensure a high merit factor  $M(U)$ , it does not guarantee the highest coded factor since it does not consider  $|F(U)|$ . Therefore, to further improve the quality of the Legendre sequence

<sup>1</sup> Note that either  $\{0, 1\}$  or  $\{-1, 1\}$  can be used to represent the sequence value as shown in (No et al. 1996).

for coded exposure imaging, we propose an algorithm for generating a modified Legendre sequence by applying three sequence operations: *rotating*, *appending*, and *flipping* to find the sequence with the maximum *coded factor*  $F_C(U)$  in Eq. (6).

#### 4.1 Rotating

The merit factor of a Legendre sequence can be improved by rotating the sequence in a cyclic manner as shown in (Høholdt and Jensen 1988). For a given sequence  $U$ , an  $r$ -rotated Legendre sequence  $V^r$  is defined as

$$V^r = (U_{r+1:n}; U_{1:r}), \quad (10)$$

where  $U_{i:j}$  is the sub-sequence of  $U$  from the  $i^{\text{th}}$  to the  $j^{\text{th}}$  element and  $(;)$  represents an operator for concatenating the two sequences. We search for the enhanced sequence in terms of the *coded factor* among all the candidate sequences  $V^r$  ( $0 \leq r \leq n - 1$ ).

#### 4.2 Appending

In (Borwein et al. 2004), Borwein *et al.* proved that appending the initial part of a rotated Legendre sequence to itself can improve the merit factor of the sequence. We adopt the appending operation to improve the sequence quality as well as to resolve a restraint of the Legendre sequence that it is only defined for a length of a prime number.

From an  $r$ -rotated Legendre sequence  $V^r$ , a  $t$ - appended Legendre sequence is obtained by appending the first  $t$  ( $0 \leq t \leq n - 1$ ) elements of the sequence to itself, and it is denoted as  $Y^t = (V^r; (V^r)_{0:t-1})$ . Using the sequence appending operation, the modified Legendre sequence with a length  $m$  can be generated from any rotated Legendre sequence with a prime length  $n$  for  $\frac{m}{2} \leq n \leq m$ .

#### 4.3 Flipping

In a recent work (Baden 2011), Baden presented an efficient optimization method for the merit factor of binary sequences by deriving a formulation for measuring the change in the merit factor by the change of value in an element (flipping) in the sequence. The formulation is given by

$$\delta_j = -8y_j((\Lambda \star Y)_j + (\Lambda \star Y^\gamma)_{m+1-j}) + 8(Y \star Y^\gamma)_{m+1-2j} + 8(m-2), \quad (11)$$

where  $\delta_j$  is the change in the autocorrelation due to the flipping of the element  $j$ ,  $\star$  represents the convolution

---

#### Algorithm 1 Optimization by *Flipping*

---

```

1: procedure OPTIMIZESEQUENCE(sequence =  $Y$ )
2:    $\hat{Y} = Y$ 
3:    $F_C(\hat{Y})$  = coded factor of  $\hat{Y}$ 
4:    $\Delta$  = candidate set of  $\hat{Y}$ 
5:    $n_\Delta$  = the number of candidates in  $\Delta$ 
6:   for  $i = 1$  to  $n_\Delta$  do
7:      $F_C(Y') = \max(F_C)$  with  $i$ -bits flipping in  $\Delta$ 
8:     if  $F_C(Y') > F_C(\hat{Y})$  then
9:        $\hat{Y} = Y'$ 
10:      go to line 3
11:    end if
12:   end for
13:   return  $\hat{O}$ 
14: end procedure

```

---

operator,  $\Lambda = [a_1, \dots, a_m]$  is an aperiodic autocorrelation of the binary sequence  $Y$  of length  $m$ , and  $\gamma$  indicates the reversal of a sequence where  $y_j^\gamma = y_{m-j+1}$ .

From Eq. (11), a candidate set of element indices that are expected to improve the merit factor is chosen as

$$\Delta_1 = \{j | \delta_j > 0\}. \quad (12)$$

To extend the above optimization method to the *coded factor*, we compute the change in the minimum of MTF by a single-element flip, as follows:

$$\kappa_j = \begin{cases} \min |\mathcal{F}(Y) + \mathcal{F}(y_j)| - \min |\mathcal{F}(Y)| & \text{if } y_j = 0, \\ \min |\mathcal{F}(Y) - \mathcal{F}(y_j)| - \min |\mathcal{F}(Y)| & \text{if } y_j = 1, \end{cases} \quad \text{s.t. } \mathcal{F}(y_j) = e^{-i\omega(j+0.5)} \frac{2}{\omega} \sin \omega, \quad (13)$$

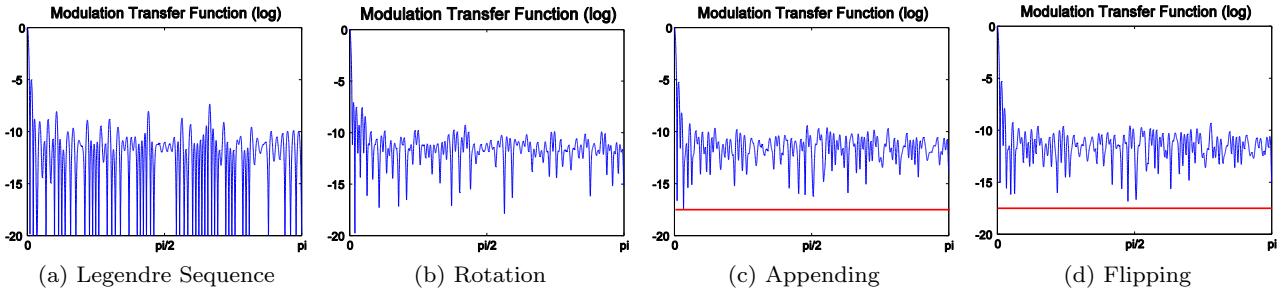
where  $\mathcal{F}(y_j)$  is the DFT of a single element  $y_j$ . Thus, the candidate set  $\Delta_2$  is determined by

$$\Delta_2 = \{j | \kappa_j > 0\}. \quad (14)$$

The two sets  $\Delta_1$  and  $\Delta_2$  are then combined to construct a new candidate set  $\Delta$  ( $\Delta = \Delta_1 \cup \Delta_2$ ). Since  $\Delta_1$  is related to the merit factor and  $\Delta_2$  is related to the MTF minimum, the new candidate set  $\Delta$  includes potential element indexes that can improve the coded factor  $F_C$  (Eq. 6). To determine the elements to flip among the candidates, we apply a variant of the *steep decent* algorithm in (Baden 2011), which is described in Algorithm 1. Since the number of candidates in  $\Delta$  is usually small, the computational load for Algorithm 1 is small.

#### 4.4 Algorithm Summary

Our framework for generating a binary sequence for the coded exposure imaging is summarized in Algorithm 2.



**Fig. 2** An example of MTF changes according to each sequence operation. The horizontal lines in (c)&(d) indicate the minimum MTF of the sequence (c).

### Algorithm 2 Modified Legendre Sequence

```

1: procedure GENERATESEQUENCE(length = m)
2:   n = {ni |  $\frac{m}{2} \leq n_i \leq m$ , ni is prime number}
3:   for all ni in n do
4:     Ui = Legendre Sequence of length ni in Eq. (9)
5:     Vi = Rotating(Ui)
6:     Yi = Appending(Vi)
7:   end for
8:   O = Yi with the highest coded factor
9:    $\hat{O}$  = Optimization by Flipping(O) in Algorithm 1
10:  return  $\hat{O}$ 
11: end procedure

```

To generate a sequence of length *m*, we first generate Legendre sequences with length **n**, which is a collection of prime numbers in the range between  $\frac{m}{2}$  and *m* (**n** = {*n<sub>i</sub>* |  $\frac{m}{2} \leq n_i \leq m$ , *n<sub>i</sub>* is prime number}). We then find the *r<sub>i</sub>*-rotated Legendre sequence using the *rotating* operation for each Legendre sequence and apply the *Appending* operation for all the rotated sequences to make length *m* sequences. Among the candidate sequences, we select a sequence with the highest *coded factor* and perform the optimization using the *flipping* operation. An example of MTF changes according to each sequence operation is shown in Fig. 2.

### 5 Memetic Algorithm for Coded Exposure

In previous searching based algorithms (Raskar et al. 2006; McCloskey et al. 2012), there is a trade-off between the quality of binary sequences and the computational time. Although our modified Legendre sequence rarely suffers from the trade-off and consistently generates high quality sequences, a searching based approach is still useful, especially for short sequences, because a globally optimal solution can be found when an exhaustive search is performed.

