Optimal Coded Sampling for Temporal Super-Resolution

Amit Agrawal[†], Mohit Gupta[‡], Ashok Veeraraghavan[†] and Srinivasa G. Narasimhan[‡]
[†]Mitsubishi Electric Research Labs (MERL), Cambridge, MA 02139
[‡]Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213
http://www.amitkagrawal.com

Abstract

Conventional low frame rate cameras result in blur and/or aliasing in images while capturing fast dynamic events. Multiple low speed cameras have been used previously with staggered sampling to increase the temporal resolution. However, previous approaches are inefficient: they either use small integration time for each camera which does not provide light benefit, or use large integration time in a way that requires solving a big ill-posed linear system.

We propose coded sampling that address these issues: using N cameras it allows N times temporal super-resolution while allowing $\sim \frac{N}{2}$ times more light compared to an equivalent high speed camera. In addition, it results in a well-posed linear system which can be solved independently for each frame, avoiding reconstruction artifacts and significantly reducing the computational time and memory. Our proposed sampling uses optimal multiplexing code considering additive Gaussian noise to achieve the maximum possible SNR in the recovered video. We show how to implement coded sampling on off-the-shelf machine vision cameras. We also propose a new class of invertible codes that allow continuous blur in captured frames, leading to an easier hardware implementation.

1. Introduction

A video camera has limited temporal resolution which is determined by the frame rate and exposure time of the camera. Temporal events occurring faster than the frame rate of a camera leads to aliasing in captured image sequence and blur in individual frames due to the camera's finite integration time. This blur could be because of motion of objects, in which case it is referred to as *motion blur*. However, temporal change of intensities can also happen when there is no motion in the scene, e.g., a flickering light or LCD screen. The goal of temporal super-resolution (SR) is to produce an aliasing-free video, where for each frame the effective integration time is small enough to avoid blur. Thus, temporal SR is more general than motion deblurring. In fact, motion blur artifacts may be removed by temporal SR [22].

A high speed camera has a fundamental light capture limit: if the frame rate is f frames/sec, the exposure duration cannot be greater than $1/f$ sec. In addition, commercial high speed cameras are expensive, require large bandwidth and are limited to capture durations (few seconds) that can fit in local memory. Multiple cameras have been used to increase the temporal resolution by staggering the start of integration across the frame time. Using N cameras each running at frame rate f, a video with an effective frame rate of Nf can be recovered by staggering the start of each camera's exposure window by $\frac{1}{Nf}$ and interleaving the captured frames in chronological order [28, 27]. However, the exposure time is set to $\frac{1}{Nf}$, similar to an equivalent high speed camera and thus this scheme is light-inefficient. We refer to this as point sampling and later show that it corresponds to an *identity* sampling matrix. The advantage here is that reconstruction process simply involves interleaving the captured frames and does not have any reconstruction artifacts.

Shechtman *et al*. [21, 22] combined several low frame rate videos to obtain a high frame rate output using an optimization framework. Their approach allow finite integration time to collect more light, which leads to motion blur in captured videos. However, the finite integration time of the camera acts as a low pass box filter and suppress high temporal frequencies. Recovering the lost high frequency information is inherently an ill-posed problem. Shechtman *et al*. [22] use regularization to solve the resulting ill-posed linear system to suppress the ringing artifacts. Moreover, using N cameras, they found that it is difficult to achieve a temporal SR by a factor of N. In addition, the reconstruction requires solving a huge sparse linear system (million variables) for modest video size of $256 \times 256 \times 16$.

We propose coded sampling that is optimal in the sense of maximizing the signal to noise ratio (SNR) of recovered high speed video assuming additive Gaussian noise in measurements. In our scheme, each low speed camera captures a different linear combination of frames of the desired high speed video. The linear combination is made invertible by employing a sampling strategy based on S-matrices [20, 9] and Hadamard multiplexing. Our approach overcomes the

978-1-4244-6985-7/10/$26.00 ©2010 IEEE

disadvantages of previous approaches: each camera can gather $N/2$ times more light compared to an equivalent high-speed camera, and the reconstruction process is well-posed, invertible and maximizes the output SNR. We show later that the our sampling matrix is block diagonal, where each block is identical and correspond to a $N \times N$ S-matrix. Thus, the corresponding frames from each camera can be processed *independently* to recover N output frames, leading to low computational time and memory requirements. Our scheme does not require any regularization or image priors and allows N times temporal SR using N cameras with minimal possible reconstruction noise.

The optimal coded sampling typically leads to discontinuous or coded blur in captured frames and requires specific triggering for implementation. While certain machine vision cameras may not support such triggering mechanism, they frequently support simple external trigger mode with continuous integration time. We propose invertible codes allowing continuous integration time that could be implemented on such cameras.

Contributions: Our paper makes the following contributions:

- We formulate the problem of temporal SR from multiple low-frame videos as a sampling problem.
- We show that the optimal sampling is achieved via coded sampling by taking invertible linear combinations of time samples.
- We demonstrate how to implement coded sampling to achieve N times temporal SR by using N coded exposure cameras. We also propose a new class of invertible codes that allow continuous integration time for easier implementation.

