

# Jittered Exposures for Image Super-Resolution

Nianyi Li<sup>1</sup>   Scott McCloskey<sup>2</sup>   Jingyi Yu<sup>3,1</sup>

<sup>1</sup>University of Delaware, Newark, DE, USA. [nianyi@eecis.udel.edu](mailto:nianyi@eecis.udel.edu)

<sup>2</sup>Honeywell ACST, Golden Valley, MN, USA. [Scott.McCloskey@Honeywell.com](mailto:Scott.McCloskey@Honeywell.com)

<sup>3</sup>ShanghaiTech University, Shanghai, China. [yujingyi@shanghaitech.edu.cn](mailto:yujingyi@shanghaitech.edu.cn)

## Abstract

Camera design involves tradeoffs between spatial and temporal resolution. For instance, traditional cameras provide either high spatial resolution (e.g., DSLRs) or high frame rate, but not both. Our approach exploits the optical stabilization hardware already present in commercial cameras and increasingly available in smartphones. Whereas single image super-resolution (SR) methods can produce convincing-looking images and have recently been shown to improve the performance of certain vision tasks, they are still limited in their ability to fully recover information lost due to under-sampling. In this paper, we present a new imaging technique that efficiently trades temporal resolution for spatial resolution in excess of the sensor’s pixel count without attenuating light or adding additional hardware. On the hardware front, we demonstrate that the consumer-grade optical stabilization hardware provides enough precision in its static position to enable physically-correct SR. On the algorithm front, we elaborately design the Image Stabilization (IS) lens position pattern so that the SR can be efficiently conducted via image deconvolution. Compared with state-of-the-art solutions, our approach significantly reduces the computation complexity of the processing step while still providing the same level of optical fidelity, especially on quantifiable performance metrics from optical character recognition (OCR) and barcode decoding.

## 1. Introduction

Image super-resolution (SR) is a long standing problem in computer vision and image processing. The goal of SR is to generate a physically correct result from a resolution-limited optical target [7, 10]. Whereas single image super-resolution methods [3, 5, 8, 14, 17] can produce convincing-looking images - and have recently been shown to improve the performance of certain vision tasks - they are still limited in their ability to fully recover information lost due to under-sampling. More recent solutions attempt to combine multiple low-resolution images into a high-resolution result.

The process can be formulated using an observation model that takes low-resolution (LR) images  $\mathcal{L}$  as the blurred result of a high resolution image  $\mathcal{H}$ :

$$\mathcal{L} = W\mathcal{H} + n_k \quad (1)$$

where  $W_k = DBM$ , in which  $M$  is the warp matrix (translation, rotation, etc.),  $B$  is the blur matrix, and  $D$  is the decimation matrix.

Most state-of-the-art image SR methods consist of three stages, i.e., registration, interpolation and restoration (an inverse procedure). These steps can be implemented separately or simultaneously according to the reconstruction methods adopted. Registration refers to the procedure of estimating the motion information or the warp function  $M$  between  $\mathcal{H}$  and  $\mathcal{L}$ . In most cases, the relative shifts between the LR image and the high resolution (HR) image are within a fraction of a pixel. Further, such shifts, by design or by nature, can be highly non-uniform and therefore a precise subpixel motion estimation and subsequent non-uniform interpolation are crucial to conduct reliable image SR. However, to robustly conduct nonuniform motion estimation and then interpolation, previous solutions rely on sophisticated optimization schemes, e.g., an iterative update procedure. Such schemes are computationally expensive and are not always guaranteed to converge, as illustrated in Table 1.

In this paper, we present a computational imaging solution for producing a (very) high-resolution image using a low-resolution sensor. Our approach exploits the optical stabilization hardware already present in commercial cameras and increasingly available in smartphones. On the hardware front, we demonstrate that consumer-grade optical stabilization hardware provides enough precision in its static position to enable physically-correct SR, and thus that expensive custom hardware (used previously for SR [1]) is unnecessary. In light of the increasing availability of optical stabilization in smartphones (Apple’s iPhone 8s, etc.), our use of this hardware will ultimately enable hand-held SR on smartphones, as described below.

On the algorithm front, we elaborately design the IS lens position pattern so that SR can be efficiently conducted vi-