In this section, we present a new algorithm to take advantage of the searching based approach. To extend the applicability of the searching based approach, we apply an efficient searching algorithm based on the memetic

### Algorithm 3 Memetic Algorithm

```

1: procedure SEARCHSEQUENCE(length = m)
2:   Initialize a candidate set  $\Omega$  of binary sequences.
3:   while termination condition is not satisfied do
4:      $\{O, U\}$  = Select two sequences in  $\Omega$ 
5:     O = Recombination(O, U, Pr)
6:     O = Mutation(O, Pm)
7:     O = LocalSearch(O)
8:      $\Omega$  = Update( $\Omega, O$ )
9:   end while
10:   $\hat{O}$  = Select the best sequence in  $\Omega$ 
11:  return  $\hat{O}$ 
12: end procedure

```

algorithm (Chen et al. 2011), which is widely used in the evolutionary computing field. The memetic algorithm is an extension of the traditional genetic algorithm with a local searching method.

Based on the traditional genetic algorithm, a candidate set of binary sequences are first generated. Then, we stochastically perform a recombination and mutation of codes in order to avoid getting stuck in suboptimal regions. To find a suboptimal sequence in a candidate sequence, we execute a local search and update the candidate set by replacing the sequence having the highest coded factor in the set with the optimized sequence. The procedure is repeated until a termination condition is satisfied. The overall procedure is summarized in Algorithm 3, where *P<sub>r</sub>* and *P<sub>m</sub>* represent the occurrence probabilities of the recombination and the mutation procedure, respectively. In the following subsections, we describe more details about each step.

#### 5.1 Memetic Algorithm

The memetic algorithm is an iterative method that updates a set of candidate solutions by an evolutionary process. The first step in the memetic algorithm is to generate an initial candidate set. In general, a memetic algorithm organizes an initial candidate set by generating random sequences (Gallardo et al. 2009). The per-

formance of the memetic algorithm degrades when the sequence length  $n$  is increased since the search space exponentially grows with the sequence length. As a better choice for an initial set, we adopt skew-symmetric sequences. In (Mertens 1996), Mertens showed that skew-symmetric sequences reduce the effective size of the computation by a factor of 2 when searching long length sequences. A skew-symmetric sequence  $S$  is defined as (Golay 1977)

$$s_{L+j} = \begin{cases} s_{L-j} & \text{if } j \text{ is even,} \\ \tilde{s}_{L-j} & \text{if } j \text{ is odd,} \end{cases} \quad \text{for } j = 1, \dots, L-1$$

$$\text{s.t. } L = \begin{cases} n/2 & \text{if } n \text{ is even,} \\ (n+1)/2 & \text{if } n \text{ is odd,} \end{cases} \quad (15)$$

where  $s_i$  is the  $i^{\text{th}}$  element of  $S$  and  $\tilde{\cdot}$  is a negation operation. The first half of a skew-symmetric sequence  $S_{1:L}$  is randomly generated.

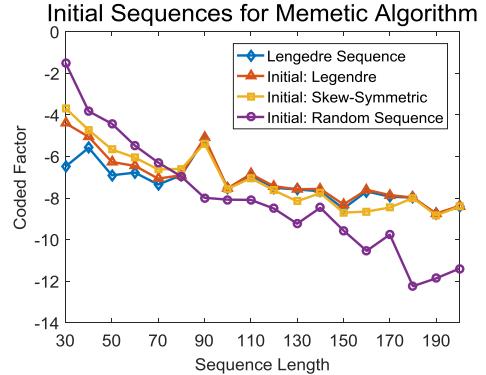
In Fig. 3, we experimentally demonstrate the change in performance based on the choice of initial sequences<sup>2</sup>. For short length sequences, randomly generated sequences are better than skew-symmetric sequences ( $n \leq 80$ ). On the other hand, skew-symmetric sequences show better performances for long length sequences. We observe that the randomly generated sequences are good initial seeds because the memetic algorithm can make various attempts to find optimal sequences in small search spaces while well-constrained initial seeds like the skew-symmetric sequences work well in large search spaces.

We also tested how performance varied when the Legendre sequences consisted of the initial sets. The Rotating in Sec. 4.1 and the Appending in Sec. 4.2 can create multiple candidate sets. As shown in Fig. 3 (blue and red lines), the coded factors are visibly increased for short sizes, but the performance improvement for long sequences is marginal. Since Legendre sequences provide a finite set of candidates, we cannot expect Legendre sequences to provide enough freedom for the memetic algorithm to explore the large space.

To avoid getting stuck in suboptimal solutions, recombination and mutation are performed as a meta-heuristic scheme. For the recombination, we randomly select two sequences  $\{O, V\}$  in the candidate set and make a new sequence  $O^*$  by combining the two subsequences as

$$O^* = (O_{1:n}; V_{n+1:m}), \quad (16)$$

where  $O_{i:j}$  is the sub-sequence of  $O$  from the  $i^{\text{th}}$  to the  $j^{\text{th}}$  element,  $n$  is a slicing point of the two sequences, and  $(;)$  represents an operator for concatenating the



**Fig. 3** Comparisons of initial sequences generated from randomly generated sequences, skew-symmetric sequences, and Legendre sequences with regards to the sequence length.

two sequences. The slicing point  $n$  is determined randomly and the occurrence probability of recombination is set to 0.9.

In the mutation step, we randomly choose one to three bits of a sequence and flip the chosen bits as recommended in (Militzer et al. 1998). Militzer et al. (Militzer et al. 1998) reported that multi-bit flipping in the mutation step is beneficial for achieving low-autocorrelation binary sequences compared to single bit flipping. We set the occurrence probability of mutation to 0.1.

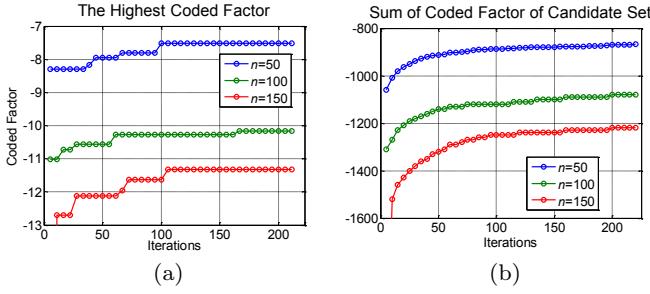
The local search in the memetic algorithm reduces the likelihood of a premature convergence and improves the fitness of a candidate sequence. For the local search method, we used the flipping operation as described in Sec. 4.3. By exploiting the flipping operation, we obtained a local optimal solution from a candidate sequence with low computation time, since the flipping operation explores a local optimal solution in an efficient optimization manner instead of an intensive local search.

## 5.2 Termination Condition

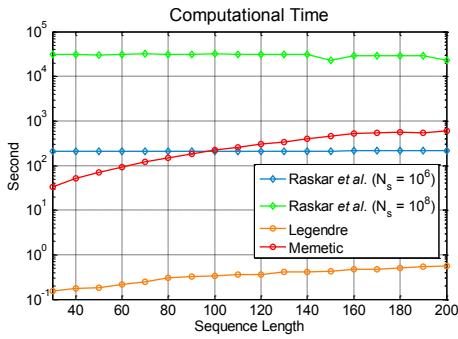
The memetic algorithm is performed iteratively until a termination condition is satisfied. The termination condition is generally determined either by a fixed number of iterations or the fitness of the highest ranking solution (Michalewicz 1996). However, without careful heuristic parameter tunings, those termination conditions may lead to over computations or premature convergence.

We show the coded factor of the highest ranking solution in Fig. 4 (a). In the figure, the highest coded factor has a tendency of stair-like transitions with regards to the number of iterations. To set a better termination

<sup>2</sup> We set the number of candidates to 100 for the experiment.



**Fig. 4** Experiments with different termination conditions on in the sequence length  $n$ . (a) The change of in the highest coded factor according to the number of iterations. (b) The change of in the sum of of the coded factor of all candidates according to the number of iterations.



**Fig. 5** Computational times(log scale) for generating binary sequences according to sequence length. The computational time of the method (Raskar et al. 2006) depends on the number of samples, while the proposed method requires a shorter time for all sequence lengths.

condition, we use the total sum of coded factors of all candidate sequences. Our algorithm terminates when the sum of coded factors does not change within a certain number of iterations. Fig. 4 (b) shows the change in the total sum of coded factors versus the number of iterations. The proposed termination condition explores all possibilities for improving the coded factor of the candidates. Then, we select one binary sequence in the candidate sets with the best coded factor Eq. (6).

Fig. 5 shows a comparison of the computational times of the different algorithms. The computational time of the random sample search method (Raskar et al. 2006) is determined by the number of iterations, while the computational times of the modified Legendre sequence and the memetic algorithm increase according to the sequence length. As will be shown in Sec. 6, the proposed memetic algorithm delivers much better performance with shorter computational time than the random sample search method.

## 6 Experiments

To evaluate the performance of the proposed algorithm, we conducted many coded exposure deblurring experiments using both synthetic and real-world datasets. We compared our results with the results obtained with the methods proposed by Raskar *et al.* in (Raskar et al. 2006) and by McCloskey *et al.* in (McCloskey et al. 2012). For our method and the (Raskar et al. 2006) method, binary sequences of length  $[30, 40, \dots, 200]$  were generated. We used the code by the author<sup>3</sup> to generate the sequences for the (Raskar et al. 2006) method as well as to deblur the images.

