

Coded Rolling Shutter Photography: Flexible Space-Time Sampling

Jinwei Gu*
Columbia University

Yasunobu Hitomi
Sony Corporation

Tomoo Mitsunaga
Sony Corporation

Shree Nayar
Columbia University

Abstract

We propose a novel readout architecture called coded rolling shutter for complementary metal-oxide semiconductor (CMOS) image sensors. Rolling shutter has traditionally been considered as a disadvantage to image quality since it often introduces skew artifact. In this paper, we show that by controlling the readout timing and the exposure length for each row, the row-wise exposure discrepancy in rolling shutter can be exploited to flexibly sample the 3D space-time volume of scene appearance, and can thus be advantageous for computational photography. The required controls can be readily implemented in standard CMOS sensors by altering the logic of the control unit.

We propose several coding schemes and applications: (1) coded readout allows us to better sample time dimension for high-speed photography and optical flow based applications; and (2) row-wise control enables capturing motion-blur free high dynamic range images from a single shot. While a prototype chip is currently in development, we demonstrate the benefits of coded rolling shutter via simulation using images of real scenes.

1. Introduction

CMOS image sensors are rapidly overtaking CCD sensors in a variety of imaging systems, from digital still and video cameras to mobile phone cameras to surveillance and web cameras. In order to maintain high fill-factor and readout speed, most CMOS image sensors are equipped with column-parallel readout circuits, which simultaneously read all pixels in a row into a line-memory. The readout proceeds row-by-row, sequentially from top to bottom. This is called *rolling shutter*. Rolling shutter has traditionally been considered detrimental to image quality, because pixels in different rows are exposed to light at different times, which often causes skew and other image artifacts, especially for moving objects [11, 13, 6].

From the perspective of sampling the space-time volume of a scene, however, we argue that the exposure discrepancy in rolling shutter can actually be exploited using computational photography to achieve new imaging functionalities

*This research was supported in part by Sony Corporation and the National Science Foundation (IIS-03-25867 and CCF-05-41259).

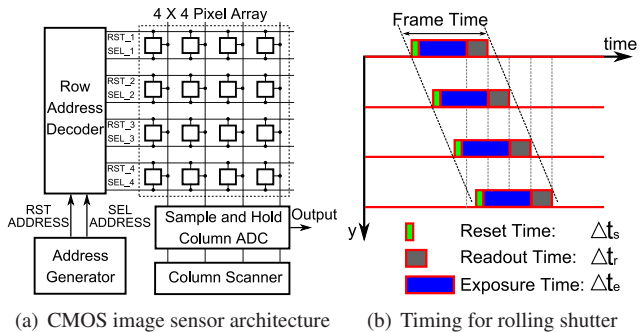


Figure 1. The address generator in CMOS image sensors is used to implement coded rolling shutter with desired row-reset and row-select patterns for flexible space-time sampling and features. In fact, a few recent studies have demonstrated the use of conventional rolling shutter for kinematics and object pose estimation [1, 2, 3].

In this paper, we propose a novel readout architecture for CMOS image sensors called *coded rolling shutter*. We show that by controlling the readout timing and exposure length for each row of the pixel array, we can flexibly sample the 3D space-time volume of a scene and take photographs that effectively encode temporal scene appearance within a single 2D image. These coded images are useful for many applications, such as skew compensation, high-speed photography, and high dynamic range imaging.

As shown in Fig. 1, the controls of row-wise readout and exposure can be readily implemented in standard CMOS image sensors by altering the logic of the address generator unit without any further hardware modification. For conventional rolling shutter, the address generator is simply a shift register which scans all the rows and generates row-reset (RST) and row-select (SEL) signals. For coded rolling shutter, new logics can be implemented to generate the desired RST and SEL signals for coded readout and exposure, as shown in Fig. 2. Since the address generator belongs to the control unit of CMOS image sensors [9, 17], it is easy to design and implement new logics in the address generator using high level tools.

We have begun the process of developing the prototype sensor. We expect to have a fully programmable coded rolling shutter sensor in 18 months. Meanwhile, in this paper, we demonstrated coding schemes and their applications

via simulations. The simulation experiments are performed with real images, *i.e.*, full space-time volumes of scene appearance recorded with high-speed cameras were used to synthesize the output images of a coded rolling shutter sensor. These synthesized images thus have similar characteristics as the images captured with a real sensor.

2. Rolling Shutter and Related Work

We first introduce some background related to rolling shutter. As shown in Fig. 1a, the exposure in CMOS image sensors is controlled by the row-reset and row-select signals sent from the row address decoder – each row becomes photosensitive after a row-reset signal, and stops collecting photons and starts reading out data after a row-select signal. Because there is only one row of readout circuits, the readout timings for different rows cannot overlap. In rolling shutter, as shown in Fig. 1b, the readout timings are shifted sequentially from top to bottom.

We denote the reset time, the readout time, and the exposure time for a row with Δt_s , Δt_r , and Δt_e , respectively. For typical CMOS sensors, Δt_s is around $1 \sim 5 \mu s$ and Δt_r is around $15 \sim 40 \mu s$. For an image sensor with M rows, we denote the reset timing (*i.e.*, the rising edge of row-reset signals) and the readout timing (*i.e.*, the falling edge of row-readout signals) for the y -th row ($1 \leq y \leq M$) with $t_s(y)$ and $t_r(y)$, respectively. For rolling shutter, we have $t_r(y) = y\Delta t_r$ and $t_s(y) = t_r(y) - \Delta t_e - \Delta t_r - \Delta t_s$.

Recent works have modeled the geometric distortion caused by rolling shutter [1, 11], and proposed methods to compensate for skew due to planar motion [13, 6]. Wilburn *et al.* [24] demonstrated the use of rolling shutter with a camera array for high-speed photography. Many components of CMOS image sensors have also been re-designed for specific applications, such as HDR imaging [25, 15] and multi-resolution readout [12]. These ideas are giving rise to a new breed of image sensors, referred to as “smart” CMOS sensors [20], which have spurred significant interest among camera manufactures.