1.1. Benefits and limitations

Our approach allows more light capture compared to [28] as well as avoids noise/reconstruction artifacts compared to [21]. It leads to a well-posed linear system of size N *independent* of the number of frames processed and support streaming output due to independent processing of frames. We use CCD cameras with global shutter that avoid rolling shutter artifacts present in typical CMOS cameras [28]. Our implementation shares some of the limitations with [28, 21], since we also use non co-located multiple cameras. We assume that the scene is either relatively planar or is far away from the camera so that the images can be aligned using projective transforms similar to [28, 21]. Non-linearities in the imaging system such as specularities, saturation, non-linear camera response and radiometric calibration errors lead to artifacts in the reconstructed high speed video. Geometric calibration errors lead to spatial jitter (wobbling artifacts) in reconstructed frames.

1.2. Related work

Multiplexed sampling has been used for increasing the capture SNR in acquiring images under variable illumination [20]. This was extended in [19] to include the effect of sensor noise and saturation. Our approach is similar, but along the temporal dimension. Multiplexing angular information by reducing spatial resolution has been used for lightfield capture using lenslets [15] and masks [25]. Pupil-plane multiplexing to capture wavelength and polarization information by reducing spatial resolution has been proposed in [10]. Assorted pixels [14] perform a point-sampling of multi-dimensional data and use learned prior models for reconstruction.

Motion deblurring: Recent interest in computational photography has spurred significant research in motion deblurring algorithms. Fergus *et al*. [8] use natural image statistics to estimate the point spread function (PSF) from a single blurred image. Joshi *et al*. [12] estimate non-parametric, spatially-varying blur functions by predicting the sharp version of a blurry input image. Recent work on deblurring algorithms [29, 23, 4] have shown promising results on motion blurred images. A coded exposure [18] camera makes motion PSF invertible so that the resulting deconvolution process becomes well-posed. Agrawal and Raskar [2] analyzed capture methods for single image motion deblurring using the similar criterion of maximizing the SNR of the deblurred output. Note that temporal SR is more general than motion deblurring and can reduce motion blur artifacts *without* any PSF estimation.

Camera arrays: Levoy and Hanrahan [13] presented one of the earliest systems for capturing scenes from multiple perspectives for static scenes. This was extended to dynamic scenes by Dayton Taylor [24] using a linear array of still cameras. Wilburn *et al*. [28, 27] used camera arrays for temporal SR as well as effects such as digital refocusing. Similar to [21], we show an array of 2×2 cameras for 4X temporal SR.

Super-resolution: Combining multiple low-resolution images to increase the spatial resolution is well-known [11, 6, 16]. A hardware solution using sub-pixel detector shifts was shown in [7]. Baker and Kanade [5] analyzed limits on achievable super-resolution factors. In [1], super-resolution and deblurring were performed simultaneously using a coded exposure camera. Shechtman *et al*. [21] combine space-time super-resolution in a common framework by formulating the low frame rate videos as *low-pass filtered* samples of high resolution space-time videos. They also propose combining still images with video. Our approach analyzes the most general sampling by imaging the low frame rate video as *coded* samples of high frame rate video. Similar to [21], spatial SR can also be incorporated in our approach, but we focus on temporal SR only.

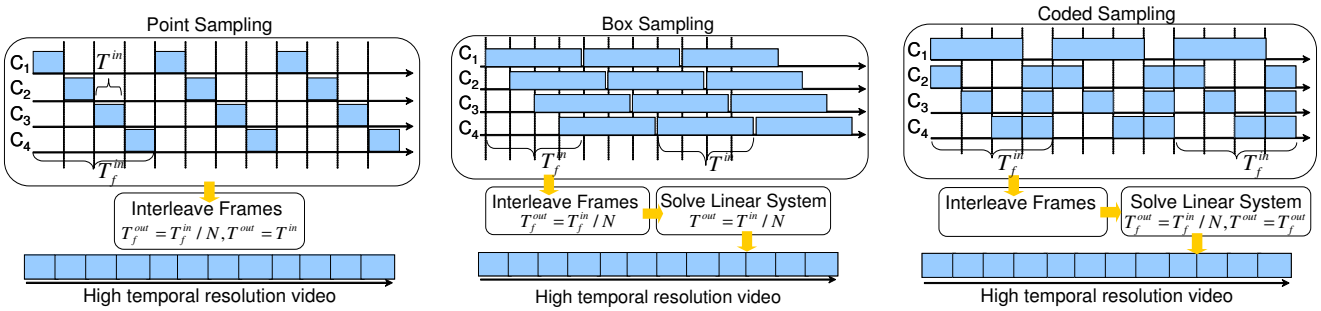

Figure 1. Comparison of sampling techniques using $N = 4$. The frame time T_f^{in} of each camera C_i is same. Point sampling captures independent samples across time with $T^{out} = T^{in} = T_f^{in}/N$. Box sampling collect more light ($T^{in} = T_f^{in}$) but captures low pass filtered samples, making it ill-posed. Coded sampling captures invertible linear combination of samples over time. Interleaving reduces frame time $T_f^{out} = T_f^{in}/N$ only for point and box sampling. Box and coded sampling requires solving a linear system to reduce the effective integration time T^{out}.