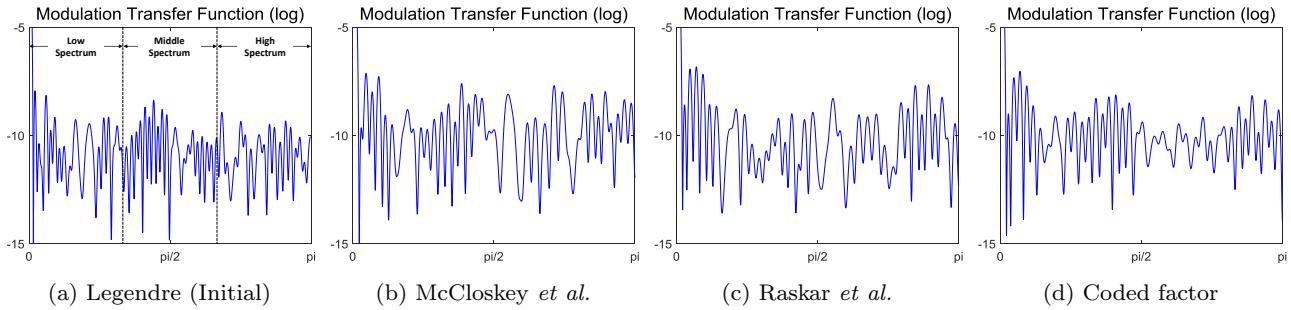
The number of random samples  $N_s$  in the (Raskar et al. 2006) method were set to  $10^6$  and  $10^8$  to manage the tradeoff between the computational time and the sequence quality. The average computational times taken to generate the sequences are shown in Fig. 5. The sequences for the (McCloskey et al. 2012) method were provided by the author for length  $[50, 60, \dots, 200]$ . The  $\lambda$  in Eq. (6) was set to 8.5 for all of our experiments. We used two deconvolution algorithms: (1) the matrix inversion approach for the sake of comparing the performance of previous methods and the proposed method, and (2) non-blind deconvolution with hyper-Laplacian prior (Krishnan and Fergus 2009) to maximize the quality of the deblurred images.

### 6.1 Quantitative Evaluations

We performed synthetic experiments for quantitative evaluations. The synthetic data consisted of 29 high quality images downloaded from Kodak Lossless True Color Image Suite (Franzen 1999). Blurred images were simulated by 1D filtering with the binary sequences generated by each method and then adding intensity dependent Gaussian noise with a standard deviation  $\sigma = 0.01\sqrt{i}$ , where  $i$  is the noise-free intensity of the blurred images in  $[0, 1]$  (Schechner et al. 2007). The peak signal-to-noise ratio (PSNR) and the gray-scale structural similarity (SSIM) (Wang et al. 2004) were used as the quality metrics, which were calculated by averaging the results of 29 synthetic images.

First, to show the effectiveness of the coded factor as a measure of a good binary sequence for coded exposure, we compared deblurring results using the binary sequences generated by the merit factor and the coded factor, which are shown in Fig. 6. The sequences generated by the coded factor show stable performance, while the sequences generated by the merit factor sometimes

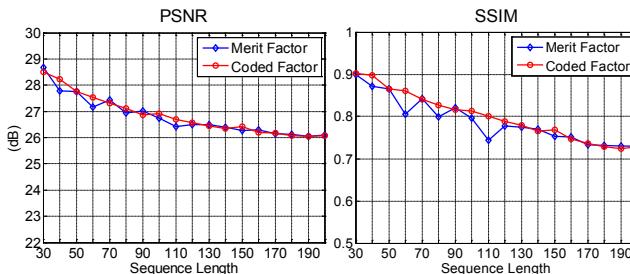
<sup>3</sup> [www.umiacs.umd.edu/~aagrawal/MotionBlur/SearchBestSeq.zip](http://www.umiacs.umd.edu/~aagrawal/MotionBlur/SearchBestSeq.zip)



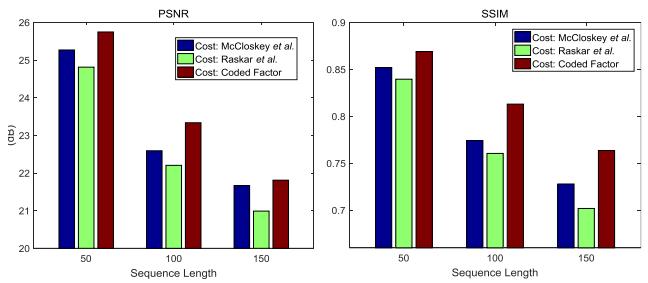
**Fig. 8** An example of MTF changes according to cost functions. A Legendre sequence with the length 100 is used as an initial sequence and the optimization is done by partially flipping all combinations of change in 1, 2, and 3 bits.

	Low spectral bands ( $0 \sim \pi/3$ )			Middle spectral bands ( $\pi/3 \sim 2\pi/3$ )			High spectral bands ( $2\pi/3 \sim \pi$ )		
	min.	mean	var.	min.	mean	var.	min.	mean	var.
Legendre (Initial)	-15.2341	-10.5242	4.5454	-14.8233	-10.8800	1.3486	-13.6837	-11.0215	1.0372
McCloskey <i>et al.</i>	-15.1381	<b>-9.7462</b>	5.6573	-13.6071	<b>-10.0992</b>	2.1356	<b>-12.6881</b>	<b>-9.8547</b>	1.4912
Raskar <i>et al.</i>	<b>-13.5823</b>	-9.7978	5.9692	-13.5748	-10.6420	1.4908	-13.3213	-10.1791	1.5939
Coded factor	-14.6124	-9.7467	<b>5.2634</b>	<b>-13.4124</b>	-10.1110	<b>0.9451</b>	-13.9681	-10.2067	<b>0.8688</b>

**Table 1** Numerical analysis of MTFs in Fig. 8. Minimum, mean and variance of MTFs of the sequences are reported according to spectral bands. We mark the best performance in bold. For minimum and mean, higher is better, and for variance, lower is better.



**Fig. 6** Comparison of the merit factor and the coded factor as a measure of a good binary sequence.



**Fig. 7** Quantitative evaluations of cost functions proposed in (McCloskey *et al.* 2012; Raskar *et al.* 2006) and the coded factor for a local optimization applied to the Legendre sequences.

work poorly, especially in terms of the SSIM, due to the peaky spectrum, as previously shown in Fig. 1(c).

Additionally, we compared the coded factor to the cost functions used in (Raskar *et al.* 2006) and (McCloskey *et al.* 2012)<sup>4</sup>. To measure the effectiveness of each cost function, we performed exhaustive local optimizations with the Legendre sequences as the initial seeds. We first computed each cost function of the initial binary sequence. We then found the 1-partial flip group, which is the collection of all possible combinations of

the initial binary sequence with one of the bits flipped. We did this again to find the 2-partial flip group and the 3-partial flip group, which have 2 and 3 bit-flip combinations, respectively. The cost function of each combination from these groups was then computed, in which the combination with the maximum cost was supplied as the seed for the next iteration until convergence.

In Fig. 7, we report the deblurring results using the optimized binary sequences. The optimized sequences from (McCloskey *et al.* 2012) and the coded factor exhibit better performance than those of (Raskar *et al.* 2006). As an example, Fig. 8 and Table 1 show the MTFs for a sequence length of 100 in Fig. 7 and the corresponding numerical analysis. The results imply that

<sup>4</sup> In (McCloskey *et al.* 2012), the weighted sum of 6 metrics was used: (1) the minimum of MTF, (2) the mean of MTF, (3) the variance of MTF, (4) the number of peaky frequencies, (5) weighted peaky frequencies and (6) the number of open chops.

the minimum frequencies of the MTF criterion are insufficient because they may amplify the variance of MTF. The difference between (McCloskey et al. 2012) and the coded factor is that the cost function of (McCloskey et al. 2012) preserves a higher contrast in the low frequency bands, because natural images have more power in low frequencies, while the coded factor measures the flatness of all spectral bands. Note that the binary sequence optimized from the coded factor shows a flat spectrum with the lowest variance in all spectral bands. We consider that the optimized sequences with the lowest variance in all spectral bands to have positive impact on better performance than that of (McCloskey et al. 2012).

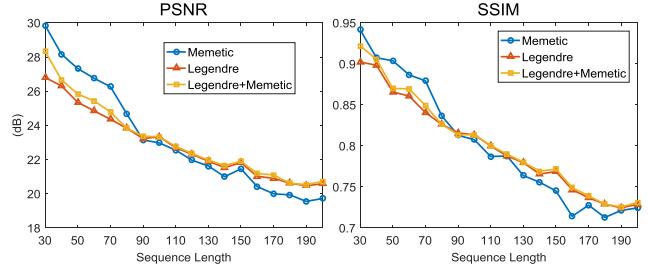
Next, we evaluated the improvement in performance when the Legendre sequences were used as initial seeds in the memetic algorithm in Fig. 9. As discussed in Sec. 5.1, the performance improvement with respect to the coded factor was marginal except for the short size sequences.