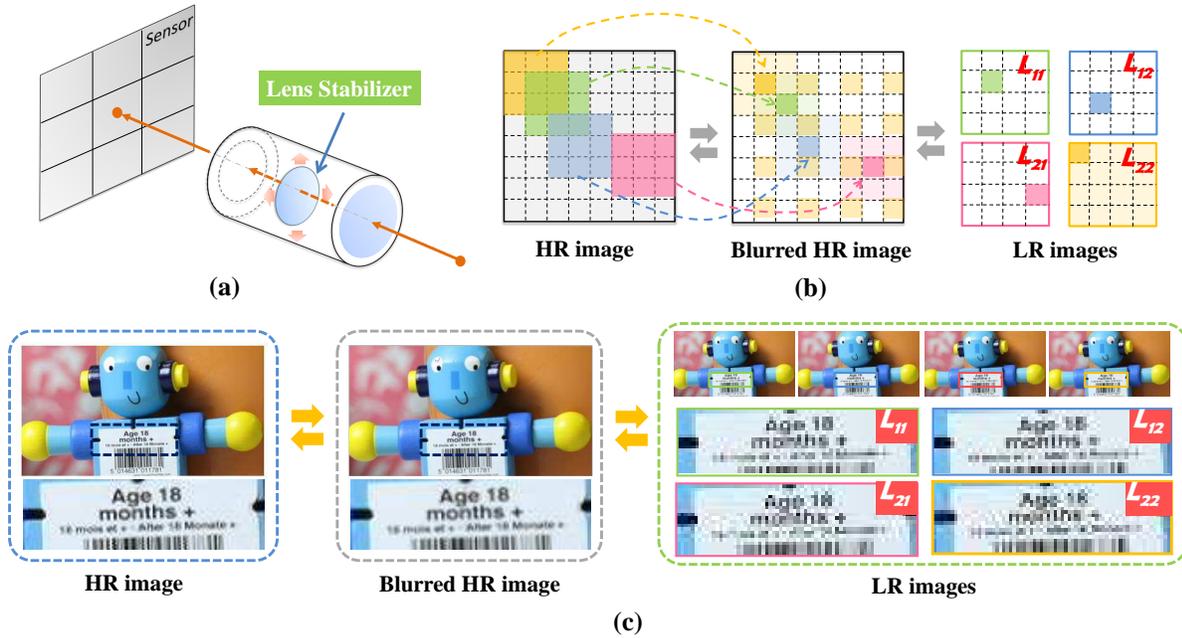


Figure 1. From Left to Right:(a)Camera with a modified image stabilizer controls the movement of IS lens unit. (b)(c) The algorithm overview.

a image deconvolution/deblurring. Compared with state-of-the-art solutions, our approach significantly reduces the computational complexity of the processing step while still providing the same level of optical fidelity. In order to avoid well-known issues with low-level image quality metrics, we demonstrate the advantages of our method with quantifiable performance metrics from optical character recognition (OCR) and barcode decoding.

### 1.1. Related Work

Single-image, learning-based SR methods, reviewed in [12], have shown continued improvement recently accelerated by deep learning [2, 5, 9, 15]. Performance-wise, single image SR methods produce aesthetically convincing results, and are more useful for vision tasks than the LR inputs [4], but still fall short of a HR image because of ambiguities which can't be resolved. This is because single image SR is ill-posed: any given low-resolution image is consistent with *multiple* high-resolution originals. Image priors and learned mappings may prove useful in certain cases, but can't render the problem well-posed. Fig. 2 illustrates this with a barcode; a gray pixel in the image is as likely to come from a sequence of a white bar followed by a black bar as from a black bar followed by a white bar. The figure also shows that the JOR method [3] - identified in [4] as one of the highest-performing single image methods - is unable to resolve this ambiguity under  $2\times$  SR *even when it is trained on barcode data*; the JOR result cannot be decoded.

Our approach falls into the category of multiple-image SR. Popular approaches include iterative back projection (IBP) [8], which achieve SR by iteratively updating the es-

timated HR image according to the difference between its simulated LR images and the observed ones, and [6] maximize a posteriori and regularized maximum likelihood to restore a SR images form several blurred, noisy and under-sampled LR images. Most recently, sparse representation in line with compressive sensing has attracted much attention [16]. Their approach learns dictionaries from sample images and the HR images can be reconstructed by a global optimization algorithm. These reconstruction techniques focus solving the ill-posed inverse problem with a conventional sensor and setup.

The most similar prior work to ours is the jitter camera [1], which captures video while the sensor 'jitters' between different positions. In addition to the user of additional hardware to control the motion, the jitter camera algorithm is based on a back-projection algorithm which, due to its iterative nature, has prohibitively high computational complexity. Our work demonstrates that *neither* the custom hardware *nor* the high computational complexity are necessary to produce an SR image with improved utility (for recognition) relative to the LR images. We capture our LR images using commodity image stabilization hardware, and demonstrate that SR can be performed much faster *without any reduction in the utility of the SR image*.

## 2. Jittered Lens Super-Resolution

We start with image SR from multiple LR images, captured under different positions of the IS hardware, with an emphasis on streamlining algorithms to reduce the temporal resolution loss. We first introduce our prototype lens,



Figure 2. Single image SR is ill-posed. An original barcode image near the resolution limit, after  $2\times$  downsampling and SR via JOR [3], cannot be recovered from a single image. Ambiguities in the green-dashed region prevent the JOR image from being decoded.

and then analyze how to locate each captured LR image to generate HR images.

## 2.1. Camera prototype

In contrast to the former jitter camera [1] which controls the movement of the sensor, our camera achieves precise sub-pixel movement without the need for additional hardware by controlling the lens stabilizer. Fig. 4 shows the prototype of our camera where, once the shutter button has been pressed, the hot shoe signal triggers an external stabilizer controller to modify the IS lens’s position to a location of our choosing. Note that if we can access the Application programming interface (API) of the stabilizer controller within the camera, we don’t need to modify the camera body at all. In that sense, it is *not* necessary to add *any* additional hardware to achieve SR beyond the sensor’s pixels count.