3. Coded Rolling Shutter: An Overview

In coded rolling shutter, both $t_s(y)$ and $t_r(y)$ can be controlled by the address generator. As a result, the exposure time, $\Delta t_e(y)$, can also be varied for different rows. Let $E(x, y, t)$ denote the radiance of a scene point (x, y) at time t , and $S(x, y, t)$ denote the shutter function of a camera. The captured image $I(x, y)$ is

$$I(x, y) = \int_{-\infty}^{\infty} E(x, y, t) \cdot S(x, y, t) dt. \quad (1)$$

Figure 2 shows four types of shutter functions. For the global shutter (Fig. 2a) widely used in CCD image sensors, $S(x, y, t)$ is a 1D rectangular function. Raskar *et al.* [22] proposed the flutter shutter (Fig. 2b), which breaks the single integration time into multiple chunks and thus intro-

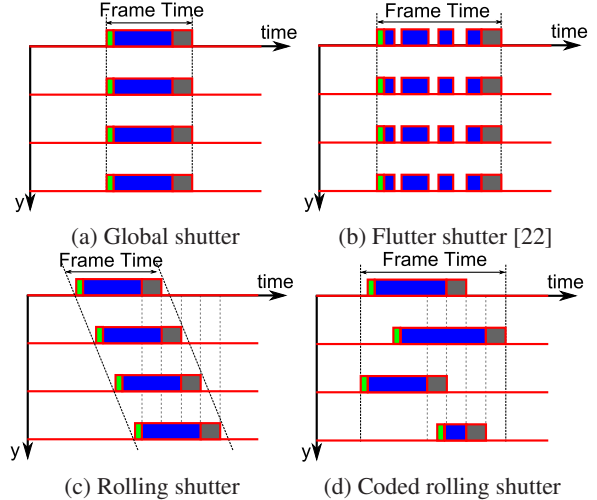


Figure 2. Timing charts for four types of camera shutter function.

duces high frequency components for motion deblur. Since the coding is fixed for all pixels, it is in effect a *coded global shutter*, which is also a 1D function of time t . For conventional rolling shutter (Fig. 2c), $S(x, y, t) = S(t - t_r(y)) = S(t - y\Delta t_r)$. It is still a 1D function because of the fixed sequential readout order.

In contrast, the proposed coded rolling shutter (Fig. 2d) extends the shutter function to 2D $S(y, t)$ in which both the readout timing $t_r(y)$ and the exposure time $\Delta t_e(y)$ can be row-specific. As mentioned earlier, because there is only one row of readout circuits, the readout timings for different rows cannot overlap, which imposes a constraint on $t_r(y)$. Specifically, for an image sensor with M rows, the total readout time for one frame is $M\Delta t_r$. Each valid readout timing scheme will correspond to a one-to-one assignment of the M readout timing slots to the M rows. This is a typical *assignment problem* in combinatorial optimization. Figure 3 shows one simple assignment, which is adopted in the conventional rolling shutter.

The remainder of the paper demonstrates several coding schemes and their applications. We limited the coding to be within one frame time.

1. Coded readout timing $t_r(y)$ for better sampling over the time dimension for optical flow and high-speed photography, as detailed in Sec. 4.
2. Coded exposure time $\Delta t_e(y)$ for high dynamic range (HDR) imaging. With this control, we propose a simple row-wise auto-exposure in Sec. 5.1, which is effective for outdoor, natural scenes. Moreover, if both $t_r(y)$ and $\Delta t_e(y)$ are controllable, we show in Sec. 5.2 that motion-blur free HDR images can be recovered from a single coded image.

4. Coded Readout and Its Applications

In this section, we show how to use coded readout to better sample the time dimension by shuffling the readout timings $t_r(y)$ among rows. We propose two coding schemes.

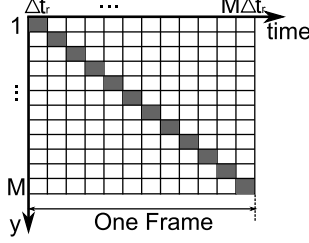


Figure 3. Coded rolling shutter is constrained as an assignment problem because readout timings cannot overlap between rows. Here we show the assignment of a conventional rolling shutter.

4.1. Interlaced Readout

Figure 4a shows the first scheme called *interlaced readout*, in which the total readout time for one frame is uniformly distributed into K sub-images ($K = 2$ in Fig. 4). Each sub-image has M/K rows while preserving full resolution in the horizontal (x) direction. This is similar to *interlacing* in video broadcast systems [5]. We note that interlaced readout is different from the skip-readout mode in CMOS image sensors [17] where only a *fixed* subset of rows are used for imaging — In contrast, interlaced readout uses all the rows and allows full-length exposure for each row. Specifically, for interlaced readout, the readout timing $t_r(y)$ for the y -th row is set as

$$t_r(y) = \left(\frac{M(y-1)}{K} - \left\lfloor \frac{y-1}{K} \right\rfloor \cdot (M-1) + 1 \right) \Delta t_r, \quad (2)$$

for an image sensor with M rows, where $\lfloor \cdot \rfloor$ is the floor function. Since the time lag between the top and the bottom row of each sub-image is $M\Delta t_r/K$, the skew in these sub-images is $1/K$ time of the conventional rolling shutter. Moreover, the time lag between two consecutive sub-images is also reduced to $M\Delta t_r/K$ (*i.e.*, the frame rate will increase K times.)