2. Temporal aliasing and blur

Temporal aliasing and motion blur are related but distinct visual effects in low frame rate videos. Let T_f be the *frame time* of the camera (inverse of the frame rate) and let $T \leq T_f$ be the integration time of each frame. T_f determines how *fast* the camera samples the temporal variations at each pixel, while T determines how *long* the camera integrates at that sampling rate. Depending on the relationship between T, T_f and the Nyquist sampling rate, one can either have blur, aliasing, a combination of both or none in the captured video. Aliasing occurs when the sampling rate is smaller than the Nyquist sampling rate, and it can occur along with blur if integration time T is large. A high speed camera avoids both blur and aliasing by sampling faster and keeping the integration time T sufficiently small. Note that since T cannot be greater than T_f for a camera, a high speed camera has a fundamental light capture limitation.

To achieve temporal SR, it is important to consider both (a) increase in frame rate or decrease in frame time T_f, and (b) decrease in integration time T. Either one is not sufficient enough. For example, one can always decrease the integration time T of a single camera to avoid motion blur, but since the frame rate is not increased, it will result in aliasing. On the other hand, consider interleaving frames from N cameras having $T = T_f$, by evenly spacing the start of the integration time across the frame time (Figure 1 (middle)). The interleaved video will automatically have higher frame time, since the temporal events are sampled faster. One can avoid aliasing artifacts in such an interleaved video, but due to large integration time, temporal blur will remain. Thus, the goal of temporal SR is to both remove aliasing *and* reduce blur in the reconstructed video frames.

2.1. Removing aliasing by interleaving frames

Figure 1 (left) shows the simplest way to achieve temporal SR, which we refer to as point sampling. By interleaving the start of integration, one can use N cameras each with a frame time T_f^{in} and integration time $T^{in} = T_f^{in}/N$ to remove aliasing, and obtain a video with frame time $T_f^{out} = T_f^{in}/N$. Note that $T^{out} = T^{in}$ and blur is avoided by keeping T^{in} small. This implementation is the one proposed in [28]. Note that point sampling does not have light advantage compared to an equivalent high speed camera. However, no extra processing is required and output does not have reconstruction noise or artifacts.

Specifically, consider N co-located cameras each with same frame time T_f^{in}. Let $T_s^i(k)$ and $T_e^i(k)$ denote the start and end of integration of camera i for frame k. Let the first camera ($i = 0$) starts integrating the first frame ($k = 0$) at $T_s^0(0) = 0$. If all cameras start integration at the same time for each frame, then $T_s^i(k) = kT_f^{in}$. Let $v_i(x, y, k)$ denote the i^{th} camera video. The *interleaved video* $u(x, y, k)$ is defined as the video obtained by temporally interleaving the corresponding frames from all cameras

$$u(x, y, k) = v_a(x, y, b), \quad b = \left\lfloor \frac{k}{N} \right\rfloor, a = k - Nb. \quad (1)$$

If the start of integration is interleaved *uniformly* according to

$$T_s^i(k) = kT_f^{in} + iT_f^{in}/N, \quad (2)$$

then the interleaved video has a smaller frame time of $T_f^{out} = T_f^{in}/N$ (higher frame rate). This is because the interleaved video frames correspond to samples at the intervals of T_f^{out}. Note that for an interleaved video, the integration time can be *larger* than the frame time, not possible for a conventional camera.

2.2. Light efficiency: box and coded sampling

Now consider the box sampling strategy (Figure 1 (middle)), which allow more light but introduces motion blur in the captured frames. In [22], a general framework with different start and integration time of input videos were proposed. But in essence their technique is similar to box sampling shown in Figure 1 (middle). Since the start of integration is interleaved, the interleaved video has higher frame rate, but needs to be processed to remove blur (to achieve effective lower integration time). However, the blur is caused

Sampling	Captured Video		Interleaved Video		Reconstruction	Linear System	FIS	Light Benefit	Computation
	Blur	Aliasing	Blur	Aliasing					
Point	No	Yes	No	No	Not Required	Well-posed	Yes	No	None
Box	Yes	Yes	Yes	No	Solve Linear System	Ill-posed	No	Yes	Depends on K
Coded	Yes	Yes	Yes	Yes	Solve Linear System	Well-posed	Yes	Yes	Constant

Figure 2. Comparison of sampling techniques for achieving N times temporal SR using N cameras. Box sampling requires solving a $NK \times NK$ ill-posed linear system for K frames, while coded sampling allows independent linear systems of size $N \times N$.

by a continuous box filter in each camera which suppress high temporal frequencies. The processing thus involves solving an ill-posed linear system.

Our proposed coded sampling is shown in Figure 1 (right). We also allow motion blur in captured frames, but in the most general form of coded blur. In each camera for each frame, the shutter is open and closed according to a code, resulting in discontinuous or coded blur. The code is chosen so as to preserve high temporal frequencies in the captured images and leads to a well-posed linear system. More importantly, the linear system can be solved independently for each set of captured frames. A key distinction with box sampling is that the start of integration is not interleaved exactly, and thus the interleaved video could have aliasing artifacts.