Fig. 10 shows comparisons of the deblurring results using the matrix inversion and the image prior on a synthetic dataset. For a more reliable evaluation, we additionally performed a synthetic experiment using 100 Berkeley Segmentation Dataset images (Martin et al. 2001). One of the synthetic results is shown in Fig. 11. Both the modified Legendre sequence and the memetic algorithm consistently produced good binary sequences for the coded exposure imaging, and the difference in performance between the proposed method and the other methods is amplified as the sequence length increases. When the sequence length is big, the previous methods fail to find good sequences, due to the large search space. As can be seen, the modified Legendre sequence becomes a valuable approach when a trade-off between computational time and the quality of sequences is considered, while the memetic algorithm is suitable for relatively short lengths of sequences, when it can effectively explore their search space in a reasonable computational time. While the PSNR and the SSIM values show small differences, the observed tendency remains consistent.

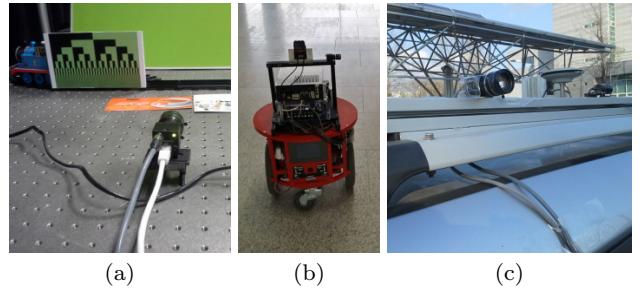
This issue of sequence length is an important one, since longer sequences are necessary for larger motion blurs or faster moving objects (Agrawal and Raskar 2007; McCloskey 2010). The work in (Agrawal and Raskar 2007) specifically emphasized the need for finding good long sequences.

## 6.2 Qualitative Evaluations

We implemented the coded exposure photography using a PointGrey Flea3 camera, which supports the *Trigger*



**Fig. 9** Quantitative evaluations on fluttering patterns generated by Memetic algorithm, Legendre sequence and memetic algorithm that uses the Legendre sequences as initial sequences.



**Fig. 12** We employ an off-the-shelf camera to capture coded exposure imagery of high-speed motion. (a) indoor experimental setup, (b) camera attached to a mobile robot, (c) camera attached to a car.

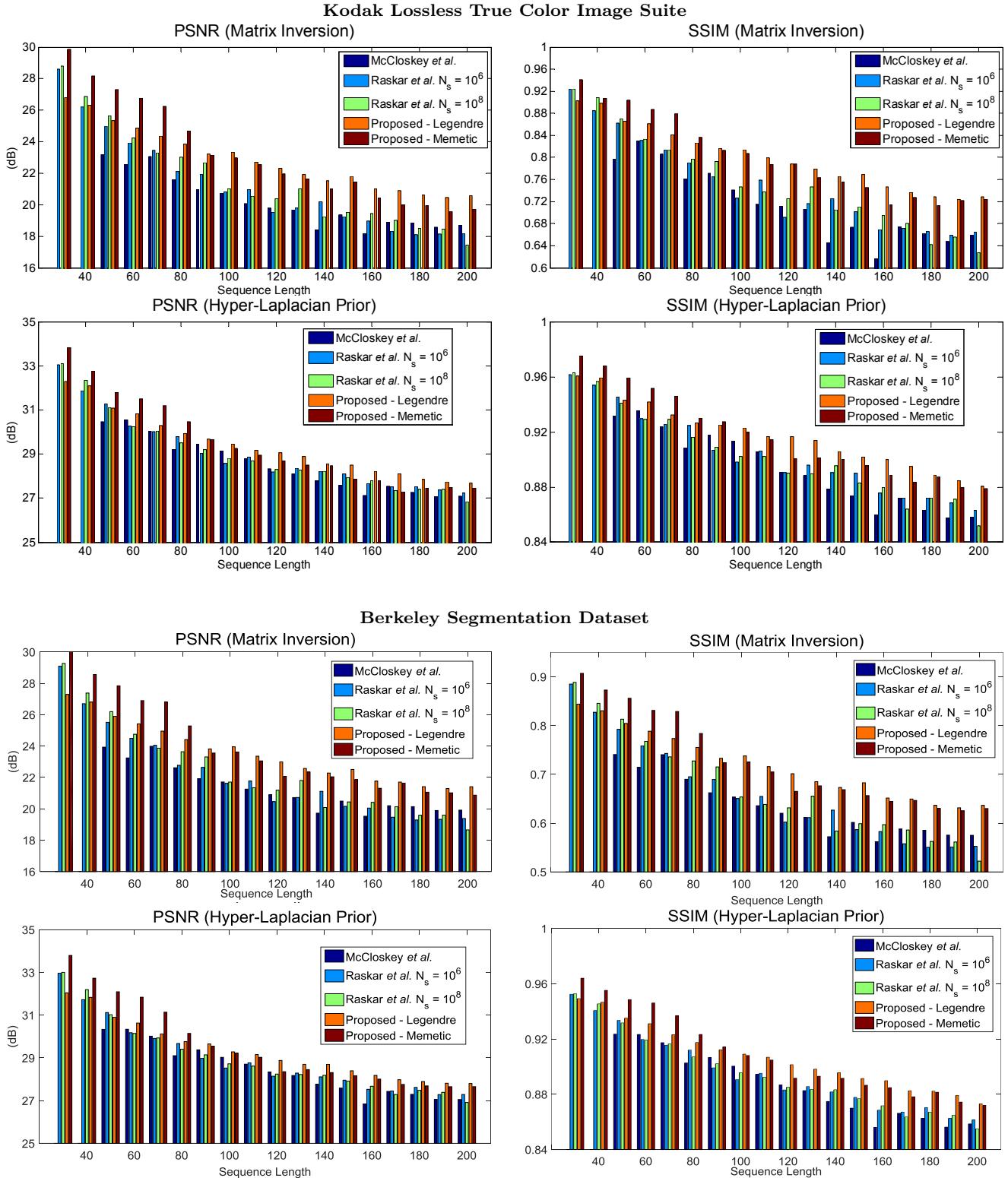
mode<sup>5</sup> which enables a multiple pulse-width trigger with a single readout (Fig. 12). When using Trigger mode 5, the frame rate of the Flea 3 camera dips to 5 frames per second. In our implementation, each shutter chop is 1 ms long, so a fluttering pattern 100 long has a capture time of 100 ms.

For qualitative evaluations, we chose the proposed methods that had the best performance for each sequence length in the quantitative evaluations (Fig. 10).

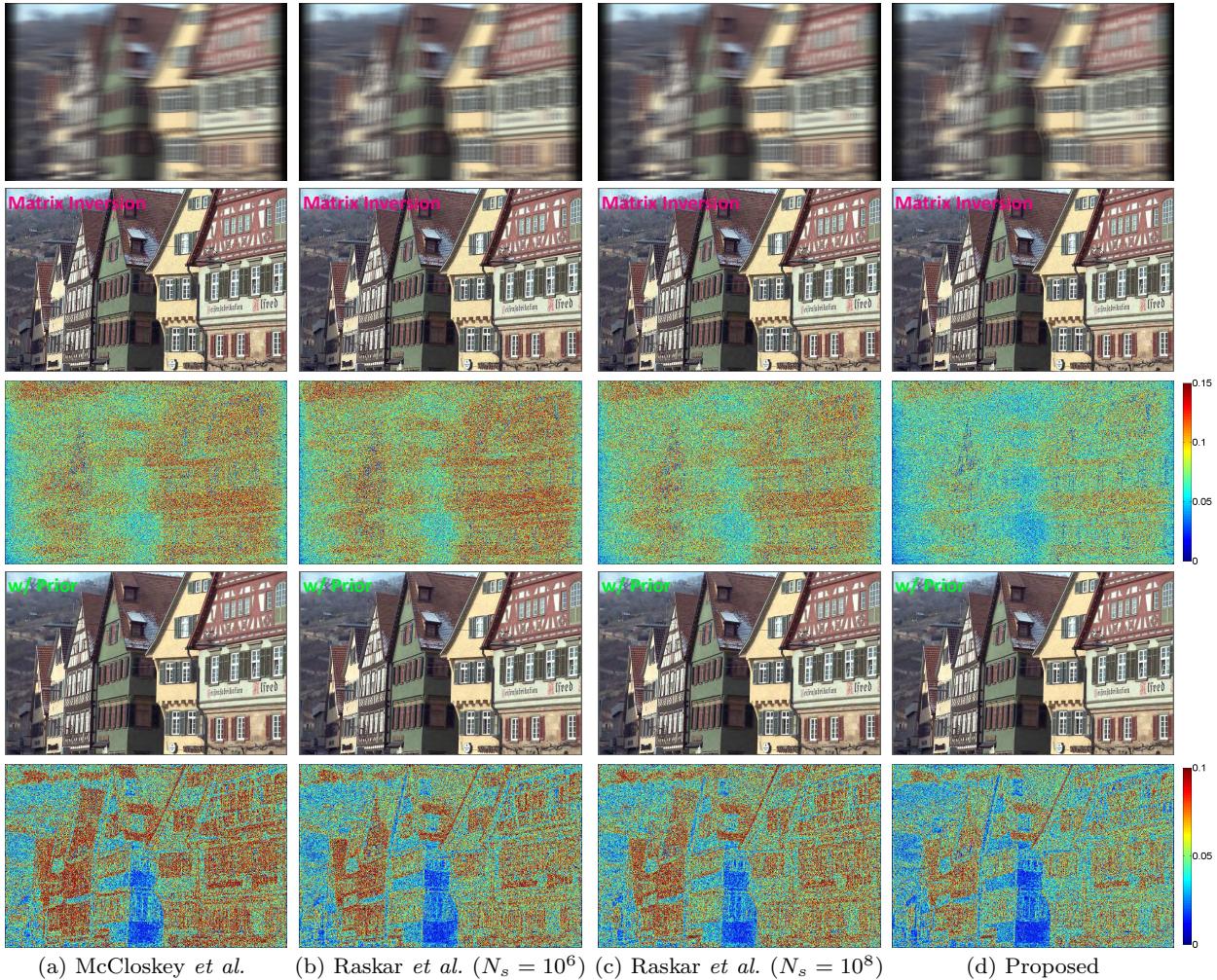
Fig. 13 and Fig. 14 show two examples of the deblurring results using the coded exposure with the fluttering patterns generated by the various methods. As expected, the deblurring results using the proposed fluttering patterns return the sharpest images, enabling the contents to be read, as opposed to the other results where the contents remain difficult to interpret.