The sub-images can be used to estimate optical flow for frame interpolation and removing skew, as depicted in Fig. 4b. The gray and red circles represents the sampled points from the input coded image. First, we use cubic interpolation to resize the two sub-images I_1 and I_2 vertically to full resolution (shown as the gray and red solid lines) and then compute the optical flow \mathbf{u}_0 between them. Intermediate images within the blue parallelogram can be recovered via bidirectional interpolation [4]:

$$I_w(p) = (1-w)I_1(p - w\mathbf{u}_w(p)) + wI_2(p + (1-w)\mathbf{u}_w(p)), \quad (3)$$

where $0 \leq w \leq 1$, $p = (x, y)$ represents one pixel, $\mathbf{u}_w(p)$ is the forward-warped optical flow computed as $\mathbf{u}_w(p + w\mathbf{u}_0(p)) = \mathbf{u}_0(p)$. For example, the black dot-dash line in Fig. 4b shows the intermediate image $I_{w=0.5}$. Moreover, we can also interpolate a *skew-free* image, $I_{w=0.5}^{\text{skew-free}}$, shown as the blue dash line in Fig. 4b, by replacing the scalar w in Eq. (3) with a vector $w = 1 - (y-1)/(M-1)$.

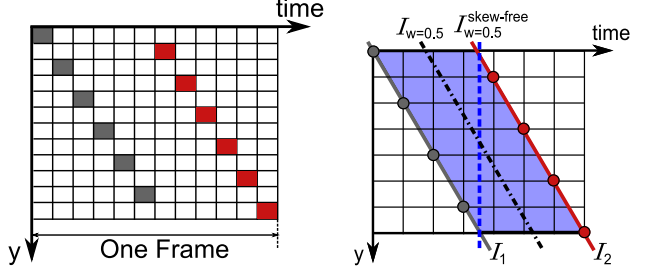


Figure 4. Diagrams of interlaced readout coding and interpolation.

Figure 5 shows an experimental result. The scene is recorded with a high-speed camera (Casio EX-F1) at 300fps with image resolution 512×384 . The recorded video is used to synthesize the image captured with conventional rolling shutter (Fig. 5a) and the coded image captured with the interlaced readout rolling shutter (Fig. 5b). Figures 5(c,d) show the two interpolated sub-images, I_1 and I_2 , where the skew is reduced by half compared to Fig. 5a. Figure 5e shows the computed optical flow \mathbf{u}_0 , which is used to interpolate the intermediate image $I_{w=0.5}$ (Fig. 5f) and the skew-free image $I_{w=0.5}^{\text{skew-free}}$ (Fig. 5g). Figures 5(h,i) show the errors between the two interpolated images and the true skew-free image, which confirms that $I_{w=0.5}^{\text{skew-free}}$ indeed removes almost all the skew and is close to the ground truth. The remaining error in $I_{w=0.5}^{\text{skew-free}}$ is caused by occlusions during the estimation of optical flow.

4.2. Staggered Readout

Figure 6a shows the second coding scheme, called *staggered readout*, which reverses the order of readout within every K neighboring rows ($K = 2$ in Fig. 6a). Similar to the previous scheme, K sub-images can be extracted from a single coded image where each sub-image has M/K rows. The readout timing $t_r(y)$ in this case is set as

$$t_r(y) = \left(\left(2 \left\lfloor \frac{y-1}{K} \right\rfloor + 1 \right) K - y + 1 \right) \Delta t_r. \quad (4)$$

Compared with the interlaced readout, there are two main differences: (1) The time lag within each sub-image for staggered readout is $(M - K + 1)\Delta t_r$. This is roughly the same as conventional rolling shutter ($M\Delta t_r$), which means the skew remains unchanged. (2) The time lag between two consecutive sub-images is Δt_r , which is on the order of $15 \sim 40 \mu s$. This is the main benefit of this coding—a simple way to achieve ultra-high speed photography for time-critical events such as a speeding bullet or a bursting balloon.

One example is shown in Fig. 6. The original clip is recorded with a Phantom v7.1 camera at 600 fps [23], which is used to synthesize the coded image of staggered readout ($K = 8$) shown in Fig. 6b. Three extracted sub-images are shown in Figs. 6(c,d,e), which capture the moment the foot

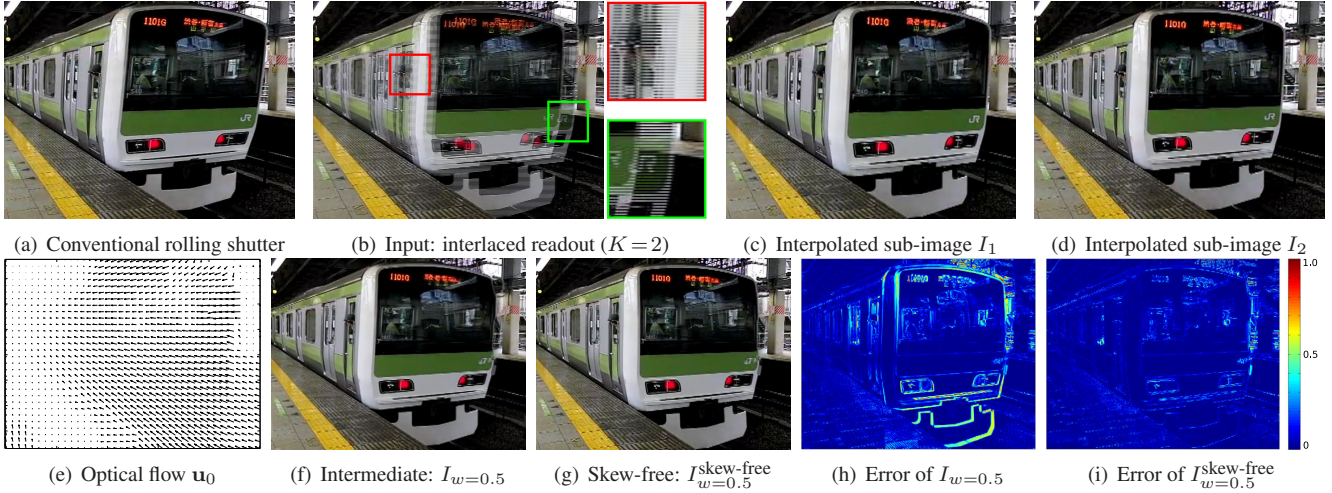


Figure 5. Results of optical flow based interpolation with interlaced readout.