Frame independent sampling (FIS): Let $T_s(k) = \min_i T_s^i(k)$ and $T_e(k) = \max_i T_e^i(k)$. $T_s(k)$ and $T_e(k)$ denote the bounds of integration time of the corresponding frames of all cameras. We call a sampling strategy *frame independent sampling* (FIS) if $T_s(k+1) \geq T_e(k)$ for all k. Thus for FIS, the temporal information in the corresponding camera frames is not shared across frames and reconstruction can be done independently for the set of corresponding camera frames. Figure 1 shows that point and coded sampling are FIS, but box sampling is not. We later show that FIS results in block diagonal sampling matrices, while frame dependent sampling (FDS) does not. Figure 2 shows an in-depth comparison between the sampling schemes.

3. Sampling matrices and linear system

The above sampling techniques can be described in terms of a linear system governed by a *sampling matrix*, which describes the relationship between the N input videos and output video. Previous approaches are equivalent to either an identity sampling matrix as in [28], or an ill-posed sampling matrix which is not block diagonalizable [22]. Coded sampling results in an invertible block diagonal sampling matrix.

For co-located cameras, each pixel is independent and so we drop the spatial coordinates for ease of discussion[1]. Let $\mathbf{s}$ denote the intensity vector of a pixel in the output video at integration time T^{out}. Let $\mathbf{u}$ denote the interleaved vector for the pixel, obtained by stacking corresponding pixels from each camera according to (1). The sampling matrix A relates the interleaved low resolution video and the desired high resolution video as

$$\mathbf{u} = A\mathbf{s}. \tag{3}$$

For point sampling, it is easy to see that matrix A is an identity matrix of size $N \times N$ for every N interleaved frames. For $N = 4$,

$$\mathbf{u}(k) = \begin{bmatrix} v_1(k) \\ v_2(k) \\ v_3(k) \\ v_4(k) \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{s}(k). \tag{4}$$

This is because each camera samples the high resolution video at a distinct time instant. Each 1 or 0 of the sampling matrix corresponds to a sample in the output video at the integration time T^{out}. If we take K video frames from each camera, the resulting A matrix correspond to an identity matrix $I_{NK \times NK}$, which is trivially block diagonalized by $I_{N \times N}$.

For box sampling, the sampling matrix $(N = 4)$ corresponds to

$$\mathbf{u} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ & & & & & & & & \\ 0 & 0 & 0 & 0 & \ddots & \ddots & \ddots & \ddots & 0 \end{bmatrix} \mathbf{s}. \tag{5}$$

Note that the sampling matrix does not have independent blocks of size $N \times N$.

3.1. Optimal sampling

The optimal sampling is the one which minimizes the mean square error (MSE) in estimating the output $\mathbf{s}$ from captured interleaved video $\mathbf{u}$. Assuming IID zero mean Gaussian noise with variance σ^2 in $\mathbf{u}$, the maximum-likelihood (ML) estimate of output, $\widehat{\mathbf{s}}$ is given by

$$\widehat{\mathbf{s}} = (A^T A)^{-1} A^T \mathbf{u}. \tag{6}$$

Thus, the covariance matrix Σ of the error $\mathbf{s} - \widehat{\mathbf{s}}$ in the estimate is given by [17]

$$\Sigma = \sigma^2 (A^T A)^{-1} A^T A (A^T A) = \sigma^2 (A^T A)^{-1}. \tag{7}$$

The MSE increases by a factor $F = \mathtt{trace}(A^T A)^{-1}/n$, where n is the size of $\mathbf{u}$. A similar problem was studied by Schechner and Nayar [20] for capturing images under multiplexed illumination. The matrix A which minimizes the above MSE is called the S-matrix [20]. If $N + 1$ is

[1] In practice, one needs to geometrically align the images.

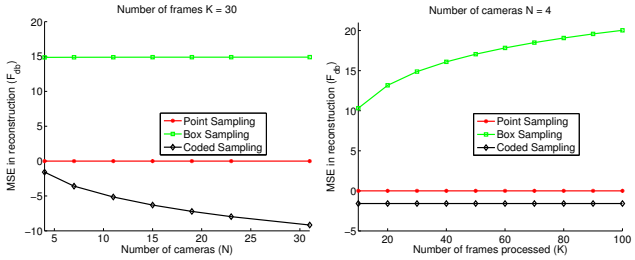

Figure 3. Reconstruction MSE F in dB. (Left) As N increases, coded sampling offers higher SNR than point and box sampling. (Right) Coded and point sampling MSE's are independent of number of frames processed due to being FIS. For box sampling, MSE increase with K.

divisible by 4, then the rows of the S-matrix correspond to Hadamard codes of length $N+1$. For S-matrix, the increase in noise $F = \frac{4N}{(N+1)^2}$, which is *less* than 1, indicating a multiplex advantage [20].

S-matrices have following properties. Firstly, each value is either 0 or 1. This implies that each row of the S-matrix correspond to the on/off sequence of a coded exposure camera [18]. Note that each bit of the code corresponds to a sample in the output video. Thus, each bit amounts to integration time of T^{out}. A 1 implies that the shutter is kept transparent and 0 implies that the shutter is kept opaque for the duration T^{out} within the integration time of the camera. Secondly, each row has $(N+1)/2$ ones implying that each camera integrates $(N+1)/2$ times more light compared to an equivalent high speed camera. Finally, inverting S-matrix is easy as shown in [20].