Fig. 15 and Fig. 16 show more examples of our coded exposure imaging in action, by imaging static objects from a fast moving camera (about 0.5 meter per second). Similar to (Park et al. 2014), Fig. 15 shows the deblurring results for an image of a poster on a wall captured from a moving mobile robot. The deblurring

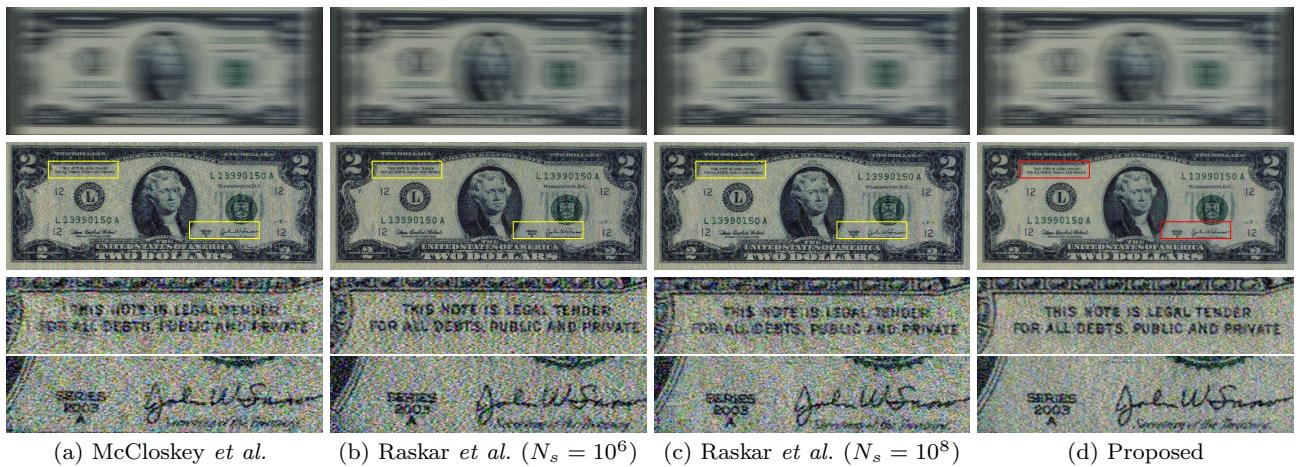
<sup>5</sup> w



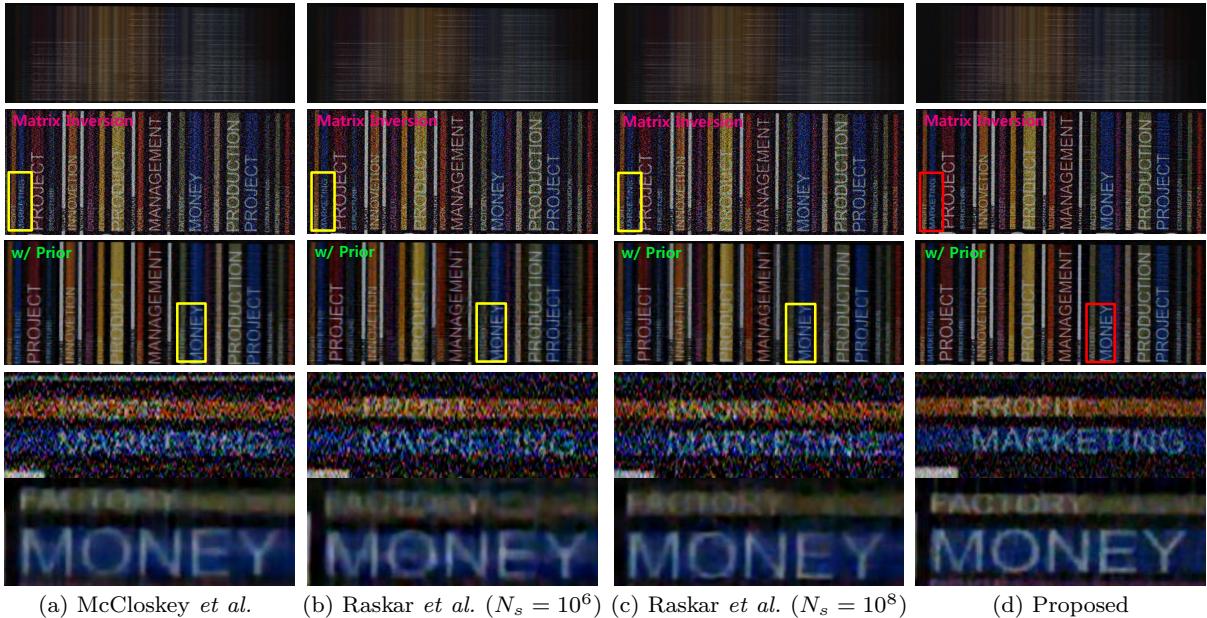
**Fig. 10** Comparison of the deblurring performance with synthetic dataset. For deblurring, (odd row) matrix Inversion and (even row) hyper-Laplacian prior are used. For the quality metric, (left) PSNR and (right) SSIM used. The difference in the performance amplifies as the sequence length increases; our method consistently generates good binary codes for the coded exposure imaging. Although the hyper-Laplacian prior enhances the quality of the deblurred images, the proposed binary sequences still show better performance than previous works.



**Fig. 11** One of the synthetic results with the fluttering patterns of length 50 generated by various methods. (1<sup>st</sup> row) blurred images. (2<sup>nd</sup> row) deblurred images using the matrix inversion and (3<sup>rd</sup> row) the error maps. (4<sup>th</sup> row) deblurred images using the hyper-Laplacian prior (Krishnan and Fergus 2009) and (5<sup>th</sup> row) the error maps.

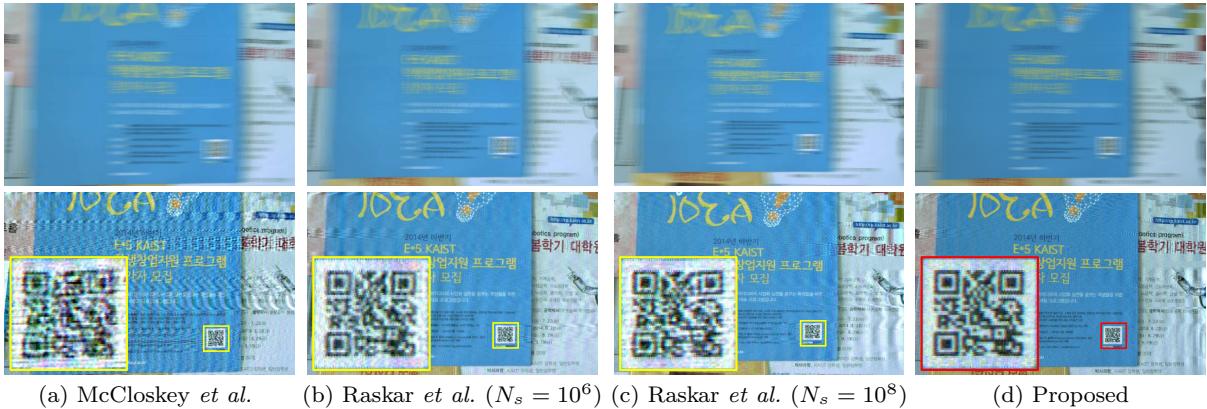


**Fig. 13** Comparison of the deblurring performance with the fluttering patterns of length 120 generated by various methods.