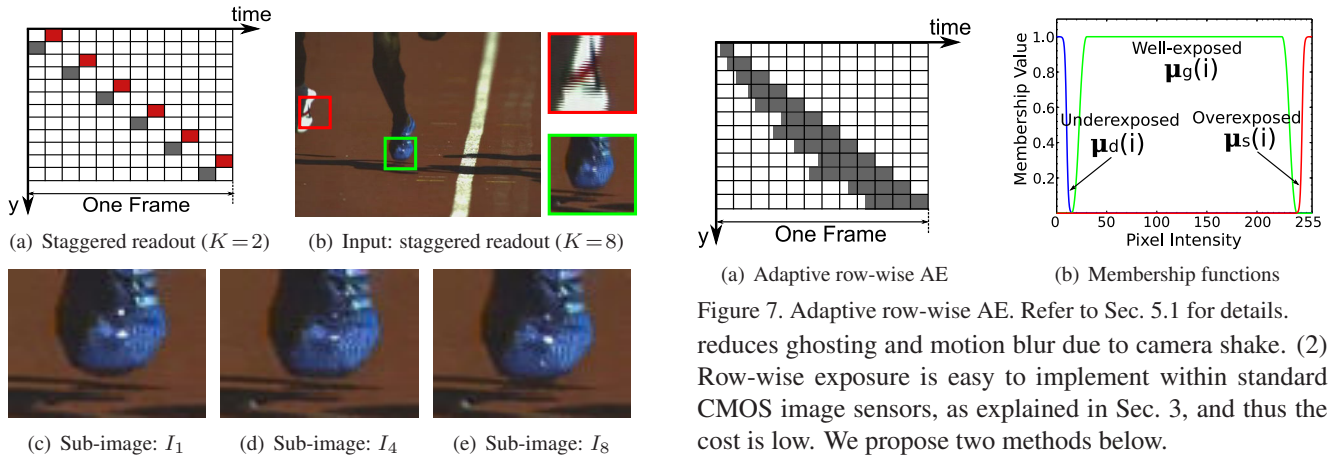


Figure 6. Staggered readout for high-speed photography.

touches the ground. This precise moment would *not* be captured using a conventional rolling shutter. More results can be found in the supplementary video.

5. Coded Exposure and Readout for High Dynamic Range (HDR) Imaging

HDR imaging typically requires either multiple images of a given scene taken with different exposures [8, 16], or special hardware supports [19, 18]. The first requires a static scene and static camera to avoid ghosting and motion blur, while the latter is expensive, thus making HDR imaging inconvenient for hand-held consumer cameras. Researchers have recently proposed methods to remove ghosting [10] and motion blur [27, 14] from multiple images.

In this section, we show that coded row-wise exposure $\Delta t_e(y)$ can be used to alleviate these problems for practical HDR imaging: (1) The dynamic range of scene radiance can be better captured by either adaptively setting the exposure per row or interlacing multiple exposures into a single image, which avoids taking multiple images and effectively

(a) Adaptive row-wise AE

(b) Membership functions

reduces ghosting and motion blur due to camera shake. (2) Row-wise exposure is easy to implement within standard CMOS image sensors, as explained in Sec. 3, and thus the cost is low. We propose two methods below.

5.1. Adaptive Row-wise Auto-Exposure

When we take a picture with auto-exposure (AE), the camera often will take two images—first it quickly captures a temporary image to gauge the amount of light and determine an optimal exposure, and then adjusts the exposure and takes a second image as the final output [17]. Most existing AE algorithms are designed to find a single exposure that is optimal for an entire image, which is highly limiting for many scenes.

In our first method, we implement a simple yet effective auto-exposure method called *adaptive row-wise auto-exposure*. As shown in Fig. 7a, the method finds an optimal exposure for each row of the pixel array, and then takes a second image where each row is adjusted for best capturing the scene radiance. The second image is normalized (*i.e.*, divided by the row-wise exposure) to generate the final output. Compared with conventional auto-exposure, row-wise auto-exposure is more flexible and effective, especially for scenes where the dynamic range is mainly spanned vertically (*e.g.*, outdoor scenes where the sky is much brighter than the ground).