Code search: Note that S-matrices are not defined for all N. For small N, one can search for all possible binary matrices and choose the one with the lowest F. In order to enforce at least 50% light throughput, each row should have at least $N/2$ ones. For $N = 4$, we search for all 2^{16} choices (took 10 seconds in Matlab) and choose the optimal coding matrix C which minimizes F, given by

$$C = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \quad (8)$$

For K frames, the sampling matrix $A_{4K\times 4K} = \texttt{kron}(I_{K\times K}, C)$, where kron denotes the kronecker product. The corresponding sampling is visualized in Figure 1 (right). For large N, one can perform a randomized greedy search similar to the search for best motion deblurring code in [18]. Typically, it requires few minutes in Matlab to search 10^6 codes.

Reconstruction noise: Figure 3 plots $10\log_{10} F$ for the three sampling techniques to depict *increase* in noise (dB) assuming $K = 30$ input frames from N cameras. Note that F_{dB} is 0 for point sampling and less than 0 for coded sampling indicating SNR gain. As N increases, the reconstruction noise for box sampling does not decrease. Thus, box sampling is inherently ill-posed. For coded sampling, more cameras allow more temporal SR with even lower MSE. Note that even at $N = 4$, coded sampling is better by 15 dB than box sampling. Another interesting observation is that MSE increases for box sampling as number of frames K increase. But since coded and point sampling are FIS, MSE is *independent* of K for them.

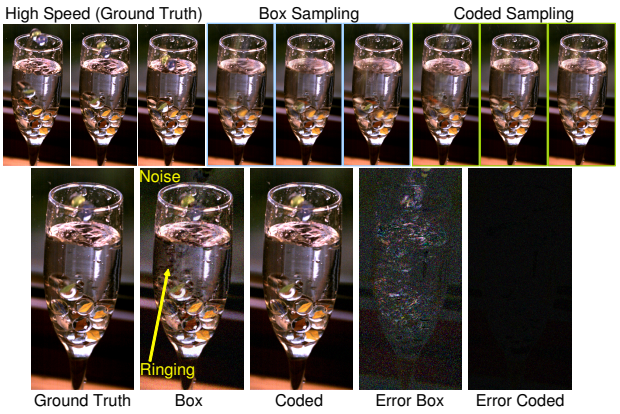

Figure 4. 15 cameras were simulated using a high-speed video for coded and box sampling. (Top) Three frames from high-speed, box and coded sampling videos. (Bottom) One of the reconstructed frame and corresponding error images. Notice the ringing artifacts and enhanced noise in box sampling reconstruction.

Figure 4 shows a simulation using a 500 fps high speed video of marbles falling into water. $N = 15$ low frame rate videos were simulated both for box and coded sampling and Gaussian noise ($\sigma = 0.1$) was added in frames. The top row shows three frames from the ground truth, box-sampling and coded-sampling videos respectively. Bottom row shows one of the reconstructed frames along with the error images. Notice the enhanced noise and ringing artifacts in the box reconstruction. The coded reconstruction gives an artifact and noise free output. The increase in MSE (F) was 32 times (15 dB) smaller for coded sampling compared to box sampling. Please see the supplementary materials for full videos.

3.2. Invertible codes with continuous blur

In general, optimal coded sampling leads to discontinuous blur in captured frames. However, this requires each camera to start and stop integration multiple times *within* the exposure time according to its code. While this feature is available in some machine vision cameras (Pointgrey Dragonfly2 [3]), several machine vision cameras do not support it. Such cameras often support external triggering followed by a *continuous* integration time. This implies that while the start of integration time can be changed, only codes that have continuous ones can be supported. Can we have invertible codes that allow continuous (box) blur?

We show that one can obtain such sampling with an increase in MSE compared to optimal sampling. A trivial continuous blur invertible code matrix for $N = 4$ is given by

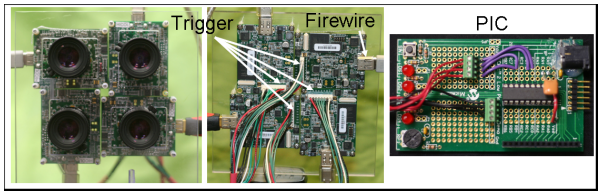

Figure 5. Our prototype using four cameras. A micro-controller (PIC) is used to trigger the cameras accurately.

$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$. We refer it to as *triangular* codes since the code matrix is a lower triangular matrix. The reconstruction noise here is larger than optimal sampling by 4 dB only for $N = 4$. However, triangular codes require a large dynamic range, since the exposure time between cameras changes by a factor of N. Ideally, we would like all codes to integrate similar amount of total light to avoid dynamic range issues. To achieve that, we search for continuous ones codes each having at least 50% light throughput.