As shown in Fig. 4, we mounted the modified Canon EF 100mm/2.8L Macro IS lens to a Canon 70D DSLR camera body. The Canon Optical Image Stabilization (OIS) module includes a light-weight lens encased in a housing with a metallic yoke, motion of which is achieved by modulating a voice coil. When the shutter release is depressed halfway, a mechanical lock is released and the lens is moved by a processor via pulse width modulation. As described in [11], the standard mode of that processor is a closed control loop where motion detected by gyroscopes mounted within the lens induces a compensating motion of the stabilizing element. In order to drive the lens to the desired positions while capturing our LR images, we break the control loop and decouple the stabilizing element from the motion sensor. We utilize an independent microprocessor to control the movement of IS lens through pulse width modulation supervised by a proportional-integral-derivative (PID) controller control loop. Once the shutter button is pressed, the hot shoe signal initiates our PID control loop to hold at a programmed position.

## 2.2. Multi-Image SR via De-blurring

For the purpose of analysis, consider the problem of recovering a high resolution 1-D signal via deconvolution. The goal is to estimate the signal  $H(x)$ , that was blurred by a linear system’s point-spread function (PSF)  $P(x)$ . The

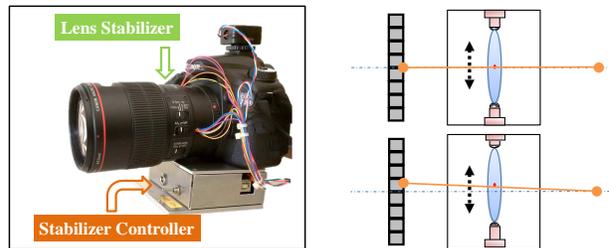


Figure 4. From Left to Right: (a) Our prototype Jitter Camera. (b) The IS lens stabilizer controller enables precise sub-pixel movement of the center of projection.

measured image signal  $I(x)$  is then known to be

$$I(x) = P(x) * H(x) \quad (2)$$

with  $*$  denoting convolution. In the ideal case, a good estimate of the image,  $H'(x)$ , can be recovered via a deconvolution filter  $P^+(x)$

$$H'(x) = P^+(x) * I(x) \quad (3)$$

In a more general case, we describe convolution using linear algebra. Let  $\mathcal{I}$  denote the blurred input image pixel values. Each pixel of  $\mathcal{I}$  is a linear combination of the intensities in the desired unblurred image  $\mathcal{H}$ , and can be written as:

$$\mathcal{I} = \mathcal{P}\mathcal{H} + \epsilon \quad (4)$$

The matrix  $\mathcal{P}$  is the smearing matrix, effecting the convolution of the input image with the point spread function  $P(x, y)$  and  $\epsilon$  describes the measurement uncertainty due to noise, quantization error and model inaccuracies. For two-dimensional PSF’s,  $\mathcal{P}$  is a block-circulant matrix. For simplicity, we only discuss how to locate LR pixels to generate SR image along horizontal direction. Extending this to the vertical direction for 2D is straightforward.

Given an estimated PSF  $P$  (or  $B_k$ ), our goal is find lens positions that can recover the blurred image  $\mathcal{I}$  from a minimal set of LR images  $\{\mathcal{L}_k | k = 1, \dots, K\}$  of the form in [12]:

$$\mathcal{L}_k = (D_k B_k) \mathcal{H} = D_k (B_k \mathcal{H}) = D_k (\mathcal{P}\mathcal{H}) = D_k \mathcal{I} \quad (5)$$

Eqn. 5 indicates that the LR images are derived directly from the blurred HR image  $\mathcal{I}$  with different downsample patterns  $D_k$ . Theoretically, to recover  $\mathcal{I}$ ,  $\{\mathcal{L}_k | k = 1, \dots, K\}$  should contain all pixels of  $\mathcal{I}$ . To ensure the minimal set  $\{\mathcal{L}_k | k = 1, \dots, K\}$ , there should be no overlapping pixels within  $\{\mathcal{L}_k | k = 1, \dots, K\}$ . As a result, we can easily conclude that the relative motion between neighboring sub-captures  $\{\mathcal{L}_k\}$  and  $\{\mathcal{L}_{k+1}\}$  should satisfy:

$$\Delta_{\mathcal{I}} = \frac{\text{size}(\text{pixel})}{K} \quad (6)$$

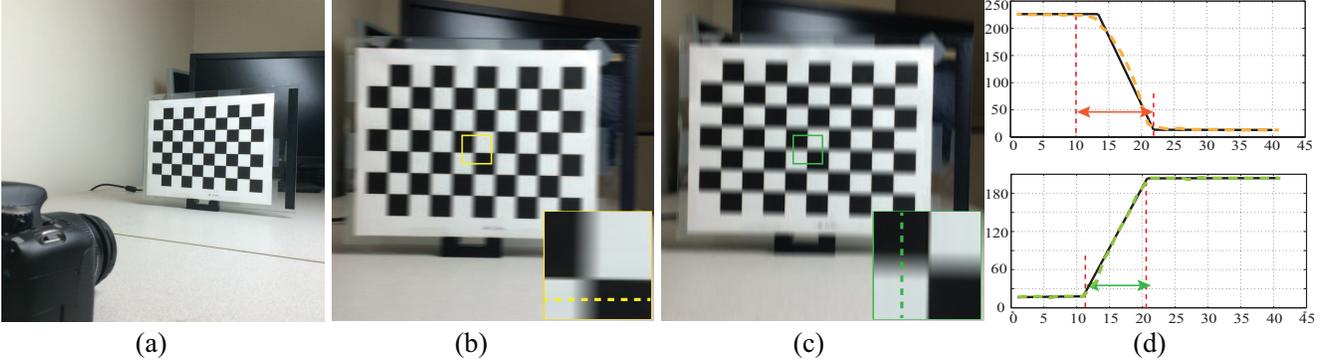


Figure 3. (a) We utilize the motion blur artifacts to calculate the pixel size of LR images. (b)(c) We capture the motion blur along horizontal and vertical axis and count the blurred pixel number during each motion direction. (d) The blurred sharp edge along horizontal (the upper) and vertical (the bottom) directions respectively. Desired (black solid line) versus actual (orange and green dash lines) blur value.

where  $K$  is the expected magnification factor, that is  $size(\mathcal{I})/size(\mathcal{L}_k) = K$ .

Suppose that the LR images are of size  $M \times N$ , and  $\mathcal{L}_k(m, n)$  denotes the pixel  $(m, n)$  on the LR image  $\mathcal{L}_k$ , where  $m = 1, \dots, M$  and  $n = 1, \dots, N$ . The corresponding relationship between  $\mathcal{L}_k(m, n)$  and  $\mathcal{I}(x, y)$  is:

$$\mathcal{I}((m-1)K + k, (n-1)K + k) = \mathcal{L}_k(m, n), \quad (7)$$

as illustrated in Fig. 1 (b). Having reconstructed  $\mathcal{I}$ , we can solve the deblurred high resolution image  $\mathcal{H}$  as:

$$\mathcal{H} = (\mathcal{P}^T \mathcal{P})^{-1} \mathcal{P}^T \mathcal{I} \quad (8)$$

Fig. 8 shows several SR results of real scenes by our method. We also compare our method with IBP in Section. 3.2.

### 3. Experiments

#### 3.1. Camera Calibration

In order to achieve the precise subpixel movement, we need to calculate the exact pixel size w.r.t. the moving unit of the IS lens. While the approximate pixel pitch (about  $4\mu\text{m}$  for our camera) on the sensor is useful for images captured at the full resolution, the effective pixel pitch for other resolutions is not publicized in the camera specifications. To estimate this quantity, we use motion blur artifacts under a known lens motion. Instead of keeping the IS lens at a fixed position during the exposure time, the images with motion blur is achieved by changing the lens position at a constant speed during exposure. By doing this, sharp edges in the scene will be motion blurred in the captured image. Since that we already know the moving length of the controlled IS lens, i.e.,  $d^l$ , during the exposure period, the pixel size w.r.t to camera lens is computed as:

$$p = \frac{d^l}{p^b} \quad (9)$$

where  $p_b$  is the blurred pixel number along the lens motion direction. The procedure is presented in Fig. 3. Fig. 3(d) shows the average blur change along  $x$  and  $y$  axis while the IS lens position changing 500 units. The black solid lines indicates the desired value curves. The orange dash line denotes the average blur value curve along  $x$  axis, and the green one corresponds to the curve along  $y$  axis. The high correspondence between the desired and actual curves indicates the high accuracy of our PID controlled IS lens motion.

As discussed in Section 3.2, an inaccurate estimation of pixel size will largely degrade our performance. To obtain the precise pixel size, we captured six motion blurred images of the same checker board, 3 images for each of the two motion.  $p_x^b$  and  $p_y^b$  are the average of the three computed values. Our estimated pixel size w.r.t. the lens unit under the resolution  $720 \times 480$  is  $[56, 44]$ . If the lens position changed 56 units along the horizontal direction, the captured image will shift 1 pixel along  $x$ -axis. For instance, to achieve  $2\times$  super-resolution, we will capture 4 images at  $\{(c_x, c_y), (c_x - 28, c_y), (c_x, c_y - 22), (c_x - 28, c_y - 22)\}$ , where  $(c_x, c_y)$  is the center location of the lens. Notice that if the captured edge is not sharp enough, the estimated  $p^b$  might be incorrect. In the ideal situation, we hope to capture sharp edges along both  $x$  and  $y$  axis.

#### 3.2. Error Estimation

We then analyze the errors that might be introduced to our capture system. Next, we compare the errors tolerance ability between our methods and the back-projection technique [6]. There are two main sources of error in our framework. One results from the incorrect predicted blur kernel (PSF), i.e.,  $\epsilon^B$ , the other one  $\epsilon^P$  is caused by the wrongly estimated pixel size.

**Error from PSF:  $\epsilon^B$**  To simplify our discussion, the PSFs discussed in this paper have the same blur kernel  $\mathcal{F}$  of

	Methods	ANR	Zeyde <i>et al.</i>	A+	SRCNN	JOR	IBP	Ours
Maximum Distance (in.)	2× super-res	<40	<40	<40	<40	<40	70	70
	3× super-res	<40	<40	<40	<40	<40	80	80
Running time (sec.)	2× super-res	0.87	3.82	0.94	4.99	87.78	5.54	0.61
	3× super-res	1.24	3.60	1.30	12.14	100.65	20.96	1.57

Table 1. The maximum distance at which we could decode a UPC symbol, and the average running time of different SR algorithms. The testing LR images are of size  $218 \times 212$ . All methods are implemented in Matlab R2013a.