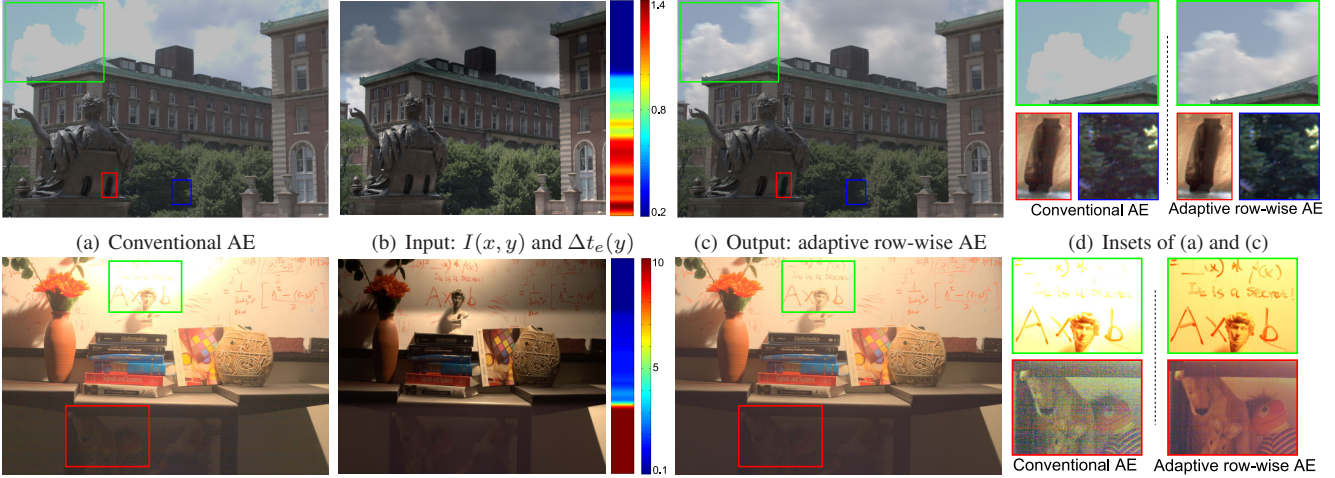


Figure 8. Results of adaptive row-wise auto-exposure and conventional auto-exposure.

To find the optimal exposure for each row, we propose a simple method using fuzzy logic. An optimal exposure for a given row should minimize the number of saturated and underexposed pixels within the row while keeping most pixels well-exposed. This heuristic is formulated as follows. As shown in Fig. 7b, we first introduce three membership functions, $\mu_s(i)$, $\mu_d(i)$, and $\mu_g(i)$ which describe the degree of being overexposed (*i.e.*, saturated), underexposed, or well-exposed for intensity i . Let I_0 denote the temporary image. It measures the scene radiance everywhere except in the saturated regions, where no information is recorded. We thus assume the scene radiance is $L = I_0(1 + s\mu_s(I_0))$, where $s \geq 0$ is a scale factor used to estimate the scene radiance in saturated regions. The smaller s is, the more conservative the AE algorithm will be. The optimal exposure $\Delta t_e(y)$ for the y -th row is found by maximizing the following functional:

$$\max_{\Delta t_l \leq \Delta t_e(y) \leq \Delta t_u} \sum_x \mu(L(x, y)\Delta t_e(y)), \quad (5)$$

where $\mu(i)$ is defined as

$$\mu(i) = \mu_s(i) + \lambda_d \mu_d(i) + \lambda_g \mu_g(i), \quad (6)$$

with weights λ_d , λ_g and lower and upper bounds of exposure adjustment Δt_l and Δt_u . In our experiments, $s = 4$, $\lambda_d = 0.2$, $\lambda_g = 0.05$, $\Delta t_l = 0.1$, $\Delta t_u = 10.0$, and the three membership functions are designed as $\mu_s(i) = \frac{1}{1 + e^{245-i}}$, $\mu_d(i) = \frac{1}{1 + e^{i-10}}$, and $\mu_g(i) = 1/(1 + |i - 128|^{60}/100)$. Once the optimal exposures are found for all rows,¹ they are used to capture the second image I . The final output image is computed as $I_r(x, y) = I(x, y)/\Delta t_e(y)$.

The experiments are performed as follows. For each scene, we use a Canon EOS 20D to take 30 images with exposures ranging from $\frac{1}{6400}$ to 1.5 seconds in the manual

¹This calculation can be done within a FPGA built in cameras.

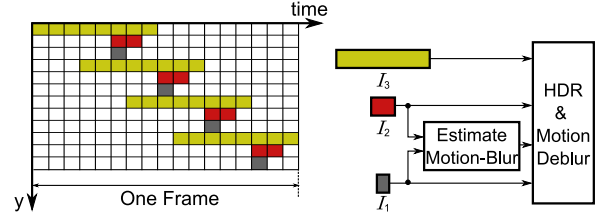


Figure 9. Staggered readout and multiple exposure coding for HDR imaging with hand-held cameras.

mode, as well as an image in the AE mode (denoted as I_0). To create the coded image I , for each row from the captured 30 images we choose the one whose exposure is the closest to the estimated optimal exposure. Figure 8 shows two sets of experimental results – Fig. 8a shows the images with conventional AE, Fig. 8b shows the coded images I and the row-wise exposures, and Fig. 8c shows the final outputs I_r . As shown in Fig. 8d, the adaptive row-wise AE produces higher quality photographs, in which the saturation (*e.g.*, the clouds and the text) as well as the noise in dark regions (*e.g.*, the statues and the toys) are significantly reduced.

This method requires almost no image processing. If further post-processing (*e.g.*, denoising) is needed, noise amplification along the vertical direction (which is known from the exposure patterns) can be considered. Moreover, for scenes where the dynamic range is predominantly spanned horizontally (*e.g.*, a dark room viewed from outside), this method reverts back to conventional auto-exposure.

5.2. Staggered Readout and Coded Exposure for HDR Imaging with Hand-held Cameras

The goal of the second method is to recover HDR from a single image for hand-held cameras. We show that with staggered readout (shown in Sec. 4.2) and row-wise exposure, not only can we code multiple exposures into one image, but we can also remove image blur due to camera shake by estimating planar camera motion.