(a) McCloskey *et al.*(b) Raskar *et al.* ( $N_s = 10^6$ )(c) Raskar *et al.* ( $N_s = 10^8$ )

(d) Proposed

**Fig. 14** Comparison of the deblurring performance with the fluttering patterns of length 100 generated by various methods. The blurred images are deblurred by using the matrix inversion method (2<sup>nd</sup> row) and the hyper-Laplacian prior (3<sup>rd</sup> row).

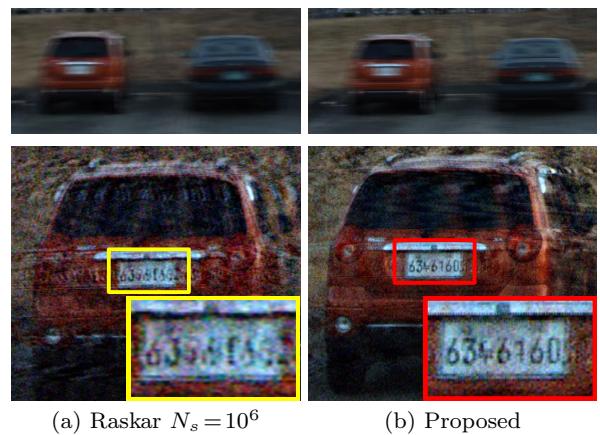
(a) McCloskey *et al.*(b) Raskar *et al.* ( $N_s = 10^6$ )(c) Raskar *et al.* ( $N_s = 10^8$ )

(d) Proposed

**Fig. 15** Comparison of the deblurring performance with the fluttering patterns of length 80 generated by various methods.

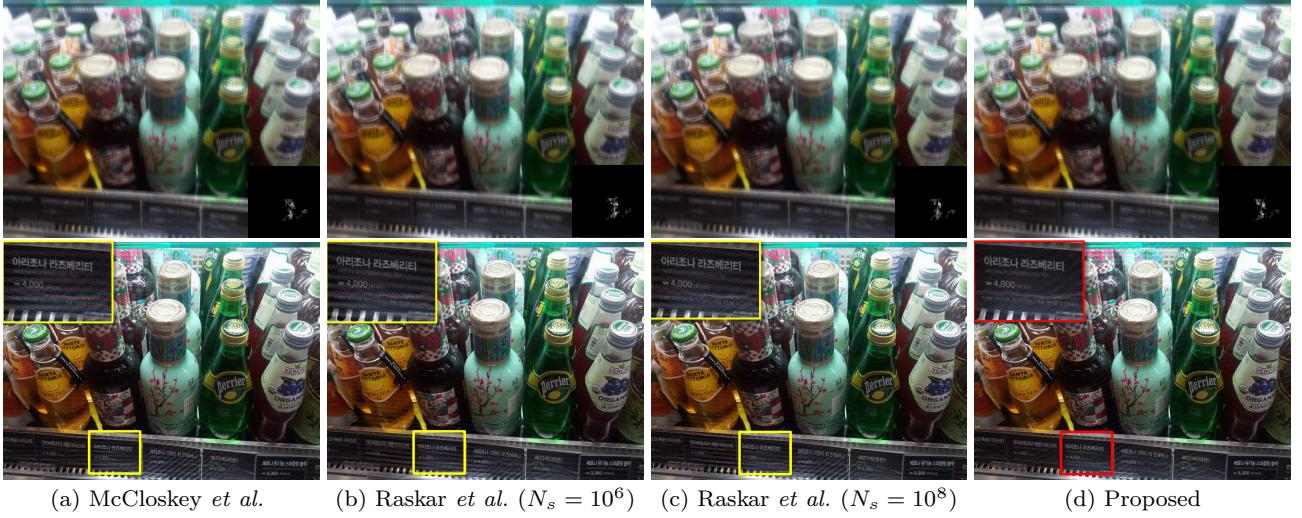
result using our pattern is sharper and clearer than the other results. To verify the deblurring results, we scanned the QR codes in the deblurred images using a QR code scanner from a mobile phone, Google Goggles. The scanner successfully recognized the code from our results, while it failed to recognize the codes from other results due to the remaining blur and noise. In Fig. 16, we captured an outdoor scene from a moving car. By using the fluttering patterns generated by our method, the contents of the scene become legible after deblurring, which otherwise would be very difficult to read.

Next, we examined the visual quality of the motion deblurring. We captured a sharp image and synthesized blur kernels, using the fluttering patterns of the proposed method, and others. We synthesized randomly

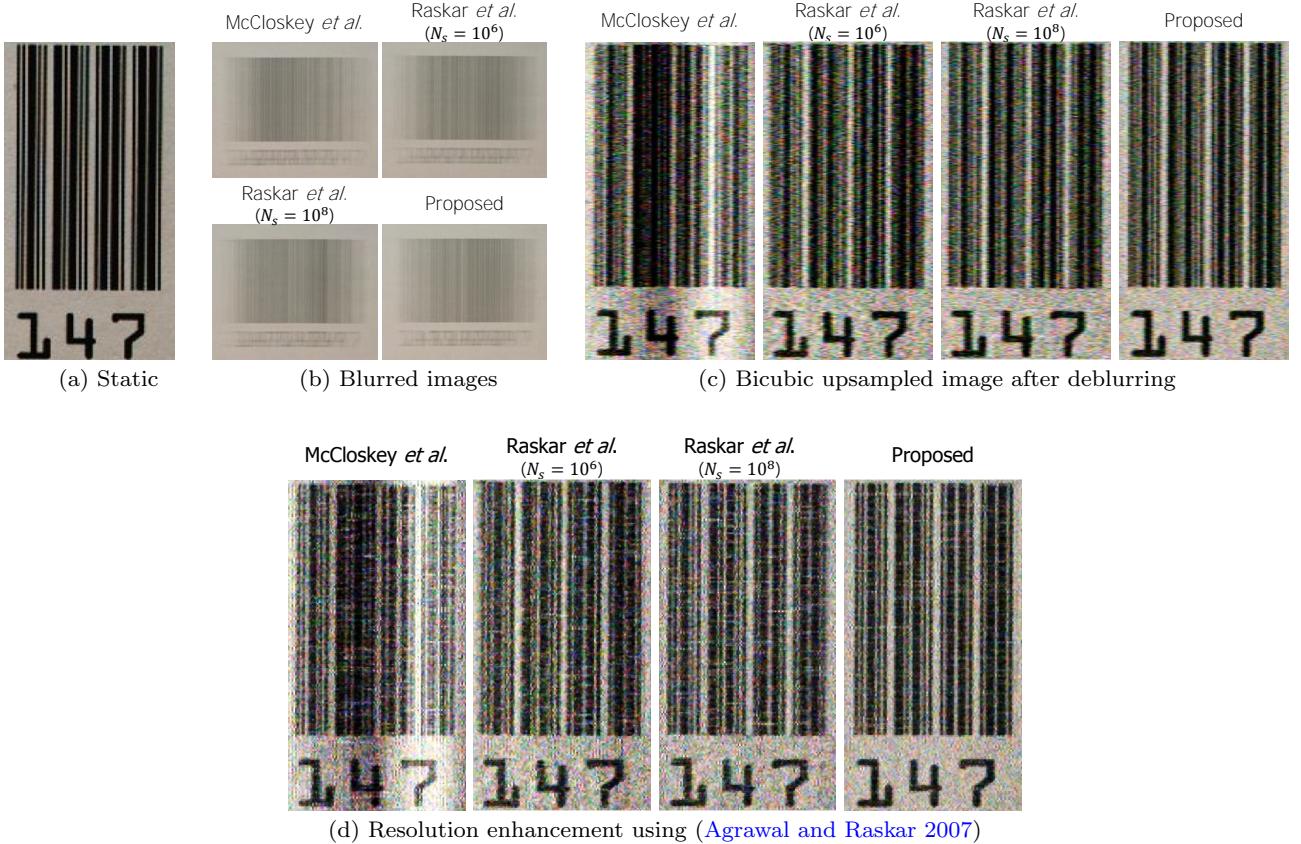
(a) Raskar  $N_s = 10^6$ 

(b) Proposed

**Fig. 16** Our coded exposure imaging in action. Fluttering patterns of length 60 are used.



**Fig. 17** An example of motion deblurring with the fluttering patterns of length 70.

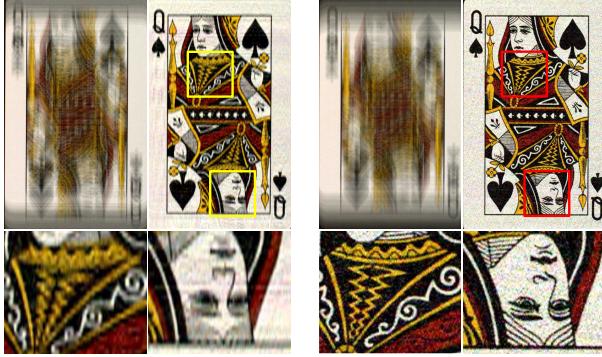


**Fig. 18** Comparison of the resolution enhancement ( $\times 2$ ) performance. (a) Static image of a barcode. (b) Captured images with different fluttering patterns of length 120. (c) Bicubic upsampled images by two after deblurring. (d) Resolution enhanced images using motion blur. In (b, c), the results with the proposed sequence are clearer than results with the other sequences.

generated blur kernels using the fluttering patterns. The blur kernels were estimated by using a blind deconvolution algorithm (Shan et al. 2008). In Fig. 17, the deblurred image captured from the proposed method

appears sharper and clearer than the results from the other methods.