Search space: For each camera, the code can have $\frac{N}{2}$, $\frac{N}{2}+1$, ..., N ones which can occur in $\frac{N}{2}+1$, $\frac{N}{2}$, ..., 1 places respectively. Therefore, the possible code choices for a single camera are $c = \frac{(N+2)(N+4)}{8}$. The total search space is thus $\binom{c}{N}$. For $N = 4$, the code matrix with minimum MSE[2] was found to be

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}. \quad (9)$$

Note that each row has continuous ones and thus would lead to box blur, but overall the linear system is well-posed. For $N = 4$, these codes are better than box sampling by 10 dB. These codes can also be thought of as traditional cameras with varying exposure and start times. While Shechtman *et al*. [22] also allows cameras with varying exposure and start time, their resulting system is not well-posed since the exposure and start times are not carefully chosen. The proposed codes here do not require regularization for reconstruction and lead to a frame-independent sampling in contrast to [22].

4. Implementation and results

Figure 5 shows our implementation using four Pointgrey Dragonfly2 cameras, each equipped with 12 mm computar lens. The cameras are arranged to keep their optical centers as close as possible and are kept ≈ 2 m away from the scene. Similar to [28], we assume that the scene is planar and perform geometric calibration using a checkerboard. We capture RAW images at resolution of 700×700 to reduce bandwidth and perform Bayer interpolation using Pointgrey SDK. After geometric calibration, the common field of view is spanned by 400×400 pixels. Color calibration is done using a Macbeth chart by computing a 3×3 color transformation for each camera. Dragonfly2 cameras support coded exposure via trigger mode 5[3]. All cameras are triggered using a Microchip PIC16F690 micro-controller which avoids temporal synchronization issues. We found that this was more stable than using a PC's parallel port [3], which could have trigger variations of 1 ms. We use $T_f^{in} = 60$ seconds, capturing the input videos at 16Hz. The reconstructed video has frame rate of 64Hz using 4 cameras, with frame integration time $T^{out} = 15$ seconds. As described in [21], the output video will be similar to the one captured with a camera having T^{out} integration time. Thus, if the scene is changing faster than 64Hz, the output video frames will also have blur. Please see videos in supplementary materials.

[2]There could be multiple solutions with same minimum MSE.

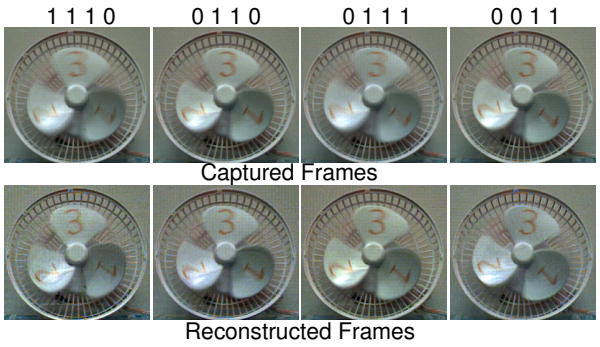

Figure 6. Captured and reconstructed frames for rotating fan using codes with continuous ones (9). In comparison with Figure 7, the reconstruction has more noise than coded sampling, but is sharper than box sampling.

Rotating fan: Figure 7 shows comparison of coded and box sampling for a fan rotating clockwise. Notice the coded blur in frames for coded sampling. The reconstructed fan blades are much sharper and closer to ground truth in coded reconstruction. The box reconstruction shows noise and ringing artifacts without regularization. By using regularization similar to one proposed in [21], noise can be reduced but blur cannot be removed completely. Thus, it is difficult to achieve N times SR with N cameras using box sampling, as also discussed in [21]. The reconstruction using coded sampling was obtained without any regularization. Similar videos were captured using continuous ones codes (9) by changing the trigger mechanism using PIC. Figure 6 shows corresponding captured and reconstructed frames. The reconstruction has more noise than coded sampling, but is sharper than box sampling.

Oscillating color chart: Figure 8 shows results on a scene where a Macbeth color chart was moved back and forth by hand. Note the enhanced noise and color/ringing artifacts in box reconstruction if no regularization is used. This shows that box sampling is inherently ill-posed. The

[3]Pointgrey requires delay between frames in trigger mode 5, which leads to an equivalent gap in reconstructed video after every N frames. It could be avoided by using external shutter [18].