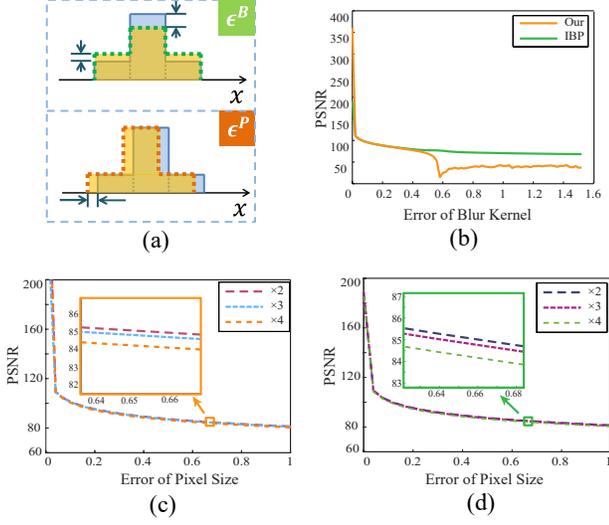


Figure 5. (a) Errors caused by wrongly estimated pixel sizes. (b) The PSNR curves of our method and IBP w.r.t.  $\|\epsilon^B\|_1$ . (c) The PSNR curves of our methods under different SR size w.r.t.  $\|\epsilon^P\|_1$ . (d) The PSNR curves of IBP.

size  $1 \times m$  along horizontal and vertical directions. Recall that valid blur kernel should satisfy two criterion: 1) its elements should be non-negative; 2) the sum of all elements in it should be equal to 1. Based on these definition, the estimated blur kernel  $\hat{\mathcal{F}}$  can be written as:

$$\hat{\mathcal{F}} = \mathcal{F} + \epsilon^B \quad (10)$$

where  $\epsilon^B$  describes the errors vector of  $\mathcal{F}$ . In our discussion, we assume that  $\hat{\mathcal{F}}$  and  $\mathcal{F}$  are of the same length. We first synthesize 4 LR images from a given HR image according to Eqn. 1. The ground truth blur kernel  $\mathcal{F}$  is a Gaussian kernel with the size of  $1 \times 3$  and  $\sigma = 0.5$ . We then apply the incorrect blur kernel  $\hat{\mathcal{F}}$  to generate an estimated HR image. Specifically, for a given  $\|\epsilon^B\|_1 = \beta$ , we set  $\epsilon^B = \frac{1}{4}[\beta, -2\beta, \beta]$ , and  $\hat{\mathcal{F}}$  is generated by Eqn. 10. Fig. 5(c) presents the relationship between PSNR curves vs.  $\|\epsilon^B\|_1$ . Notice that when  $\mathcal{P}^{-1}$  in Eqn. 8 is approximate to a singular matrix, there will be serious artifacts in our result (when  $\|\epsilon^B\|_1 > 0.5$ ). A detailed analysis of this artifacts is shown in supplementary materials. We can tell that keep-

ing  $\|\epsilon^B\|_1$  under 0.1 is essential in rendering high quality super-resolution images.

**Error from Pixel Size:  $\epsilon^P$**  Another crucial errors in our scheme is caused by the inaccurate estimated pixel size  $(p_x, p_y)$ , as shown in Fig. 5(a).  $\epsilon_x^P$  and  $\epsilon_y^P$  describe the estimation errors of pixel size along  $x$  and  $y$ -axis respectively. The incorrect estimated pixel size is  $(\hat{p}_x, \hat{p}_y)$ , where  $\hat{p}_x = p_x + \epsilon_x^P$  and  $\hat{p}_y = p_y + \epsilon_y^P$ . Fig. 5(d) plots the PSNR curves of IBP and our method w.r.t.  $\|\epsilon_x^P\|_1$  under  $2\times$ ,  $3\times$  and  $4\times$  super-resolution respectively. Suppose that we want to achieve  $M\times$  super-resolution, according to Section 2.2,  $M^2$  numbers of low resolution images are needed to construct the blurred high resolution image  $\mathcal{I}$ . If we only require information along  $x$ -axis, the relative locations between subcapture images to the first LR image should be

$$\left\{ i \cdot \frac{\hat{p}_x}{M}, i = 0, \dots, M-1 \right\}$$

This indicates that  $\epsilon^P$  is accumulated as the increasing of super-resolution size, which has been verified in Fig. 5 (d).

**When is Iteration Required?** Our experiments clearly indicate that consumer-grade IS hardware provides sufficient positional accuracy to obviate the iterative approach of IBP, as we have the same recognition performance while being an order of magnitude faster (as shown in Table. 1). At what level of error, then, does it become necessary to incur additional computational complexity? To compare the error tolerance of our method to IBP, we test the two algorithms on 10 sets of synthetic images and plot their average PSNR curves w.r.t. two nuisance factors: having an error in the PSF  $\|\epsilon^B\|_1$ , and in the pixel size  $\|\epsilon^P\|_1$ . Fig. 5 (b)(c)(d) shows this for different super-resolution factors, revealing that our algorithm achieves higher PSNR value when the errors are small. As errors get larger, the performance advantage switches to IBP.