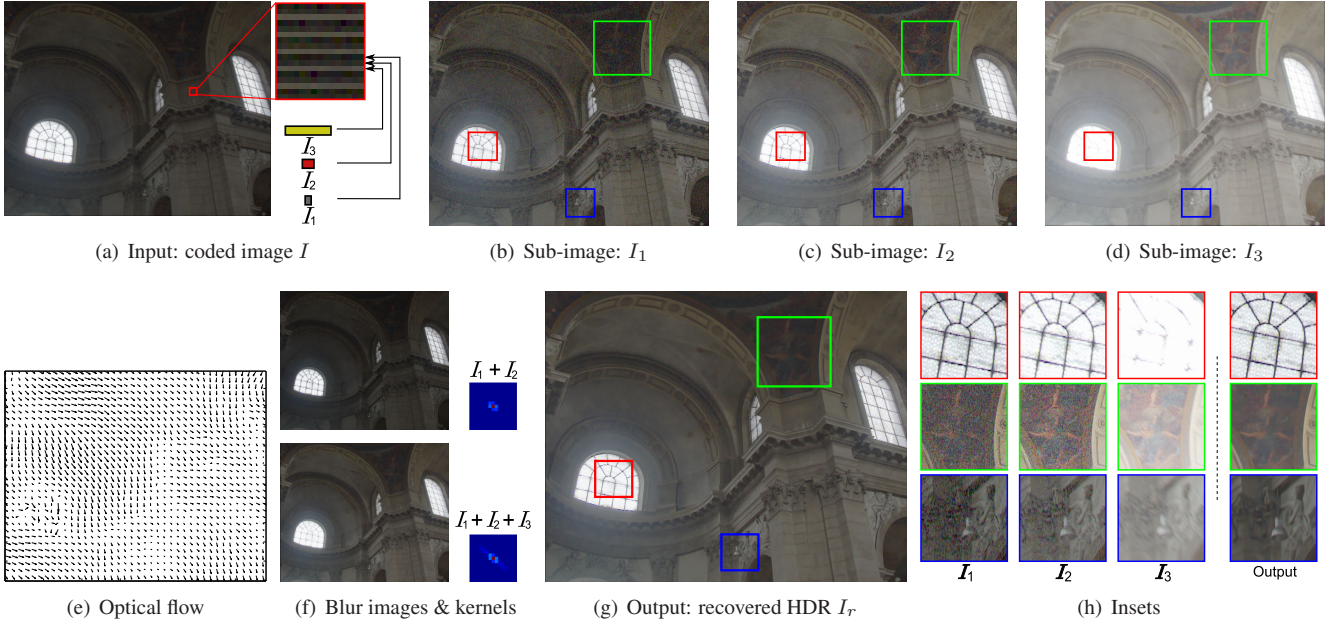


Figure 10. Results of staggered readout and coded exposure for HDR imaging for hand-held cameras.

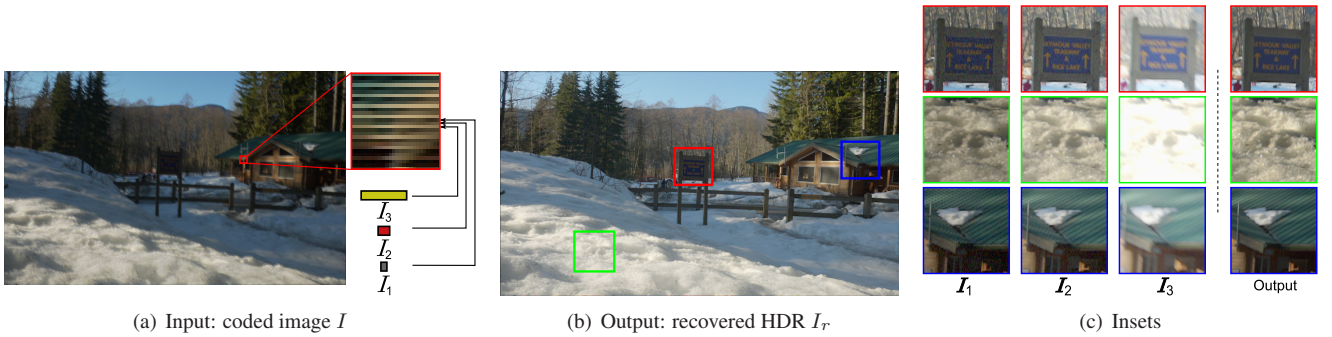


Figure 11. Another result for HDR imaging with coded rolling shutter for hand-held cameras.

As shown in Fig. 9, the pixel array of a CMOS image sensor is coded with staggered readout ($K = 3$) and three exposures, Δt_{e1} , Δt_{e2} , and Δt_{e3} . Thus, from a single input image, I , we can extract three sub-images, I_1 , I_2 , and I_3 . These sub-images are resized vertically to full resolution using cubic interpolation. For static scenes/cameras, these sub-images can be directly used to compose a HDR image. For hand-held cameras, however, camera shake is inevitable, especially for long exposures. Because the sub-images are captured with staggered readout, the time lag between them is small. We can thus assume camera motion as translation only with a fixed velocity between sub-images. The motion vector $\vec{u} = [u_x, u_y]$ can be estimated from I_1 and I_2 using optical flow:

$$\vec{u} = \text{mean}(\text{computeFlow}(I_1, I_2 - I_1)). \quad (7)$$

The computed flow is used to estimate the two blur kernels for I_2 and I_3 , respectively. Instead of deblurring I_2 and I_3 directly, we found that deblurring two composed images, $I_1 \oplus I_2$ and $I_1 \oplus I_2 \oplus I_3$, will effectively suppress the

ringing,² where the operator \oplus means the images are first center-aligned using the motion vector \vec{u} and then added together. We denote the two deblurred images as $I_{b1} = \text{deblur}(I_1 \oplus I_2, \vec{u}, \Delta t_{e1}, \Delta t_{e2})$, and $I_{b2} = \text{deblur}(I_1 \oplus I_2 \oplus I_3, \vec{u}, \Delta t_{e1}, \Delta t_{e2}, \Delta t_{e3})$. Finally, the output HDR image is:

$$I_r = \left(\frac{I_1}{\Delta t_{e1}} + \frac{I_{b1}}{\Delta t_{e1} + \Delta t_{e2}} + \frac{I_{b2}}{\Delta t_{e1} + \Delta t_{e2} + \Delta t_{e3}} \right) / 3. \quad (8)$$

The optimal exposure ratios $\Delta t_{e3} : \Delta t_{e2} : \Delta t_{e1}$ should be determined by considering both the desired extended dynamic range as well as the noise amplification due to the motion deblurring. Intuitively, the larger $\Delta t_{e3} : \Delta t_{e1}$ is, the larger the extended dynamic range should be, but a larger ratio can also amplify more noise during motion deblurring and in turn lower the effective dynamic range. An analysis of the noise amplification and the selection of the exposure ratios can be found in the supplementary document.