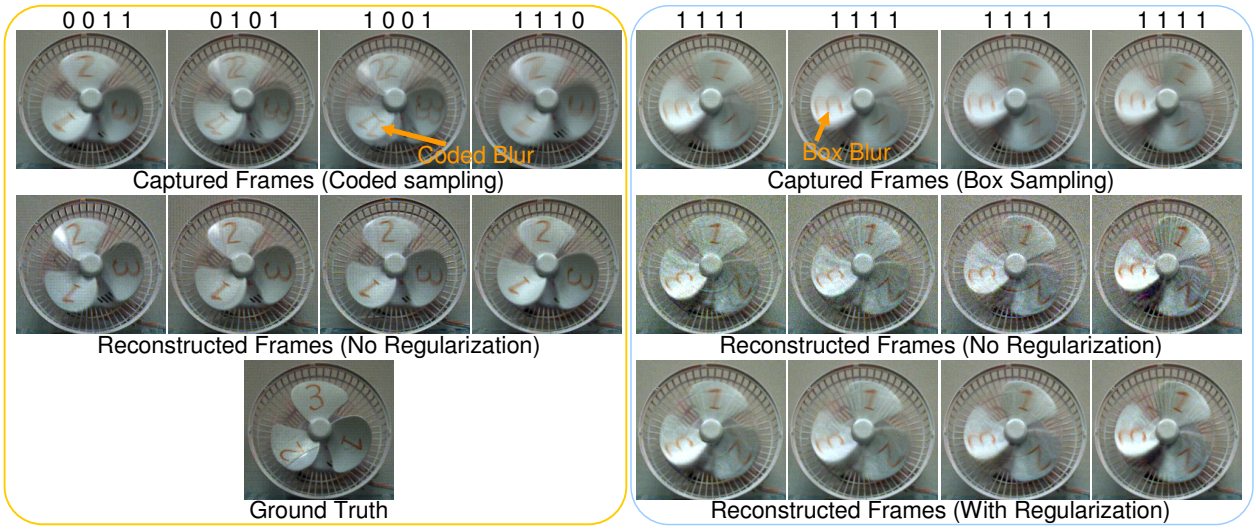

Figure 7. Comparison of box and coded sampling for a fan rotating clockwise. Coded sampling provides sharper reconstruction without any regularization compared to box sampling which has more noise and reconstruction artifacts.

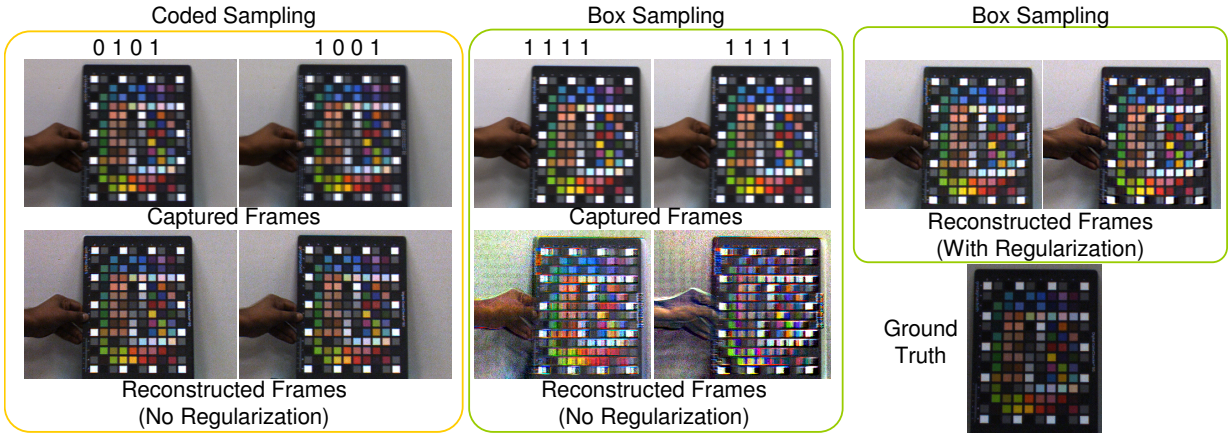

Figure 8. Moving Color Chart. Without regularization, box reconstruction has enhanced noise, color and ringing artifacts. Regularization can suppress noise at the expense of more blur (less temporal SR). In contrast, coded reconstruction produce sharp color edges on the color chart without using any regularization.

reconstruction using coded sampling was obtained without any regularization.

Facial expressions: Figure 9 shows a person making fast facial expressions. Notice the 'double' teethes in captured frames corresponding to camera with code 1001. The reconstructed frames show reduced blur without any motion estimation.

5. Discussions

Coded sampling promises exciting avenues for computational photography and vision research beyond motion deblurring [18]. The ability to capture more light and have immediate streaming reconstruction without reconstruction artifacts for temporal SR is a big benefit. This could be useful for medical imaging such as laryngoscopy and endoscopy, where reconstruction artifacts are undesirable. Combining coded sampling in time with coded aperture techniques [25] can lead to a unified treatment of motion and focus blur, which are generally handled separately. Utilizing image priors and domain knowledge can allow greater than N super-resolution factors with N cameras. We use the same code for each camera across frames, but due to frame-independent sampling, the codes can be dynamically modified as in [26]. Similar to [19], intensity dependent Poisson noise and saturation effects can also be incorporated for better codes. While our implementation uses four cameras, it is easily scalable by using external triggering based on proposed codes. By using *same* temporal modulation for few cameras (out of N), spatial SR can be achieved at the cost of temporal SR.

Conclusions: We formulated temporal SR using multiple low frame rate videos as a sampling problem and analyzed its motion blur and aliasing aspects. We showed that optimal sampling for temporal SR involves taking invertible

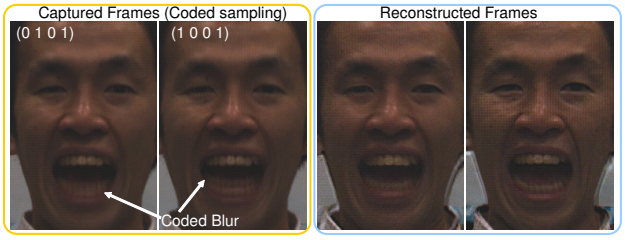

Figure 9. Facial expressions. (Left) Two frames from captured video corresponding to camera C_2 and C_3 with codes 0101 and 1001. (Right) Frames from reconstructed video. Blur in captured frames is removed via temporal SR without any motion estimation.

linear combination of frames, which can be implemented using multiple coded exposure cameras. Our proposed sampling captures more light compared to an equivalent high speed camera, and results in a well-posed linear system which can be solved independently for frames. Thus, it overcomes the limitations of previous approaches in terms of light capture, reconstruction noise, and computational requirements. We also proposed a new class of invertible codes that lead to an easier implementation on most machine vision cameras.