### 3.3. Performance Evaluation

We capture real scenes with our camera at the resolution  $720 \times 480$ . Because none of the higher resolution modes are

an integer multiple of these dimensions, we are unable to directly capture the ground truth HR images to compute traditional SR metrics, i.e., *PSNR*, *SSIM*, *IFC* etc. Instead, we evaluate image *utility* metrics which we believe - in light of [4] - is more meaningful than evaluating the SR images *quality*. One measures character recognition rate, and the other tests whether the reconstructed SR barcode images can be decoded. We compare to 6 state-of-the-art SR algorithms: *ANR* [14], *Zeyde* [17], *A+* [13], *SRCNN* [5], *JOR* [3] and *IBP* [8].

**Character Recognition** We first compare the character recognition accuracy of the estimated SR images. We use the OCR in Google drive to convert SR text images into documents and compare with the ground truth text using an online comparison tool<sup>1</sup>, which counts and highlights the mismatching characters in text files automatically.

We use a printed text with 153 words and 880 characters (including spaces), captured at 7 different distances from the camera: [40, 50, 60, 70, 80, 90, 100] inches. At each, we take two sets of LR images for 2× and 3× SR. For single image SR methods, we use the first LR image. Fig. 7 shows curves for 2× SR results of each methods: one for character-level precision and the other for word-level precision. Our algorithm and IBP are very comparable, achieving the best results.

We also note that for single image super-resolution techniques, the text recognition accuracy shows little improvement as the increasing of super-resolution size. By contrast, since IBP and our method use more images to obtain higher SR factors, our text recognition performance increases correspondingly. For instance, when the object-camera distance is 100 inches, the character and word recognition precision of our algorithm are 78.6% and 60.13% in 2× super-resolution and 84.1% and 66.7% in 3× super-resolution; the single image super-resolution models hardly detect anything on both 2× and 3× results at that distance.

**Barcode Recognition** The second metric checks whether the HR barcodes reconstructed by different algorithm can be decoded. We make use of online barcode scanner software<sup>2</sup> to test each HR image. This online barcode scanner shows high quality barcode recognition ability for barcodes even with slight blur, rotation and occlusion.

Since a barcode either decodes properly or doesn't, we don't use recognition rate to measure SR performance here. Instead, we choose the maximum decodable distance between the barcode and camera to quantify SR performance. Similar to the text recognition experiment, at each location, we capture two sets of images for 2× and 3× super-

resolution. Table. 1 shows the maximum recognizable distance for different algorithms. Fig. 8 presents the barcode SR results of different algorithms. Since all the single image SR techniques fail at barcode recognition at 40 inches distance, it is meaningless to further test the exactly recognizable distance for the rest algorithms.

We do not capture images at the full resolution 5472 × 3648 for two reasons. Firstly, higher resolution indicates larger recognition distance. At full resolution, a small barcode of size 1" × 0.6" will be recognizable at even 200 inches, which means that we need to capture object at 7000 inches to draw Table 1 and Fig. 7. Also, as shown in Fig. 3(d), our PID controlled IS lens matches the desired value quite well. Since the actual size of sensor pixel is 4.1 μm, the minimum length of the IS lens motion is

$$\frac{4.1}{Sd^l} = 0.06\mu m \quad (11)$$

where  $S = 5472/720 = 3648/480 = 7.6$  is the downsample scale and  $d^l = 500$  is the IS lens's moving length. Our camera has already achieve sub-pixel shifts on the order of 0.01 μm, and we can therefore choose the low resolution for better implementation and illustration.

**Running Time.** Table. 1 compares the running time of different algorithms. Even though our methods utilize multiple images to reconstruct the SR image, our efficiency is still high. This is because our method eliminates registration and iterative updating, which are widely adopted by other multi-image SR methods.

## 4. Discussion and Conclusion

We have presented a new computational imaging solution for conducting multi-image super-resolution. In our present implementation, we capture the multiple SR images from a stationary camera, and our approach is limited to stationary scenes. While the ability to handle non-stationary scenes will necessarily be limited by the ability to register (potentially multiple) moving objects, the current limitation to a stationary camera is not fundamental. The same stabilization hardware we use to move the lens between captures is more commonly used to maintain the position of the camera's center of projection (relative to the sensor) of a camera while it moves *during* capture. By combining the two, it will be possible to capture a sequence of LR images with the needed shifts, each of which has been stabilized during its own exposure period. We have not done this because we do not wish to duplicate the functionality of the in-exposure stabilization, and because of the amount of engineering effort required to do so.