In our experiments, we set $\Delta t_{e2} = 2\Delta t_{e1}$ and $\Delta t_{e3} =$

²More discussion is in Section 6.

$8\Delta t_{e1}$, and thus the improvement in dynamic range will be $20\log(\Delta t_{e3}/\Delta t_{e1}) = 20\log 8 = 18.06\text{dB}$. We set the camera motion to be $\vec{u}_0 = [1, 1]$ pixels per Δt_{e1} time. We use the deblurring algorithm presented in [7]. Simulation experiments are performed using a set of ten high-res HDR images, collected from multiple sources online [21].

Quantitative evaluation is shown in Table 1. We compared our method (*i.e.*, I_r) with the three other single-shot methods using a conventional rolling shutter (*i.e.*, short exposure $I_1/\Delta t_{e1}$, medial exposure $I_2/\Delta t_{e2}$, or long exposure $I_3/\Delta t_{e3}$). The performance for each method is measured as the Normalized Root Mean Square Error (NRMSE) between the recovered HDR image and the original scene, taking into account the dynamic range of the original scene: $\text{NRMSE}(I_0, \hat{I}_0) = \frac{\sqrt{\|I_0 - \hat{I}_0\|^2/N}}{\max(I_0) - \min(I_0)}$, where I_0 is the original scene, \hat{I}_0 is the output image for a given method (*i.e.*, I_r or $I_i/\Delta t_{ei}$, $i = 1, 2, 3$), and N is the number of pixels.

We ran the simulation with two types of image noise. First, we assumed Gaussian additive noise (*i.e.*, scene independent noise), and performed the simulation with seven levels of noise. Second, we considered Gaussian photon noise. We measured the photon noise parameters for a Canon EOS 20D camera at five ISO values, and used them to simulate the photon noise in the captured images. For each method and each level of image noise, we simulated the captured image with motion blur due to camera shake, image noise, and saturation due to limited dynamic range. The simulated images were used to recover HDR image I . We repeated the simulation on the ten HDR images and took the average.

The results are listed in Table 1. Our method (the coded rolling shutter) performs best across all levels of noise. Moreover, as expected, among the three exposures using a conventional rolling shutter camera, for low image noise, the short exposure recovers the HDR image well. As image noise increases, the medial exposure yields better result. With extensive noise, despite of saturation and motion blur, the long exposure is better. With a coded rolling shutter, our method combines the merits of these three exposure settings and performs consistently better than the others.

Figure 10 shows one example. The simulated input image I is shown in Fig. 10a, generated according to the coding pattern in Fig. 9. Gaussian noise ($\sigma = 0.005$) is added in I . The three sub-images, I_1 , I_2 , and I_3 , are shown in Figs. 10(b,c,d). Compared with the final output image I_r , these sub-images are either too dark and noisy or too blurry and saturated, as shown in Figs. 10(g,h). Figure 11 shows another set of experimental results.

6. Conclusion and Discussion

Summary In this paper, we proposed a new readout architecture for CMOS image sensors, called *coded rolling shutter*. By controlling the readout timing and exposure per row,

we demonstrated several coding schemes that can be applied within one frame and their applications. The required controls can be readily implemented in standard CMOS image sensors. As summarized in Table 2, we achieve benefits such as less skew (*i.e.*, time lag within a sub-image) or higher temporal resolution (*i.e.*, time lag between two consecutive sub-images) or higher dynamic range, at the cost of reduced vertical resolution. One future direction is to design coding schemes for multiple frames, where existing de-interlacing methods could be leveraged to increase vertical resolution.

Vertical Resolution and Aliasing As mentioned, all the other applications trade off vertical resolution for other features (except for the adaptive row-wise auto-exposure in Sec. 5.1). Aliasing due to the cubic interpolation might cause noticeable artifacts, especially for the motion deblurring in Sec. 5.2. Based on [26], we analyzed the aliasing caused by image down-sampling and up-sampling, and found that by simply combining the down-sampled images at different phases (*e.g.*, the image of all odd rows and the image of all even rows), the aliasing will be effectively alleviated (when the blur kernels are the same for the down-sampled images, aliasing can be completely avoided.) — this is why we used the combined images for HDR imaging in Sec. 5.2. We note that horizontal resolution is always fully retained. One interesting future direction is to *transfer* the high frequency details from the horizontal direction to the vertical direction.

Random Coding Pattern and Sparse Reconstruction If we model the scene brightness for one pixel (x, y) over time t as a 1-D signal, the corresponding pixel intensity in the captured image is a *linear projection* of this 1-D signal with the exposure pattern. Thus, with (random) coded exposure patterns, we attempted to reconstruct the space-time volume (with zero skew) from a single shot by exploiting the sparsity in signal gradients. In simulation, we found that although the method could effectively remove skew, many high-frequency artifacts would be present, especially around strong vertical edges. Removal of these artifacts will be the subject of our future research.

Pixel-wise Exposure Control CMOS image sensors are able to address individual pixels [17], provided that there is enough bandwidth for data transmission on the chip. One future work is to look into possible implementations of pixel-wise exposure control on chip and achieve even more flexibility for space-time sampling.

References

- [1] O. Ait-Aider, N. Andreff, J. M. Lavest, and P. Martinet. Simultaneous object pose and velocity computation using a single view from a rolling shutter camera. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2006.