Acknowledgements We thank the anonymous reviewers for their suggestions. We thank Jay Thornton, Keisuke Kojima, John Barnwell, and Haruhisa Okuda, Mitsubishi Electric, Japan, for help and support. Narasimhan and Gupta also acknowledge support from the Okawa Foundation, ONR Grant N00014-08-1-0330 and NSF CAREER Award IIS-0643628.

References

[1] A. Agrawal and R. Raskar. Resolving objects at higher resolution from a single motion-blurred image. In *CVPR*, June 2007.

[2] A. Agrawal and R. Raskar. Optimal single image capture for motion deblurring. In *CVPR*, June 2009.

[3] A. Agrawal and Y. Xu. Coded exposure deblurring: Optimized codes for psf estimation and invertibility. In *CVPR*, June 2009.

[4] A. Agrawal, Y. Xu, and R. Raskar. Invertible motion blur in video. *ACM Trans. Graph.*, 28(3), 2009.

[5] S. Baker and T. Kanade. Limits on super-resolution and how to break them. *IEEE Trans. Pattern Anal. Machine Intell.*, 24:1167–1183, Sept. 2002.

[6] B. Bascle, A. Blake, and A. Zisserman. Motion deblurring and super-resolution from an image sequence. In *ECCV*, volume 2, pages 573–582, 1996.

[7] M. Ben-Ezra, A. Zomet, and S. Nayar. Video super-resolution using controlled subpixel detector shifts. *IEEE Trans. Pattern Anal. Machine Intell.*, 27:977–987, June 2005.

[8] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Trans. Graph.*, 25(3):787–794, 2006.

[9] M. Harwit and N. J. A. Sloane. *Hadamard transform optics*. Academic Press, New York, 1979.

[10] R. Horstmeyer, G. Euliss, R. Athale, and M. Levoy. Flexible multimodal camera using a light field architecture. In *ICCP*, Apr. 2009.

[11] M. Irani and S. Peleg. Improving resolution by image registration. In *CVGIP*, volume 53, pages 231–239, 1991.

[12] N. Joshi, R. Szeliski, and D. Kriegman. PSF estimation using sharp edge prediction. In *CVPR*, June 2008.

[13] M. Levoy and P. Hanrahan. Light field rendering. In *SIGGRAPH 96*, pages 31–42, 1996.

[14] S. Narasimhan and S. Nayar. Enhancing Resolution along Multiple Imaging Dimensions using Assorted Pixels. *IEEE Trans. Pattern Anal. Machine Intell.*, 27(4):518–530, Apr 2005.

[15] R. Ng, M. Levoy, M. Brdif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a hand-held plenoptic camera. Technical report, Stanford Univ., 2005.

[16] A. Patti, M. Sezan, and A. Tekalp. Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time. *IEEE Trans. Image Processing*, 6:1064–1076, Aug. 1997.

[17] H. Poor. *An Introduction to Signal Detection and Estimation*. Springer-Verlag, 1988.

[18] R. Raskar, A. Agrawal, and J. Tumblin. Coded exposure photography: motion deblurring using fluttered shutter. *ACM Trans. Graph.*, 25(3):795–804, 2006.

[19] N. Ratner and Y. Y. Schechner. Illumination multiplexing within fundamental limits. In *CVPR*, June 2007.

[20] Y. Schechner, S. Nayar, and P. Belhumeur. A theory of multiplexed illumination. In *ICCV*, volume 2, pages 808–815, Oct. 2003.

[21] E. Shechtman, Y. Caspi, and M. Irani. Increasing space-time resolution in video. In *ECCV*, pages 753–768, 2002.

[22] E. Shechtman, Y. Caspi, and M. Irani. Space-time super-resolution. *IEEE Trans. Pattern Anal. Machine Intell.*, 27(4):531–545, Apr. 2005.

[23] Y. Tai, D. Hao, M. S. Brown, and S. Lin. Correction of spatially varying image and video motion blur using a hybrid camera. *IEEE Trans. Pattern Anal. Machine Intell.*, 2009.

[24] D. Taylor. Virtual camera movement: The way of the future? American Cinematographer 77, 1996.

[25] A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin. Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *ACM Trans. Graph.*, 26(3):69:1–69:12, July 2007.

[26] A. Veeraraghavan, D. Reddy, and R. Raskar. Coded strobing photography: Compressive sensing of high-speed periodic events. *to appear in PAMI*, 2010.

[27] B. Wilburn, N. Joshi, V. Vaish, M. Levoy, and M. Horowitz. High-speed videography using a dense camera array. In *CVPR*, volume 2, pages 294–301, June 2004.

[28] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy. High performance imaging using large camera arrays. *ACM Trans. Graph.*, 24(3):765–776, 2005.

[29] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Progressive inter-scale and intra-scale non-blind image deconvolution. *ACM Trans. Graph.*, 27(3):1–10, 2008.