The work in (Agrawal and Raskar 2007) showed that coded exposure imaging is not only effective for motion deblurring but also for resolution enhancement.



**Fig. 19** Comparison of the deblurring performance with different sequence lengths under the same exposure. (a) sequence length = 40, 1 chop duration = 3 ms. (b) sequence length = 120, 1 chop duration = 1 ms.

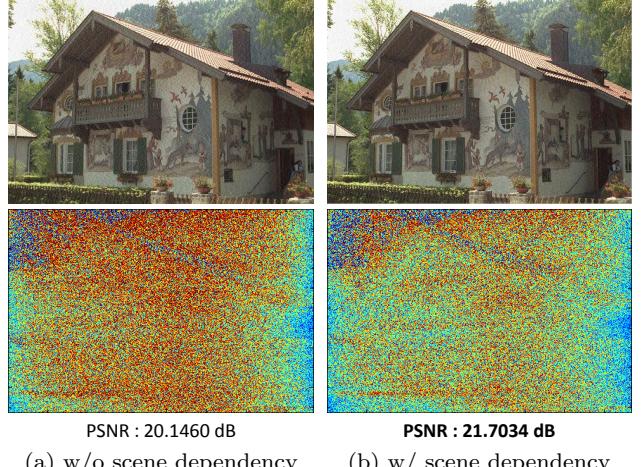
In their analysis, the optimal code length was approximately  $k * s$  for a given enhancement factor  $s$  and a blur size  $k$ . Therefore, they emphasized the importance of a long binary sequence, as mentioned previously. To show its applicability, we implemented the resolution enhancement method in (Agrawal and Raskar 2007). Fig. 18 compares the performance using different binary sequences of length 120, and as expected, the sequence generated by our method provides better visual quality in both deblurring and resolution enhancement.

In Fig. 19, we compared deblurring performance using the same exposure times, but with different sequence lengths. Sequences of length 40 and 120 generated by the proposed method were used for this experiment, and we controlled the single chop time so that the exposure times were the same under different sequence lengths. As shown in Fig. 19, the deblurred image with the longer sequence preserves more spatial frequencies of the blurred image than the shorter sequence.

## 7 Discussions

In this paper, we have presented new methods for computing the fluttering sequence for coded exposure photography, by modifying the Legendre sequence and by using the memetic algorithm. We have also proposed a new cost function for generating the binary codes for coded exposure imaging, called the coded factor.

We validated the efficiency of our algorithms through various experiments, and were able to achieve better deblurring and resolution enhancement performance by using the binary codes generated with our algorithms. Through our experiment, we empirically confirmed that coded exposure imaging has about a 4.5 dB performance gain in terms of PSNR over conventional image deblurring with a hyper-Laplacian prior, for images



**Fig. 20** A comparison of deblurring results from Legendre sequences optimized from flipping operation (a) without and (b) with a scene dependent prior: (top row) deblurring images. (bottom row) error map.

captured under the same illumination condition and exposure time.

There is still room for improvement, and one intuitive way to further enhance performance is to combine the strengths of the proposed algorithms. Initializing the memetic algorithm with a set of Legendre sequences and random sequences is expected to achieve better results. This strategy will find optimal solutions while maintaining performance at least comparable to that of the Legendre sequences.

Another way would be to incorporate additional knowledge in the imaging condition. In this work, we have focused on the high-quality fluttering pattern generated at each length without considering any photographic conditions, such as the illumination or the objects velocity, etc. However, an optimal fluttering pattern can only be generated when all the information in a scene and camera are known in advance. For example, a long sequence is beneficial for higher SNR in low light in ideal conditions, but the actual performance improvement will be unpredictable if the sensor dependent parameters, such as quantum efficiency, the size of pixels and camera readout noise, are not considered.

In addition, it was demonstrated in (McCloskey 2010) that the open chop duration for optimal fluttering patterns should depend on the objects velocity. Recent coded exposure work in (Jeon et al. 2015) determined that utilizing complementary sets of fluttering patterns can alleviate the issue of object velocity. It was also shown in (Agrawal and Raskar 2009) that optimal patterns do not need to have an equal number of zeros and ones if signal-dependent noise is taken into account. As an important part of our future work, we plan to take

those factors into consideration and implement additional hardware to solve those issues.

Another interesting direction for future work is to develop scene dependent fluttering patterns. As an example, we optimized a Legendre sequence with length 140 using the flipping operation with a scene dependent prior, which imposed a flatness constraint in the middle spectral bands. As shown in Fig. 20, we were able to improve the deblurring result using the spectral prior of the scene. We also expect that our method can be utilized for per-frame coded exposure imaging (Jeon *et al.* 2015) as an initial sequence set. Lastly, we would like to explore applying the proposed algorithms to other coded systems such as microscopy, coded aperture, programmable aperture, and image multiplexing (Ma *et al.* 2015; Zuo *et al.* 2016; Nagahara *et al.* 2010; Asif *et al.* 2015).

**Acknowledgements** This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (No.2010-0028680). Hae-Gon Jeon was partially supported by Global PH.D Fellowship Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NRF-20151034617).

## Appendix

### Appendix 1: Derivation of Eq. (7)

Let  $\hat{U}$  be a fluttering shutter pattern with the elements  $\hat{u}_i \in \{0, 1\}$ . We denote  $B$  as  $B = \hat{U} - \mu$  with elements  $b_i \in \{-\mu, 1 - \mu\}$ , where  $\mu$  is the mean value of the elements in  $\hat{U}$ . Then we introduce  $U = 2(B + \mu - 0.5)$ , where  $u_i \in \{-1, 1\}$ . The difference between  $\hat{U}$  and  $U$  is that the sequence values have changed from  $\{0, 1\}$  to  $\{-1, 1\}$ .

Let  $\hat{a}_k$  be the autocovariance of  $\hat{U}$  and  $t_k$  be the autocorrelation of  $B$ , then  $\hat{a}_k = t_k$  (Boufounos 2007). We

denote  $a_k$  as the autocorrelation of  $U$ , which is derived as follows.

$$\begin{aligned}
 a_k &= \sum_{i=0}^{n-k-1} u_i u_{i+k} \\
 &= 4 \sum_{i=0}^{n-k-1} (b_i + m)(b_{i+u} + m) \quad (\text{where } m = \mu - 0.5) \\
 &= 4 \sum_{i=0}^{n-k-1} (b_i b_{i+k} + mb_i + mb_{i+k} + m^2) \\
 &= 4 \sum_{i=0}^{n-k-1} b_i b_{i+k} + 4 \sum_{i=0}^{n-k-1} (m^2 + mb_i + mb_{i+k}) \\
 &= 4 \sum_{i=0}^{n-k-1} b_i b_{i+k} + 4 \left( \sum_{i=0}^{n-k-1} m^2 + m \sum_{i=0}^{n-k-1} (b_i + b_{i+k}) \right) \\
 &= 4t_k + 4((n - k)m^2 + m \sum_{i=0}^{n-k-1} (b_i + b_{i+k})) \\
 &= 4\hat{a}_k + 4m \sum_{i=0}^{n-k-1} (\hat{u}_i + \hat{u}_{i+k} + 3\mu - 0.5) \\
 &\quad (\text{where } \sum_{i=0}^{n-k-1} b_i b_{i+k} = \hat{a}_k).
 \end{aligned}$$

As mentioned in the original paper,  $m$  becomes 0 with the assumption that the sequence is balanced with equal number of zeros and ones for optimal autocorrelation properties.