In the future, we expect that our method can be extended to provide optimized, on-demand SR for optical recognition tasks. Instead of capturing an image with the highest

<sup>1</sup> <http://text-compare.com/>

<sup>2</sup>Intelligent Barcode Scanner: <http://www.ibscanner.com/online-barcode-scanner>

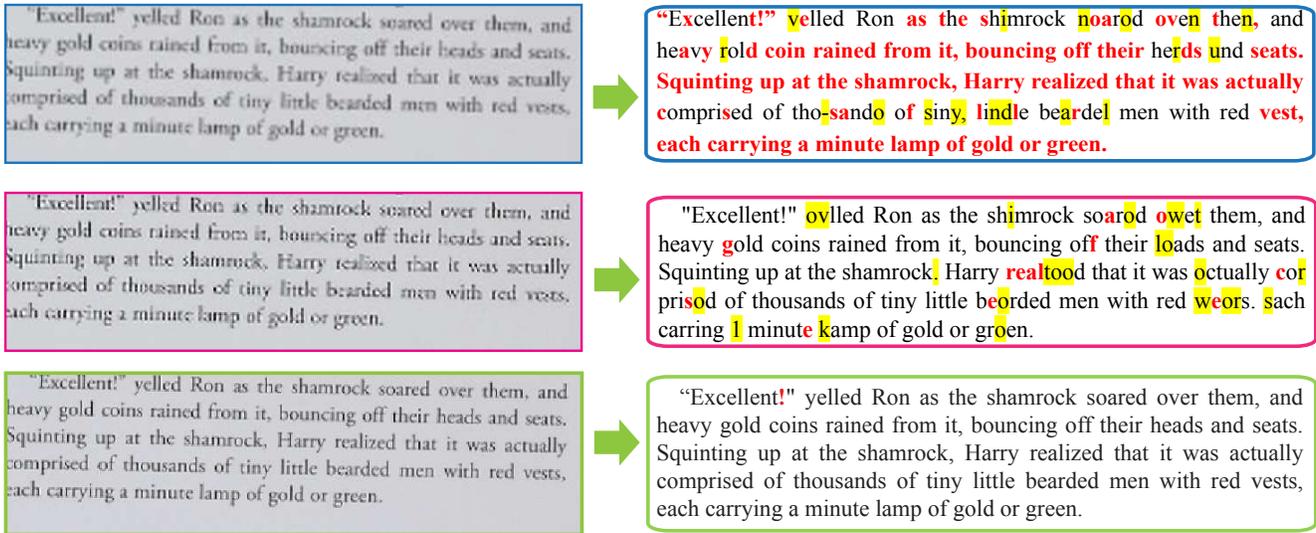


Figure 6. From Left to Right: (a) The SR images with textual contents: Bicubic (the top row) JOR (the middle row) and ours (the bottom row). (b) The text recognition results on different schemes. Our method only produces one recognition error. Red corresponds to missing letter and highlight corresponds to wrong recognition.

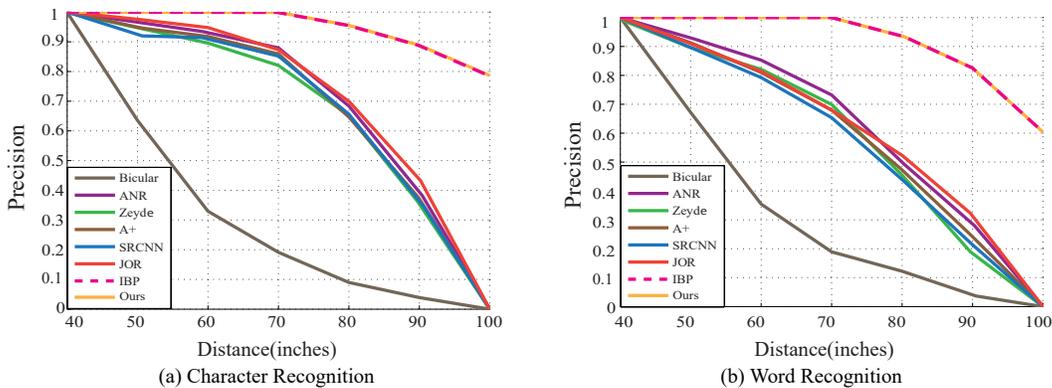


Figure 7. The character and word recognition rates of different SR algorithms w.r.t. different object to camera distance. Our algorithm are of the highest recognition precision.

resolution possible at all times - which increases the computational complexity of recognition applied to the image - we propose instead to capture a first LR image which can be quickly analyzed. If that first image lacks the resolution needed to recognize the target, we can capture additional L-R images and apply our SR method to produce only as much resolution as is needed to recognize the target. For instance, if our first image contains a 1D barcode captured with insufficient resolution, we can capture the minimum number of additional images needed to produce a HR image with the necessary resolution to decode it. By only super-resolving the image in the critical direction, we will reduce the HR image size and allow for more efficient recognition.