Table 1. Performance (NRMSE) of HDR imaging with camera shake using coded rolling shutter

Methods	Gaussian Additive Noise (σ)							Gaussian Photon Noise				
	0.0005	0.001	0.002	0.005	0.01	0.02	0.05	ISO 100	ISO 200	ISO 400	ISO 800	ISO 1600
$I_1/\Delta t_{e1}$	0.0184	0.0201	0.0250	0.0417	0.0641	0.0962	0.1579	0.0181	0.0184	0.0190	0.0202	0.0222
$I_2/\Delta t_{e2}$	0.0224	0.0226	0.0235	0.0280	0.0375	0.0550	0.0940	0.0224	0.0225	0.0227	0.0230	0.0236
$I_3/\Delta t_{e3}$	0.0644	0.0644	0.0644	0.0645	0.0650	0.0666	0.0739	0.0644	0.0644	0.0644	0.0645	0.0645
I_r	0.0176	0.0178	0.0183	0.0205	0.0255	0.0373	0.0736	0.0176	0.0177	0.0178	0.0180	0.0182

* The best performer for each noise level is shown in bold.

Table 2. Summary of Coded Rolling Shutter Photography

Control Schemes	Sub-image Rows (Before Interpolation)	Output Images	Time Lag within one Sub-image	Time Lag between two consecutive Sub-images	Applications
none	M	1	$M\Delta t_r$	$M\Delta t_r$	conventional rolling shutter
interlaced readout	M/K	K	$M\Delta t_r/K$	$M\Delta t_r/K$	slow motion, skew compensation
staggered readout	M/K	K	$(M-K+1)\Delta t_r$	Δt_r	high speed photography
row-wise exposure	M	1 (HDR)	$M\Delta t_r$	$M\Delta t_r$	adaptive row-wise auto-exposure
staggered readout, row-wise exposure	M/K	1 (HDR)	$(M-K+1)\Delta t_r$	Δt_r	HDR imaging for hand-held cameras

* The benefits are shown in bold.

- [2] O. Ait-Aider, A. Bartoli, and N. Andreff. Kinematics from lines in a single rolling shutter image. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.
- [3] O. Ait-Aider and F. Berry. Structure and kinematics triangulation with a rolling shutter stereo rig. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2009.
- [4] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2007.
- [5] E. B. Bellers and G. de Haan. *De-Interlacing: A Key Technology for Scan Rate Conversion*. Elsevier Science B. V., 2000.
- [6] D. Bradley, B. Atcheson, I. Ihrke, and W. Heidrich. Synchronization and rolling shutter compensation for consumer video camera arrays. In *IEEE International Workshop on Projector-Camera Systems (PRO-CAMS)*, 2009.
- [7] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, August 2007.
- [8] P. E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of SIGGRAPH*, 1997.
- [9] E. R. Fossum. CMOS image sensors: Electronic camera-on-a-chip. *IEEE Transactions on Electron Devices*, 44(10):1689–1698, 1997.
- [10] O. Gallo, N. Gelfand, W.-c. Chen, M. Tico, and K. Pulli. Artifact-free high dynamic range imaging. In *Proceedings of IEEE International Conference on Computational Photography (ICCP)*, 2009.
- [11] C. Geyer, M. Meingast, and S. Sastry. Geometric models of rolling-shutter cameras. In *IEEE Workshop on Omnidirectional Vision*, 2005.
- [12] S. E. Kemeny, R. Panicacci, B. Pain, L. Matthies, and E. R. Fossum. Multiresolution image sensor. *IEEE Transactions on Circuits and Systems For Video Technology*, 7(4):575–583, August 1997.
- [13] C.-K. Liang, L.-W. Chang, and H. H. Chen. Analysis and compensation of rolling shutter effect. *IEEE Transactions on Image Processing*, 17(8):1323–1330, 2008.
- [14] P.-Y. Lu, T.-H. Huang, M.-S. Wu, Y.-T. Cheng, and Y.-Y. Chuang. High dynamic range image reconstruction from hand-held cameras. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- [15] M. Mase, S. Kawahito, M. Sasaki, Y. Wakamori, and M. Furuta. A wide dynamic range CMOS image sensor with multiple exposure-time signal outputs and 12-bit column-parallel cyclic A/D converters. *IEEE Journal of Solid-State Circuits*, 40(12):2787–2795, December 2005.
- [16] T. Mitsunaga and S. K. Nayar. Radiometric self calibration. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 374–380, June 1999.
- [17] J. Nakamura. *Image Sensors and Signal Processing for Digital Still Cameras*. CRC Press, 2006.
- [18] S. K. Nayar and V. Branzoi. Adaptive dynamic range imaging: Optical control of pixel exposures over space and time. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1168–1175, October 2003.
- [19] S. K. Nayar and T. Mitsunaga. High dynamic range imaging: Spatially varying pixel exposures. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 472–479, June 2000.
- [20] J. Ohta. *Smart CMOS Image Sensors and Applications*. CRC Press, 2007.
- [21] PFSTools. <http://pfstools.sf.net/>.
- [22] R. Raskar, A. Agrawal, and J. Tumblin. Coded exposure photography: Motion deblurring using flutter shutter. *ACM Transactions on Graphics (SIGGRAPH)*, 25(3):795–804, July 2006.
- [23] VisionResearch. <http://www.visionresearch.com/>.
- [24] B. Wilburn, N. Joshi, V. Vaish, M. Levoy, and M. Horowitz. High speed video using a dense camera array. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.
- [25] O. Yadid-Pecht and E. R. Fossum. Wide intrascene dynamic range CMOS APS using dual sampling. *IEEE Transactions on Electron Devices*, 44(10):1721–1723, October 1997.
- [26] A. Youssef. Image downsampling and upsampling methods. In *International Conference on Imaging, Science, Systems, and Technology (CISST '99)*, pages 132–138, Las Vegas, June 1999.
- [27] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Image deblurring with blurred and noisy image pairs. *ACM Transactions on Graphics (SIGGRAPH)*, 26(3):1:1–1:10, July 2007.