## References

- Agrawal, A., & Raskar, R. (2007) Resolving objects at higher resolution from a single motion-blurred image. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [2](#), [3](#), [10](#), [14](#), [15](#)
- Agrawal, A., & Raskar, R. (2009) Optimal single image capture for motion deblurring. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [15](#)
- Agrawal, A., & Xu, Y. (2009) Coded exposure deblurring: Optimized codes for psf estimation and invertibility. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [2](#), [3](#)
- Asif, M. S., Ayremlou, A., Sankaranarayanan, A., Veeraraghavan, A., & Baraniuk, R. (2015) Flatcam: Thin, bare-sensor cameras using coded aperture and computation. arXiv preprint arXiv:150900116 [16](#)
- Baden, J. M. (2011) Efficient optimization of the merit factor of long binary sequences. IEEE Transactions on Information Theory 57(12), 8084–8094 [2](#), [5](#)

## Appendix 2: Example Sequences

Sequences Length	Sequence (Hex)
30	16A3809B
40	6ED109A838
50	02952CB8A2301F
60	0769257A06018B8E
70	2661230C25F85A3554
80	260186CCE097C62AAD2B
90	00733FCFA45A75E02AF092B2
100	0FB14E90234D6C91D5B787A3B9
110	03490F849D2CCFE500CCB9D73F55
120	BE7192BABA931CFA742D801B42E5F3
130	0073E9D0B6006D0B97CE3257752639F4E
140	0FC807BC9964B787C2CA919522842A33AF39
150	011CC82A74E8D5F663BE69680E8FE9698239 18
160	8C098D191405CBC2E24B3F98EF2F802E596A AF34
170	0166AF63826461899071BD5BACA0B7E1BB0F DA0A6BB5
180	011E755D200EE87C2C789E85219089979468E D49BBED96
190	12CBFD0221072F92F972F0E2660C6192B02D8 9A2A6B11DC5
200	7302CA07604A05BC6937C5BB2A2AC848B5A ED76A600E3F170E

**Table 2** Table of the proposed fluttering patterns.

Borwein, P., Choi, K. K., & Jedwab, J. (2004) Binary sequences with merit factor greater than 6.34. *IEEE Transactions on Information Theory* 50(12), 3234–3249 [5](#)

Borwein, P., Kaltofen, E., & Mossinghoff, M. J. (2007) Irreducible polynomials and barker sequences. *ACM Communications in Computation Algebra* 41(4), 118–121 [2](#)

Boufounos, P. (2007) Generating binary processes with all-pole spectra. In: *Proceeding of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* [16](#)

Chen, X., Ong, Y.-S., Lim, M.-H., & Tan, K. C. (2011) A multi-facet survey on memetic computation. *IEEE Transactions on Evolutionary Computation* 15(5), 591–607 [6](#)

Cossairt, O., Gupta, M., & Nayar, S. K. (2013) When does computational imaging improve performance? *IEEE Transactions on Image Processing (TIP)* 22(2), 447–458 [1](#)

Fergus, R., Singh, B., Hertzmann, A., Roweis, S. T., & Freeman, W. T. (2006) Removing camera shake from a single photograph. *ACM Transactions on Graphics* 25(3), 787–794 [2](#)

Franzen, R. (1999) Kodak lossless true color image suite. URL <http://www.r0k.us/graphics/kodak/> [8](#)

Gallardo, J. E., Cotta, C., & Fernández, A. J. (2009) Finding low autocorrelation binary sequences with memetic algorithms. *Applied Soft Computing* 9(4), 1252–1262 [2, 6](#)

Golay, M. (1983) The merit factor of legendre sequences (corresp.). *IEEE Transactions on Information Theory* 29, 934–936 [4](#)

Golay, M. J. (1977) Sieves for low autocorrelation binary sequences. *IEEE Transactions on Information Theory* 23(1), 43–51 [7](#)

- Gorthi, S. S., Schaak, D., & Schonbrun, E. (2013) Fluorescence imaging of flowing cells using a temporally coded excitation. *Optics Express* 21(4), 5164–5170 [2](#)
- Høholdt, T., & Jensen, H. E. (1988) Determination of the merit factor of legendre sequences. *IEEE Transactions on Information Theory* 34(1), 161–164 [5](#)
- Jedwab, J. (2005) A survey of the merit factor problem for binary sequences. In: *Proceeding of Sequences and Their Applications* [2](#)
- Jensen, J. M., Jensen, H. E., & Høholdt, T. (1991) The merit factor of binary sequences related to difference sets. *IEEE Transactions on Information Theory* 37(3), 617–626 [4](#)
- Jeon, H.-G., Lee, J.-Y., Han, Y., Kim, S. J., & Kweon, I. S. (2013) Fluttering pattern generation using modified legendre sequence for coded exposure imaging. In: *Proceedings of International Conference on Computer Vision (ICCV)* [3](#)
- Jeon, H.-G., Lee, J.-Y., Han, Y., Kim, S. J., & Kweon, I. S. (2015) Complementary sets of shutter sequences for motion deblurring. In: *Proceedings of International Conference on Computer Vision (ICCV)* [15, 16](#)
- Krishnan, D., & Fergus, R. (2009) Fast image deconvolution using hyper-laplacian. In: *Advances in Neural Information Processing Systems (NIPS)* [8, 12](#)
- Lempel, A., Cohn, M., & Eastman, W. (1977) A class of balanced binary sequences with optimal autocorrelation properties. *IEEE Transactions on Information Theory* 23(1), 38–42 [4](#)
- Levin, A., Fergus, R., Durand, F., & Freeman, W. T. (2007) Image and depth from a conventional camera with a coded aperture. *ACM Transactions on Graphics* 26(3) [2](#)
- Levin, A., Weiss, Y., Durand, F., & Freeman, W. (2009) Understanding and evaluating blind deconvolution algorithms. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp 1964–1971 [2](#)
- Lucy, L. B. (1974) An iterative technique for the rectification of observed distributions. *Astronomical Journal* 79, 745–754 [2](#)
- Ma, C., Liu, Z., Tian, L., Dai, Q., & Waller, L. (2015) Motion deblurring with temporally coded illumination in an led array microscope. *Optics Letters* 40(10), 2281–2284 [16](#)
- Martin, D., Fowlkes, C., Tal, D., & Malik, J. (2001) A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *Proceedings of International Conference on Computer Vision (ICCV)* [10](#)
- McCloskey, S. (2010) Velocity-dependent shutter sequences for motion deblurring. In: *Proceedings of European Conference on Computer Vision (ECCV)* [2, 3, 10, 15](#)
- McCloskey, S., Ding, Y., & Yu, J. (2012) Design and estimation of coded exposure point spread functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 34(10), 2071–2077 [2, 3, 6, 8, 9, 10](#)
- Mertens, S. (1996) Exhaustive search for low-autocorrelation binary sequences. *Journal of Physics A* 29, 473–481 [2, 7](#)
- Michalewicz, Z. (1996) Genetic algorithms+ data structures= evolution programs. *springer* [7](#)
- Militzer, B., Zamparelli, M., & Beule, D. (1998) Evolutionary search for low autocorrelated binary sequences. *IEEE Trans Evolutionary Computation* 2(1), 34–39 [7](#)
- Nagahara, H., Zhou, C., Watanabe, T., Ishiguro, H., & Nayar, S. K. (2010) Programmable aperture camera using lcos. In: *Proceedings of European Conference on Computer Vision (ECCV)* [16](#)
- No, J.-S., Lee, H.-K., Chung, H., Song, H.-Y., & Yang, K. (1996) Trace representation of legendre sequences of mersenne prime period. *IEEE Transactions on Information*

- Theory 42(6), 2254–2255 4
- Park, K., Shin, S., Jeon, H.-G., Lee, J.-Y., & Kweon, I. S. (2014) Motion deblurring using coded exposure for a wheeled mobile robot. In: The 11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI) 10
- Raskar, R., Agrawal, A., & Tumblin, J. (2006) Coded exposure photography: motion deblurring using fluttered shutter. ACM Transactions on Graphics 25(3), 795–804 1, 2, 3, 4, 6, 8, 9
- Richardson, W. H. (1972) Bayesian-based iterative method of image restoration. Journal of the Optical Society of America (JOSA) 62, 55–59 2
- Schechner, Y. Y., Nayar, S. K., & Belhumeur, P. N. (2007) Multiplexing for optimal lighting. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 29(8), 1339–1354 8
- Shan, Q., Jia, J., & Agarwala, A. (2008) High-quality motion deblurring from a single image. ACM Transactions on Graphics 27(3), 73:1–73:10 2, 14
- Tai, Y.-W., Kong, N., Lin, S., & Shin, S. Y. (2010) Coded exposure imaging for projective motion deblurring. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2
- Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004) Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing (TIP) 13(4), 600–612 8
- Wiener, N. (1964) Extrapolation, Interpolation, and Smoothing of Stationary Time Series. The MIT Press 2
- Xiong, T., & Hall, J. I. (2011) Modifications of modified jacobi sequences. IEEE Transactions on Information Theory 57(1), 493–504 2
- Zhou, C., Lin, S., & Nayar, S. (2011) Coded aperture pairs for depth from defocus and defocus deblurring. International Journal on Computer Vision (IJCV) 93(1), 53 3
- Zuo, C., Sun, J., Feng, S., Zhang, M., & Chen, Q. (2016) Programmable aperture microscopy: A computational method for multi-modal phase contrast and light field imaging. Optics and Lasers in Engineering 80, 24–31 16