## References

- [1] M. Ben-Ezra, A. Zomet, and S. K. Nayar. Video super-resolution using controlled subpixel detector shifts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(6):977–987, 2005. 1, 2, 3
- [2] Z. Cui, H. Chang, S. Shan, B. Zhong, and X. Chen. Deep network cascade for image super-resolution. In *European Conference on Computer Vision*, pages 49–64. Springer, 2014. 2
- [3] D. Dai, R. Timofte, and L. Van Gool. Jointly optimized regressors for image super-resolution. In *Eurographics*, 2015. 1, 2, 3, 6
- [4] D. Dai, Y. Wang, Y. Chen, and L. Van Gool. Is image super-resolution helpful for other vision tasks? In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016. 2, 6

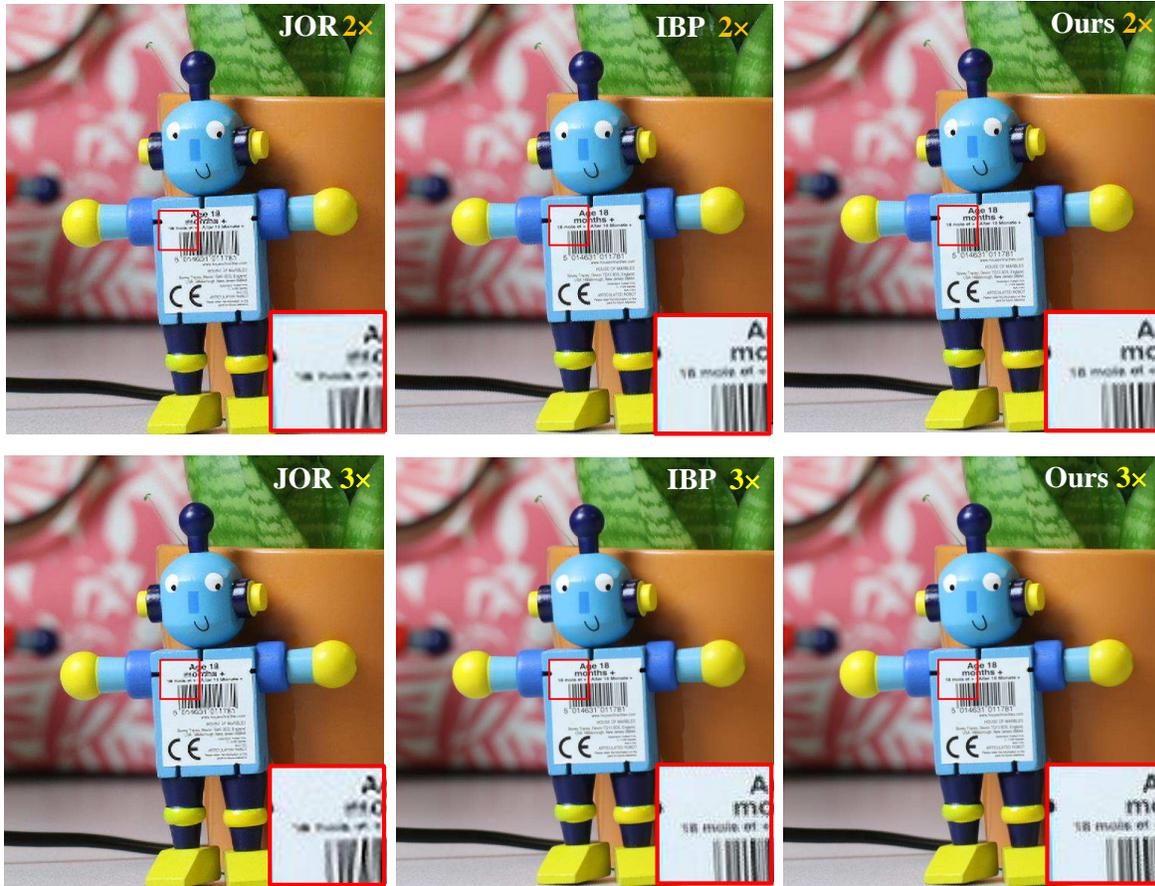


Figure 8. Visual Comparison between results of JOR, IBP and our method at 2 $\times$  and 3 $\times$ .

- [5] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision*, pages 184–199. Springer, 2014. 1, 2, 6
- [6] M. Elad and A. Feuer. Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images. *Image Processing, IEEE Transactions on*, 6(12):1646–1658, 1997. 2, 4
- [7] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 349–356. IEEE, 2009. 1
- [8] M. Irani and S. Peleg. Improving resolution by image registration. *CVGIP: Graphical models and image processing*, 53(3):231–239, 1991. 1, 2, 6
- [9] J. Kim, J. Kwon Lee, and K. Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2
- [10] K. I. Kim and Y. Kwon. Single-image super-resolution using sparse regression and natural image prior. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(6):1127–1133, 2010. 1
- [11] S. McCloskey, K. Muldoon, and S. Venkatesha. Motion aware motion invariance. In *Computational Photography (ICCP), 2014 IEEE International Conference on*, pages 1–9. IEEE, 2014. 3
- [12] S. C. Park, M. K. Park, and M. G. Kang. Super-resolution image reconstruction: a technical overview. *IEEE signal processing magazine*, 20(3):21–36, 2003. 2, 3
- [13] R. Timofte, V. De Smet, and L. Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *Computer Vision-ACCV 2014*, pages 111–126. Springer, 2014. 6
- [14] R. Timofte, V. Smet, and L. Gool. Anchored neighborhood regression for fast example-based super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1920–1927, 2013. 1, 6
- [15] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang. Deep networks for image super-resolution with sparse prior. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 370–378, 2015. 2
- [16] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *Image Processing, IEEE Transactions on*, 19(11):2861–2873, 2010. 2
- [17] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces*, pages 711–730. Springer, 2010. 1, 6